

Research Article

Feature Selection for Very Short-Term Heavy Rainfall Prediction Using Evolutionary Computation

Jae-Hyun Seo,¹ Yong Hee Lee,² and Yong-Hyuk Kim¹

¹ Department of Computer Science and Engineering, Kwangwoon University, 20 Kwangwoon-Ro, Nowon-Gu, Seoul 139-701, Republic of Korea

² Forecast Research Laboratory, National Institute of Meteorological Research, Korea Meteorological Administration, 45 Gisangcheong-gil, Dongjak-gu, Seoul 156-720, Republic of Korea

Correspondence should be addressed to Yong-Hyuk Kim; yhdffy@kw.ac.kr

Received 16 August 2013; Revised 23 October 2013; Accepted 1 November 2013; Published 6 January 2014

Academic Editor: Sven-Erik Gryning

Copyright © 2014 Jae-Hyun Seo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We developed a method to predict heavy rainfall in South Korea with a lead time of one to six hours. We modified the AWS data for the recent four years to perform efficient prediction, through normalizing them to numeric values between 0 and 1 and undersampling them by adjusting the sampling sizes of no-heavy-rain to be equal to the size of heavy-rain. Evolutionary algorithms were used to select important features. Discriminant functions, such as support vector machine (SVM), k -nearest neighbors algorithm (k -NN), and variant k -NN (k -VNN), were adopted in discriminant analysis. We divided our modified AWS data into three parts: the training set, ranging from 2007 to 2008, the validation set, 2009, and the test set, 2010. The validation set was used to select an important subset from input features. The main features selected were *precipitation sensing* and *accumulated precipitation for 24 hours*. In comparative SVM tests using evolutionary algorithms, the results showed that genetic algorithm was considerably superior to differential evolution. The equitable treatment score of SVM with polynomial kernel was the highest among our experiments on average. k -VNN outperformed k -NN, but it was dominated by SVM with polynomial kernel.

1. Introduction

South Korea lies in the temperate zone. In South Korea, we have clearly distinguished four seasons, where spring and fall are short relatively to summer and winter. It is geographically located between the parallels $125^{\circ}04''$ E and $131^{\circ}52''$ E and the meridians $33^{\circ}06''$ N and $38^{\circ}27''$ N in the Northern Hemisphere, on the east coast of the Eurasian Continent, and also adjacent to the Western Pacific, as shown in Figure 1. Therefore, it has complex climate characteristics which show both continental and oceanic features. It has a wide interseasonal temperature difference and much more precipitation than that of the Continent. In addition, it has obvious monsoon season wind, a rainy period from the East Asian Monsoon, locally called *Changma* [1], typhoons, and frequently heavy snowfalls in winter. The area belongs to a wet region because of more precipitation than that of the world average.

The annual mean precipitation of South Korea, as shown in Figure 2, is around 1,500 mm and 1,300 mm in the central

part. Geoje-si of Gyeongsangnam-do has the largest amount of precipitation, 2007.3 mm, and Baegryeong island of Incheon has the lowest amount of precipitation, 825.6 mm.

When a stationary front lingers across the Korean Peninsula for about a month in summer, more than half of the annual precipitation falls during the Changma season. Precipitation for the winter is less than 10% of the total. Changma is a part of the summer Asian monsoon system. It brings frequent heavy rainfall and flash floods for 30 days on average, and serious natural disasters often occur.

The heavy rainfall is one of the major severe weather phenomena in South Korea. The weather phenomena can lead to serious damage and losses of both life and infrastructure, and it is very important to forecast heavy rainfall. However, it is considered a difficult task because it takes place in very short time interval [2].

We need to predict this torrential downpour to prevent the losses of life and property [1, 3]. Heavy rainfall forecasting is very important to avoid or minimize natural disasters

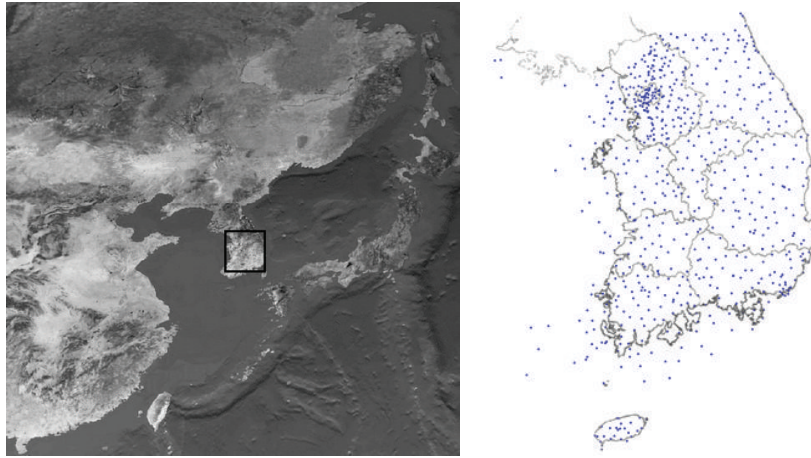


FIGURE 1: The location of South Korea in East Asia and the dispersion of automatic weather stations in South Korea.

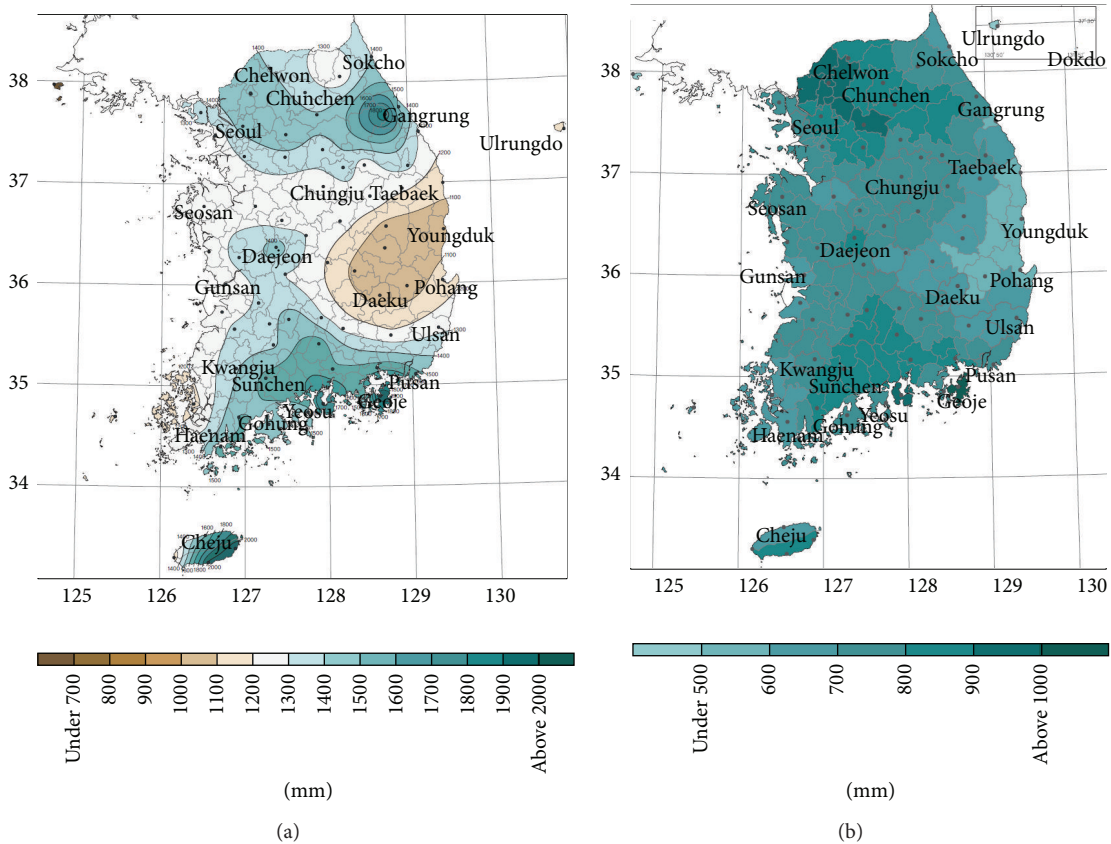


FIGURE 2: Annual (a) and summer (b) mean precipitation in South Korea (mm) [4].

before the events occur. We used real weather data collected from 408 automatic weather stations [4] in South Korea, for the period from 2007 to 2010. We studied the prediction of one hour to six hours of whether or not heavy rainfall will occur in South Korea. To the best knowledge of the authors, this problem has not been handled by other researchers.

There have been many studies on heavy rainfall using various machine learning techniques. In particular, several studies focused on weather forecasting using an artificial

neural network (ANN) [5–11]. In the studies of Ingrisawang et al. [11] and Hong [12], support vector machine was applied to develop classification and prediction models for rainfall forecasts. Our research is different from previous work on how to process weather datasets.

Kishtawal et al. [13] studied the prediction of summer rainfall over India using genetic algorithm (GA). In their study, the genetic algorithm found the equations that best describe the temporal variations of the seasonal rainfall over

India. The geographical region of India has been divided into five homogeneous zones (excluding the North-West Himalayan zone). They used the monthly mean rainfall during the months of June, July, and August. The dataset consist of the training set, ranging from 1871 to 1992, and the validation set, ranging from 1993 to 2003. The experiment of the first evolution process and the second evolution process were conducted using the training set and the validation set, in order. The performance of the algorithm for each case was evaluated, using the statistical criteria of standard error and fitness strength. Chromosome was made up of five homogeneous zones, annual precipitation, and four elementary arithmetic operators. The strongest individuals (equations with best fitness) were then selected to exchange parts of the character strings between reproduction and crossover, while individuals less fitted to the data are discarded. A small percentage of the equation strings' most basic elements, single operators and variables, are mutated at random. The process was repeated a large number of times (about 1,000–10,000) to improve the fitness of the evolving population of equations. The major advantage of using genetic algorithm versus other nonlinear forecasting techniques, such as neural networks, is that an explicit analytical expression for the dynamic evolution of the rainfall time series is obtained. However, they used quite simple or typical parameters of a genetic algorithm. If they conducted experiments by tuning various parameters of their genetic algorithm, they would report the experimental results showing better performance.

Liu et al. [14] proposed a filter method for feature selection. Genetic algorithm was used to select major features in their study, and the features were used for data mining based on machine learning. They proposed an improved Naive Bayes classifier (INBC) technique and explored the use of genetic algorithms (GAs) for selection of a subset of input features in classification problems. They then carried out a comparison with several other techniques. This sets a comparison of the following algorithms, namely, (i) genetic algorithm with average classification or general classification (GA-AC, GA-C), (ii) C4.5 with pruning, and (iii) INBC with relative frequency or initial probability density (INBC-RF, INBC-IPD), on the real meteorological data in Hong Kong. In their experiments, the daily observations of meteorological data were collected from the Observatory Headquarters and King's Park for training and test purposes, for the period from 1984 to 1992 (Hong Kong Observatory). During this period, they were only interested in extracting data from May to October (for the rainy season) each year. INBC achieved about a 90% accuracy rate on the rain/no-rain (Rain) classification problems. This method also attained reasonable performance on rainfall prediction with three-level depth (Depth 3) and five-level depth (Depth 5), which was around 65%–70%. They used a filter method for feature selection. In general, it is known that a wrapper method performs better than a filter method. In this study, we try to apply a wrapper method to feature selection.

Nandargi and Mulye [15] analyzed the period of 1961–2005 to understand the relationship between the rain and rainy days, mean daily intensity, and seasonal rainfall over the Koyana catchment in India, on monthly, as well as seasonal,

scale. They compared a linear relationship with a logarithmic relationship, in the case of seasonal rainfall versus mean daily intensity.

Routray et al. [16] studied a performance-based comparison of simulations carried out using nudging (NUD) technique and three-dimensional variation (3DVAR) data assimilation system, of a heavy rainfall event that occurred during 25–28 June, 2005, along the west coast of India. In the experiment, after observations using the 3DVAR data assimilation technique, the model was able to simulate better structure of the convective organization, as well as prominent synoptic features associated with the mid-tropospheric cyclones (MTC), than the NUD experiment, and well correlated with the observations.

Kouadio et al. [17] investigated relationships between simultaneous occurrences of distinctive atmospheric easterly wave (EW) signatures that cross the south equatorial Atlantic, intense mesoscale convective systems (lifespan > 2 hours) that propagate westward over the western south equatorial Atlantic, and subsequent strong rainfall episodes (anomaly > 10 mm-day⁻¹) that occur in eastern Northeast Brazil (ENEB). They forecasted rainfall events through real-time monitoring and the simulation of this ocean-atmosphere relationship.

Afandi et al. [2] investigated heavy rainfall events that occurred over Sinai Peninsula and caused flash flood, using the Weather Research and Forecasting (WRF) model. The test results showed that the WRF model was able to capture the heavy rainfall events over different regions of Sinai and predict rainfall in significant consistency with real measurements.

Wang and Huang [18] studied on finding the evidence of self-organized criticality (SOC) for rain datasets in China, by employing the theory and method of SOC. For that reason, they analyzed the long-term rain records of five meteorological stations in Henan, a central province of China. They found that the long-term rain processes in central China exhibit the feature of self-organized criticality.

Hou et al. [19] studied the impact of three-dimensional variation data assimilation (3DVAR) on the prediction of two heavy rainfall events over southern China in June and July. They used two heavy rainfall events: one affecting several provinces in southern China with heavy rain and severe flooding; the other is characterized by nonuniformity and extremely high rainfall rates in localized areas. Their results suggested that the assimilation of all radar, surface, and radiosonde data had a more positive impact on the forecast skill than the assimilation of either type of data only, for the two rainfall events.

As a similar approach to ours, Lee et al. [20] studied feature selection using a genetic algorithm for heavy-rain prediction in South Korea. They used ECMWF (European Centre for Medium-Range Weather Forecasts) weather data collected from 1989 to 2009. They selected five features among 254 weather elements to examine the performance of their model. The five features selected were height, humidity temperature, U-wind, and V-wind. In their study, a heavy-rain criterion is issued only when precipitation during six hours is higher than 70 mm. They used a wrapper-based feature

TABLE 1: Modified weather elements [4, 21].

Index	Contents (original)	Contents (modified)
—	Station number	—
—	Day	—
—	Latitude	—
—	Longitude	—
—	Height	—
1	—	Month (1–12)
2	Mean wind direction for 10 minutes (0.1 deg)	Mean wind direction for 10 minutes (0.1 deg)
3	Mean wind velocity for 10 minutes (0.1 m/s)	Mean wind velocity for 10 minutes (0.1 m/s)
4	Mean temperature for 1 minute (0.1 C)	Mean temperature for 1 minute (0.1 C)
5	Mean humidity for 1 minute (0.1%)	Mean humidity for 1 minute (0.1%)
6	Mean atmospheric pressure for 1 minute (0.1 hPa)	Mean atmospheric pressure for 1 minute (0.1 hPa)
—	Mean sea level pressure for 1 minute (0.1 hPa)	—
7	Accumulated precipitation for 1 hour (0.1 mm)	Accumulated precipitation for 1 hour (0.1 mm)
8	Precipitation sensing (0 or 1)	Precipitation sensing (0 or 1)
9	—	Accumulated precipitation for 3 hours (0.1 mm)
10	—	Accumulated precipitation for 6 hours (0.1 mm)
11	—	Accumulated precipitation for 9 hours (0.1 mm)
12	Accumulated precipitation for 24 hours (0.1 mm)	Accumulated precipitation for 24 hours (0.1 mm)

selection method using a simple genetic algorithm and SVM with RBF kernel as the fitness function. They did not explain errors and incorrectness for their weather data. In this paper, we use the weather data collected from 408 automatic weather stations during the recent four years from 2007 to 2010. Our heavy-rain criterion is exactly that of Korea Meteorological Administration in South Korea, as shown in Section 3. We validate our algorithms with various machine learning techniques, including SVM with different kernels. We also explain and fixed errors and incorrectness for our weather data in Section 2.

The remainder of this paper is organized as follows. In Section 2, we propose data processing and methodology for very short-term heavy rainfall prediction. Section 3 describes the environments of our experiments and analyzes the results. The paper ends with conclusions in Section 4.

2. Data and Methodology

2.1. Dataset. The weather data, which are collected from 408 automatic weather stations during the recent four years from 2007 to 2010, had a considerable number of missing data, erroneous data, and unrelated features. We analyzed the data and corrected the errors. We preprocessed the original data given by KMA, in accordance with Table 1. Some weather elements of the original data had incorrect value, and we replaced the value with a very small one (-10^7). We created several elements, such as *month* (1–12) and *accumulated precipitation for 3, 6, and 9 hours* (0.1 mm), from the original data [21]. We removed or interpolated each day data of the original data, when important weather elements of the day data had very small value. Also, we removed or interpolated new elements, such as *accumulated precipitation for 3, 6, and*

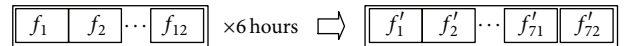


FIGURE 3: Representation with 72 features (accumulated weather factors for six hours).

9 hours, which had incorrect value. We undersampled the weather data that were adjusted for the proportion of heavy-rain against no-heavy-rain to be one in the training set, as shown in Section 2.3.

The new data were generated in two forms: whether or not we applied normalization. The training set, ranging from 2007 to 2008, was generated by undersampling. The validation set, the data for 2009, was used to select an important subset from input features. The selected important features were used for experiments with the test set, the data for 2010. Representation of our GA and DE was composed of 72 features accumulated for the recent six hours, as shown in Figure 3. The symbols f_{1-12} shown in Figure 3 mean modified weather elements in order by index number shown in Table 1. The symbol “—” in Table 1 means (NA not applicable).

2.2. Normalization. The range of each weather element was significantly different (see Table 2), and the test results might rely on the values of a few weather elements. For that reason, we preprocessed the weather data using a normalization method. We calculated the upper bound and lower bound of each weather factor from the original training set. The value of each upper bound and lower bound was converted to 1 and 0, respectively. Equation (1) shows the process for the used normalization. In (1), d means each weather element. The validation set and the test set were normalized, in accordance

TABLE 2: The upper and lower bound ranges of weather data.

Weather elements	Upper bound	Lower bound
Latitude	38.53	32.50
Longitude	131.88	32.50
Height	1673	1.5
Mean wind direction for 10 minutes (0.1 deg)	3600	0
Mean wind velocity for 10 minutes (0.1 m/s)	424	0
Mean temperature for 1 minute (0.1°C)	499	-399
Mean humidity for 1 minute (0.1%)	1000	0
Mean atmospheric pressure for 1 minute (0.1 hPa)	10908	0
Mean sea level pressure for 1 minute (0.1 hPa)	11164	0
Precipitation sensing (0/1)	1	0
Accumulated precipitation for 1 hour (0.1 mm)	1085	0
Accumulated precipitation for 24 hours (0.1 mm)	8040	0

TABLE 3: Heavy rainfall rate.

Year	Heavy-rain (hours)	No-heavy-rain (hours)	Ratio (%)
2007	10.18	8749.82	0.0012
2008	9.71	8774.29	0.0011
2009	19.32	8716.68	0.0022
2010	14.66	8721.35	0.0017

with the ranges in the original training set. *Precipitation sensing* in Table 2 means whether or not it rains:

$$d_{\max} = \max \{d\}, \quad d_{\min} = \min \{d\},$$

$$\tilde{d}_i = \frac{d_i - d_{\min}}{d_{\max} - d_{\min}}. \quad (1)$$

2.3. Sampling. Let l be the frequency of heavy rainfall occurrence in the training set. We randomly choose l among the cases of no-heavy-rain in the training set. Table 3 shows the proportion of heavy-rain to no-heavy-rain every year. On account of the results of Table 3, we preprocessed our data using this method called undersampling. We adjusted the proportion of heavy rainfall against the other to be one, as shown in Figure 4 and Pseudocode 1.

Table 4 shows ETS for prediction after 3 hours and the effect of undersampling [22] and normalization for 3 randomly chosen stations. The tests without undersampling showed a low equitable threat score (ETS) and required too long a computation time. In tests without undersampling, the computation time took 3, 721 minutes in k -NN and 3, 940 minutes in k -VNN (see Appendix B), the “reached max number of iterations” error was raised in SVM with polynomial kernel (see Appendix C), and a and b of ETS were zero. In tests with undersampling, the computation time took around 329 seconds in k -NN, 349 seconds in k -VNN, and 506 seconds in SVM with polynomial kernel. The test results

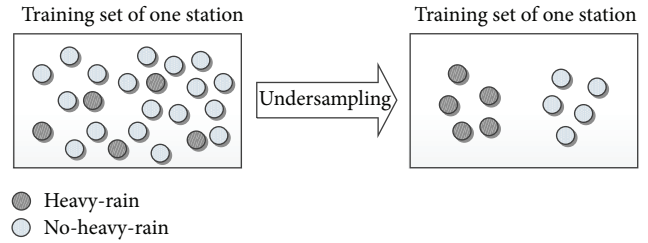


FIGURE 4: Example of our undersampling process.

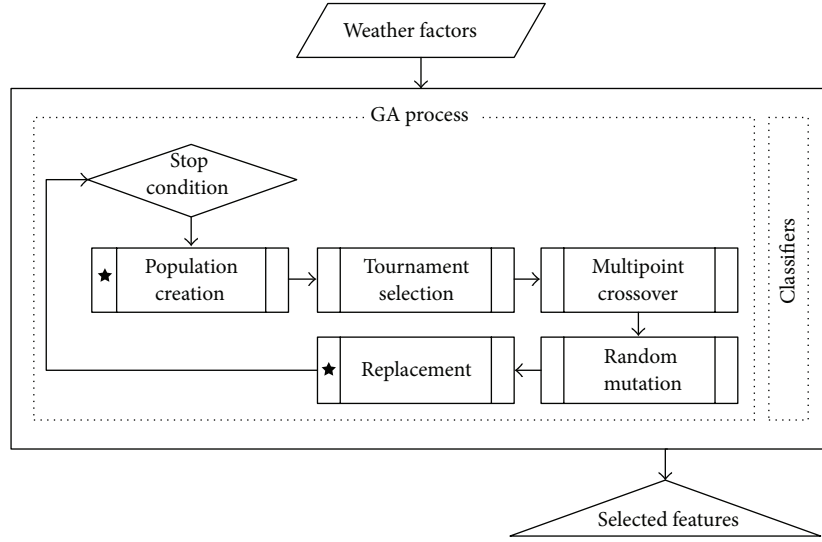
with normalization showed about 10 times higher, than those without normalization.

2.4. Genetic-Algorithm-Based Feature Selection. Pseudocode 2 shows the pseudocode of a typical genetic algorithm [23]. In this figure, if we define that n is the count of solutions in the population set, we create n new solutions in a random way. The evolution starts from the population of completely random individuals, and the fitness of the whole population is determined. Each generation consists of several operations, such as selection, crossover, mutation, and replacement. Some individuals in the current population are replaced with new individuals to form a new population. Finally, this generational process is repeated, until a termination condition has been reached. In a typical GA, the whole number of individuals in a population and the number of reproduced individuals are fixed at n and k , respectively. The percentage of individuals to copy to the new generation is defined as the ratio of the number of new individuals to the size of the parent population, k/n , which we called “generation gap” [24]. If the gap is close to $1/n$, the GA is called a steady-state GA.

We selected important features, using the wrapper methods that used the inductive algorithm to estimate the value of a given subset. The selected feature subset is the best individual among results of the experiment with the validation set. The experimental results in the test set with the selected features showed better performance than those using all features.

The steps of the GA used are described in Box 1. All steps will be iterated, until the stop condition (the number of generations) is satisfied. Figure 5 shows the flow diagram of our steady-state GA.

2.5. Differential-Evolution-Based Feature Selection. Khushaba et al. [25, 26] proposed a differential-evolution-based feature selection (DEFS) technique which is shown schematically in Figure 6. The first step in the algorithm is to generate new population vectors from the original population. A new mutant vector is formed by first selecting two random vectors, then performing a weighted difference, and adding the result to a third random (base) vector. The mutant vector is then crossed with the original vector that occupies that position in the original matrix. The result of this operation is called a trial vector. The corresponding position in the new population will contain either the trial vector (or its corrected version) or the original target vector depending on which one of those achieved a higher fitness (classification accuracy). Due to the



★ This step requires a classifier process

FIGURE 5: Flow diagram of the proposed steady-state GA.

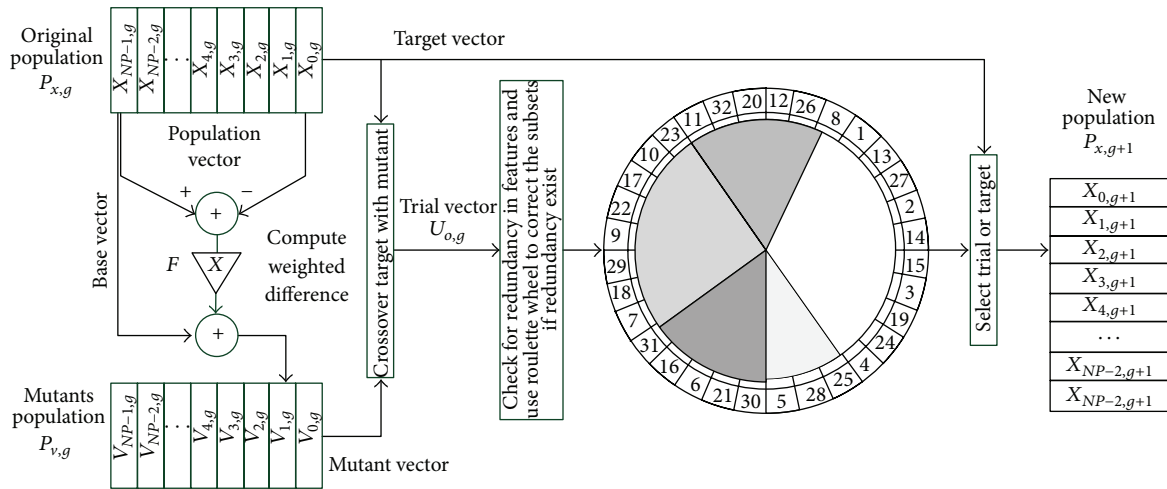


FIGURE 6: The DEFS algorithm [25, 26].

fact that a real number optimizer is being used, nothing will prevent two dimensions from settling at the same feature coordinates. In order to overcome such a problem, they proposed to employ feature distribution factors to replace duplicated features. A roulette wheel weighting scheme is utilized. In this scheme, a cost weighting is implemented, in which the probabilities of individual features are calculated from the distribution factors associated with each feature. The distribution factor of feature f_i is given by the following equation:

$$FD_i = a_1 * \left(\frac{PD_i}{PD_i + ND_i} \right) + a_2 * \left(1 - \frac{PD_i + ND_i}{\epsilon + \max(PD_i + ND_i)} \right), \quad (2)$$

where a_1, a_2 are constants and ϵ is a small factor to avoid division by zero. PD_i is the positive distribution factor that is computed from the subsets that achieved an accuracy that is higher than the average accuracy of the whole subsets. ND_i is the negative distribution factor that is computed from the subsets that achieved an accuracy that is lower than the average accuracy of the whole subsets. This is shown schematically in Figure 7, with the light gray region being the region of elements achieving less error than the average error values and the dark gray being the region with elements achieving higher error rates than the average. The rationale behind (2) is to replace the replicated parts of the trial vectors according to two factors. The $PD_i/(PD_i+ND_i)$ factor indicates the degree to which f_i contributes to forming good subsets. On the other hand, the second term in (2) aims at favoring exploration, where this term will be close to 1, if the overall usage of a specific feature is very low.

TABLE 4: Effect of undersampling (sampled 3 stations, prediction after 3 hours).

	w/o undersampling			w/undersampling		
	k -NN (min)	k -VNN (min)	SVM (min)	k -NN (sec)	k -VNN (sec)	SVM (sec)
w/o normalization	0.000 (3323)	0.000 (3760)	N/A (>10000000)	0.003 (301)	0.014 (329)	0.024 (285)
w/normalization	0.000 (3721)	0.000 (3940)	N/A (>10000000)	0.032 (329)	0.094 (349)	0.267 (506)

```

// A: set of heavy-rain cases in training set
// B: set of no-heavy-rain cases in training set
// R: set of no-heavy-rain cases sampled from B, that is,  $R \subseteq B$ 
// T: undersampled training set

 $l \leftarrow$  the number of heavy-rain cases, that is,  $|A|$ ;
initialize R to be empty;
while ( $l > 0$ )
  randomly choose one value from B;
  if the value is not in R, then
    add the value to R;
     $l \leftarrow l - 1$ ;
  end if
end while
 $T \leftarrow$  the union of A and R;
Return T;

```

PSEUDOCODE 1: A pseudocode of our undersampling process.

```

Create an initial population of size  $n$ ;
repeat
  for  $i = 1$  to  $k$ 
    choose  $p_1$  and  $p_2$  from the population;
    offspring $_i$  = crossover( $p_1$ ,  $p_2$ );
    offspring $_i$  = mutation(offspring $_i$ );
  end for
  replace(population, [offspring $_1$ , offspring $_2$ , ..., offspring $_k$ ]);
until (stopping condition);
return the best solution;

```

PSEUDOCODE 2: The pseudocode of a genetic algorithm.

3. Experimental Results

We preprocessed the original weather data. Several weather elements are added or removed, as shown in Table 1. We undersampled and normalized the modified weather data. Each hourly record of the data consists of twelve weather elements, and representation was made up of the latest six hourly records, 72 features, as shown in Figure 3. We extracted a feature subset using the validation set and used the feature subset to do experiments with the test set.

The observation area has 408 automatic weather stations in the southern part of the Korean peninsula. The prediction time is from one hour to six hours. We adopted GA and DE among the evolutionary algorithms. SVM, k -VNN, and k -NN are used as discriminant functions. Table 5 shows the parameters of a steady-state GA and DE, respectively. LibSVM [27] is

adopted as a library of SVM, and we set SVM type, one of the SVM parameters, as *C_SVC* that regularizes support vector classification, and the kernel functions used are *polynomial*, *linear*, and *precomputed*. We set k to be 3 in our experiments.

In South Korea, a heavy-rain advisory is issued when precipitation during six hours is higher than 70 mm or precipitation during 12 hours is higher than 110 mm. A heavy-rain warning is issued when precipitation during 6 hours is higher than 110 mm, or precipitation during 12 hours is higher than 180 mm. We preprocessed the weather data using this criterion. To select the main features, we adopted a wrapper method, which uses classifier itself in feature evaluation differently from a filter method.

An automatic weather station (AWS) [28] is an automated version of the traditional weather station, either to save human labor or to enable measurements from remote

- (1) Population Initialization: generate m random solutions.
- (2) Selection: a number Tour of individuals is chosen randomly from the population, and the best individual from this group is selected as parent.
- (3) Crossover: create an offspring by the genetic recombination of Parent₁ and Parent₂.
- (4) Mutation: change each gene of the offspring at the rate of 5 percent.
- (5) Replacement: if the offspring is superior to the worst individual of population, replace the worst one with the offspring.

Box 1: Steps of the used GA.

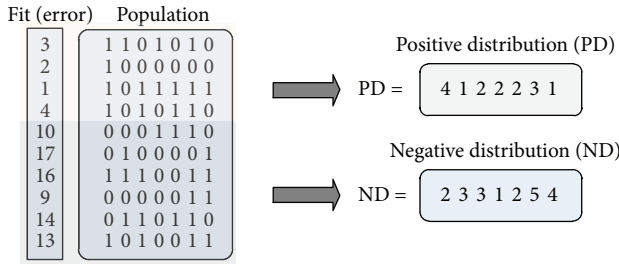


FIGURE 7: The feature distribution factors [25, 26].

areas. An automatic weather station will typically consist of a weather-proof enclosure, containing the data logger, rechargeable battery, telemetry (optional), and the meteorological sensors, with an attached solar panel or wind turbine and mounted upon a mast. The specific configuration may vary, due to the purpose of the system. In Table 6, Fc and Obs are abbreviations for forecast and observed, respectively. The following is a measure for evaluating precipitation forecast skill:

ETS (equitable threat score)

$$= \frac{(a - a_r)}{(a + b + c - a_r)}, \quad a_r = \frac{(a + b)(a + c)}{n},$$

FBI (frequency bias index) = $(a + b)(a + c)$,

$$PC \text{ (proportion correct)} = \frac{(a + d)}{n}, \quad (3)$$

$$POD \text{ (probability of detection)} = \frac{a}{(a + c)},$$

$$PAG \text{ (post-agreement)} = \frac{a}{(a + b)}.$$

These experiments were conducted using LibSVM [27] on an Intel Core2 duo quad core 3.0 GHz PC. Each run of GA took about 201 seconds in SVM test with normalization and about 202 seconds without normalization; it took about 126 seconds in k -NN test with normalization and about 171 seconds without normalization; it took about 135 seconds in k -VNN test with normalization and about 185 seconds without normalization.

Each run of DE took about 6 seconds in SVM test with normalization and about 5 seconds without normalization;

TABLE 5: Parameters in GA/DE.

GA parameters	
Fitness function	k -NN ($k = 3$), k -VNN ($k = 3$), SVM (type: C_SVC , kernel function: <i>polynomial</i> , <i>linear</i> , and <i>precomputed</i>) [27]
Encoding	Binary (72 dimensions)
No. of populations	20
No. of generations	100
Selection	Tournament selection
Crossover	Multipoint crossover (3 points)
Mutation	Genewise mutation ($P = 0.005$)
Replacement	If an offspring is superior to the worst individual in the population, we replace it with the worst one
DE parameters	
Fitness function	k -NN ($k = 3$), k -VNN ($k = 3$), SVM (type: C_SVC , kernel function: <i>polynomial</i>)
Encoding	Real number (23 dimensions)
No. of populations	20
No. of generations	100
Crossover rate	0.03
FVal	0.05
Replacement	If an offspring is superior to the parent in the population, we replace it with the parent

it took about 5 seconds in k -NN test with normalization and about 4 seconds without normalization; it took about 5 seconds in k -VNN test with normalization and about 4 seconds without normalization.

The heavy-rain events, which meet the criterion of heavy rainfall, consist of a consecutive time interval, which has a beginning time and an end time. The *coming event* is to discern whether or not it is a heavy rain on the beginning time. For each hour from the beginning time to the end time, discerning whether or not it is a heavy rain means the *whole process*. We defined CE and WP to be forecasting the *coming event* and the *whole process* of heavy rainfall, respectively.

Table 7 shows the experimental results for GA and DE. Overall, GA was about 1.42 and 1.49 times better than DE in CE and WP predictions, respectively. In DE experiments, SVM and k -VNN were about 2.11 and 1.10 times better than k -NN in CE prediction, respectively. SVM and k -VNN were about 2.48 and 1.08 times better than k -NN in WP prediction,

TABLE 6: Contingency table.

Forecast	Event		Marginal total
	Yes	No	
Yes	Hit (a)	False alarm (b)	Fc Yes ($a + b$)
No	Miss (c)	Correct nonevent (d)	Fc No ($c + d$)
Marginal total	Obs Yes ($a + c$)	Obs No ($b + d$)	Sum total ($a + b + c + d = n$)

TABLE 7: Experimental results (1–6 hours) by ETS.

Prediction type	Prediction hour											
	1		2		3		4		5		6	
	CE	WP	CE	WP	CE	WP	CE	WP	CE	WP	CE	WP
DE												
k -NN	0.096	0.183	0.062	0.127	0.043	0.093	0.026	0.059	0.020	0.049	0.014	0.035
k -VNN	0.098	0.187	0.073	0.147	0.049	0.104	0.030	0.069	0.021	0.048	0.015	0.037
SVM (<i>polynomial</i>)	0.192	0.383	0.139	0.320	0.140	0.329	0.090	0.238	0.027	0.105	0.005	0.019
GA												
k -NN	0.070	0.265	0.068	0.212	0.056	0.160	0.035	0.105	0.025	0.078	0.009	0.044
k -VNN	0.179	0.314	0.152	0.279	0.113	0.230	0.084	0.184	0.047	0.117	0.029	0.078
SVM												
<i>Polynomial</i>	0.276	0.516	0.239	0.481	0.160	0.373	0.102	0.271	0.040	0.148	0.008	0.046
<i>Linear</i>	0.043	0.095	0.096	0.196	0.127	0.200	0.083	0.150	0.152	0.240	0.102	0.173
<i>Precomputed</i>	0.048	0.102	0.055	0.126	0.040	0.086	0.079	0.157	0.048	0.090	0.040	0.074

CE: forecasting the *coming event* of heavy rainfall. WP: forecasting the *whole process* of heavy rainfall.

respectively. In GA experiments, SVM with polynomial kernel showed better performance than that with linear or precomputed kernel on average. SVM with polynomial kernel and k -VNN were about 2.62 and 2.39 times better than k -NN in CE prediction, respectively. SVM with polynomial kernel and k -VNN were about 2.01 and 1.49 times better than k -NN in WP prediction, respectively. As the prediction time is longer, ETS shows a steady downward curve. SVM with polynomial kernel shows the best ETS among GA test results. Figure 8 visually compares CE and WP results in GA experiments.

Consequently, SVM showed the highest performance among our experiments. k -VNN showed that the degree of genes' correlation had significantly effects on the test results, in comparison with k -NN. Tables 8, 9, 10, and 11 show detailed SVM (with polynomial kernel) test results for GA and DE.

We selected the important features using the wrapper methods using the inductive algorithm to estimate the value of a given set. All features consist of accumulated weather factors for six hours, as shown in Figure 3. The selected feature subset is the best individual among the experimental results, using the validation set. Figure 9 shows the frequency for the selected features after one hour to six hours. The test results using the selected features were higher than those using all features. We define a feature as f . The derived features from the statistical analysis, which has a 95 percent confidence interval, were the numbers $f_3, f_7, f_8, f_{10}, f_{12}, f_{19}, f_{20}, f_{21}, f_{22}, f_{23}, f_{24}, f_{31}, f_{32}, f_{36}, f_{43}, f_{44}, f_{46}, f_{48}, f_{55}, f_{56}$, and

f_{68} . The main seven features selected were the numbers $f_8, f_{12}, f_{20}, f_{24}, f_{32}, f_{44}$, and f_{56} and were evenly used by each prediction hour. These features were *precipitation sensing* and *accumulated precipitation for 24 hours*.

We compared the heavy rainfall prediction test results of GA and DE, as shown in Table 7. The results showed that GA was significantly better than DE. Figure 10 shows precipitation maps for GA SVM test results with normalization and undersampling, from one to six hours. The higher ETS is depicted in the map in the darker blue color. The numbers of automatic weather stations by prediction hours are 105, 205, 231, 245, 223, and 182, in order from one to six hours, respectively. The reasons for the differential numbers of automatic weather stations by prediction hours are as follows. First, we undersampled the weather data by adjusting the sampling sizes of no-heavy-rain to be equal to the size of heavy-rain in the training set, as shown in Section 2.3. Second, we excluded the AWS number in which the record number of the training set is lower than three. Third, we excluded the AWS in which *hit* and *false alarm* are 0 from the validation experimental results. Finally, we excluded the AWS in which *hit*, *false alarm*, and *miss* are 0 from the test experimental results.

The weather data collected from automatic weather stations during the recent four years had a lot of missing data and erroneous data. Furthermore, our test required more than three valid records in the training set. For that reason, the number of usable automatic weather stations was the

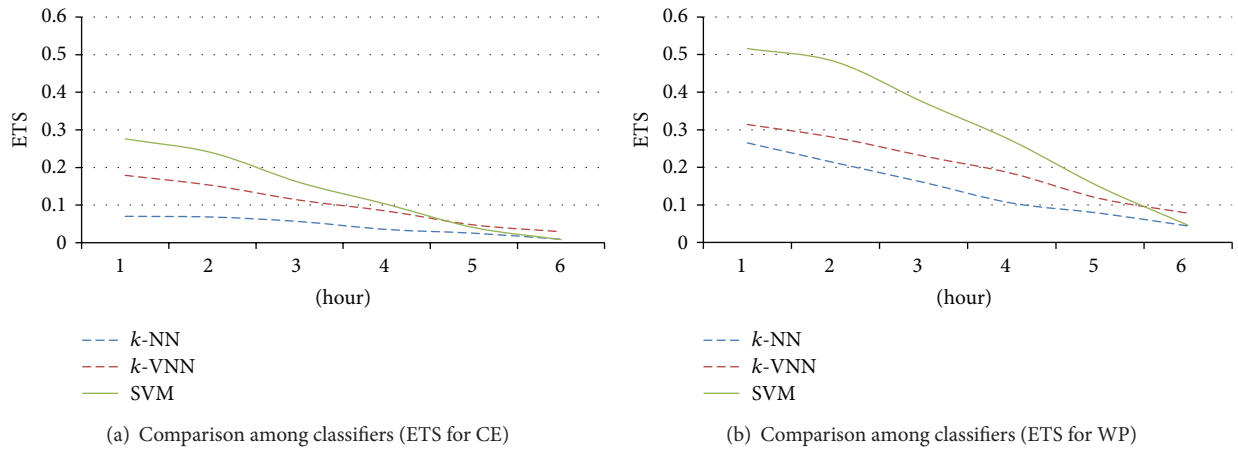


FIGURE 8: Experimental results for GA from 1 to 6 hours.

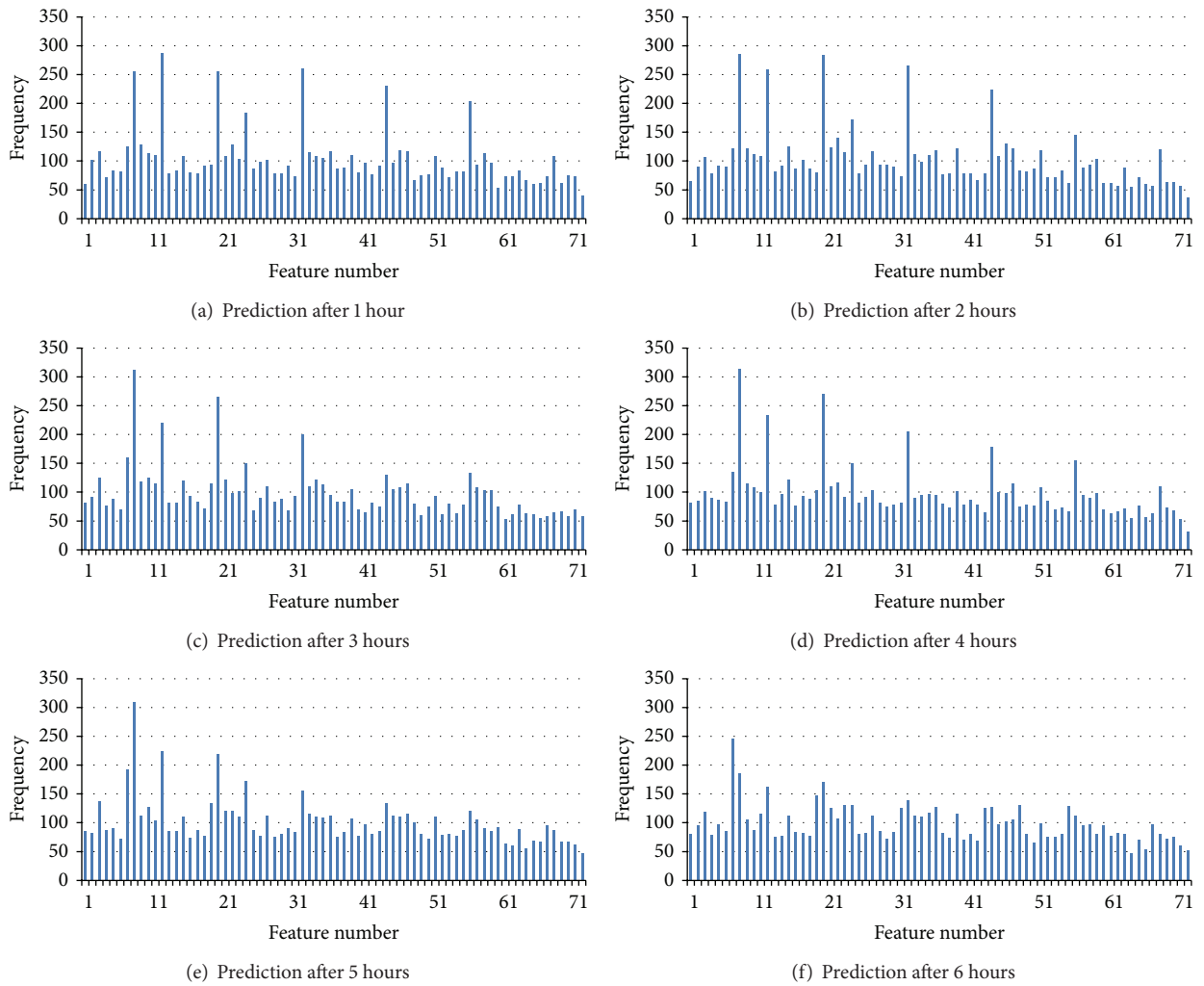


FIGURE 9: Frequency for selected features after from 1 to 6 hours.

TABLE 8: Results of DE with SVM from 1 to 6 hours (CE).

Hour	ETS	FBI	PC	POD	PAG	Hit	False alarm	Miss	Correct nonevent	No. of AWSs
1	0.192	4.116	0.994	0.627	0.340	11.619	41.305	7.067	8160.305	105
2	0.139	5.108	0.994	0.531	0.332	8.737	45.332	7.902	8139.571	205
3	0.140	5.615	0.994	0.512	0.301	8.238	41.710	8.338	8102.411	231
4	0.090	9.517	0.990	0.486	0.264	7.878	69.261	9.008	8048.094	245
5	0.027	30.133	0.977	0.419	0.116	5.707	183.378	8.053	7942.960	223
6	0.005	79.798	0.901	0.589	0.041	5.484	817.126	3.874	7315.505	182

TABLE 9: Results of DE with SVM from 1 to 6 hours (WP).

Hour	ETS	FBI	PC	POD	PAG	Hit	False alarm	Miss	Correct nonevent	No. of AWSs
1	0.383	2.558	0.994	0.813	0.535	30.295	41.305	7.067	8160.305	105
2	0.320	3.055	0.994	0.766	0.538	25.356	45.332	7.902	8139.571	205
3	0.329	3.308	0.994	0.756	0.512	24.814	41.710	8.338	8102.411	231
4	0.238	5.252	0.990	0.744	0.475	24.820	69.261	9.008	8048.094	245
5	0.105	13.148	0.977	0.741	0.312	23.156	183.378	8.053	7942.960	223
6	0.019	31.885	0.901	0.846	0.144	23.341	817.126	3.874	7315.505	182

TABLE 10: Results of GA with SVM from 1 to 6 hours (CE).

Hour	ETS	FBI	PC	POD	PAG	Hit	False alarm	Miss	Correct nonevent	No. of AWSs
1	0.276	2.168	0.997	0.589	0.403	10.581	19.524	8.105	8182.086	105
2	0.239	2.398	0.997	0.529	0.383	8.771	19.824	7.868	8165.078	205
3	0.160	3.613	0.995	0.463	0.316	8.000	32.918	8.576	8111.203	231
4	0.102	6.421	0.992	0.417	0.291	7.747	57.514	9.139	8059.841	245
5	0.040	20.543	0.984	0.397	0.117	5.695	122.857	8.126	8007.287	223
6	0.008	66.609	0.944	0.420	0.025	4.192	437.291	5.984	7546.929	182

TABLE 11: Results of GA with SVM from 1 to 6 hours (WP).

Hour	ETS	FBI	PC	POD	PAG	Hit	False alarm	Miss	Correct nonevent	No. of AWSs
1	0.516	1.577	0.997	0.797	0.622	29.686	19.524	8.105	8182.086	105
2	0.481	1.671	0.997	0.766	0.610	25.805	19.824	7.868	8165.078	205
3	0.373	2.274	0.995	0.735	0.561	24.970	32.918	8.576	8111.203	231
4	0.271	3.685	0.992	0.713	0.540	25.069	57.514	9.139	8059.841	245
5	0.148	10.285	0.984	0.733	0.341	23.363	122.857	8.126	8007.287	223
6	0.046	27.701	0.944	0.786	0.165	23.154	437.291	5.984	7546.929	182

lowest in the prediction after one hour and increased as the prediction time became longer.

4. Conclusion

In this paper, we realized the difficulty, necessity, and significance of very short-term heavy rainfall forecasting. We used various machine learning techniques, such as SVM, k -NN, and k -VNN based on GA and DE, to forecast heavy rainfall after from one hour to six hours. The results of GA were significantly better than those of DE. SVM with polynomial kernel among various classifiers in our GA experiments showed the best results on average. A validation set was used

to select the important features, and the selected features were used to predict very short-term heavy rainfall. We derived 20 features from the statistical analysis, which has a 95 percent confidence interval. The main features selected were *precipitation sensing* and *accumulated precipitation for 24 hours*.

In future work, we will preprocess the weather data by various methods, such as representation learning, cyclic loess, contrast, and quantile normalization algorithms. Also, we will apply other machine learning techniques, such as statistical relational learning, multilinear subspace learning, and association rule learning. As more appropriate parameters are applied to the evolutionary algorithm or machine

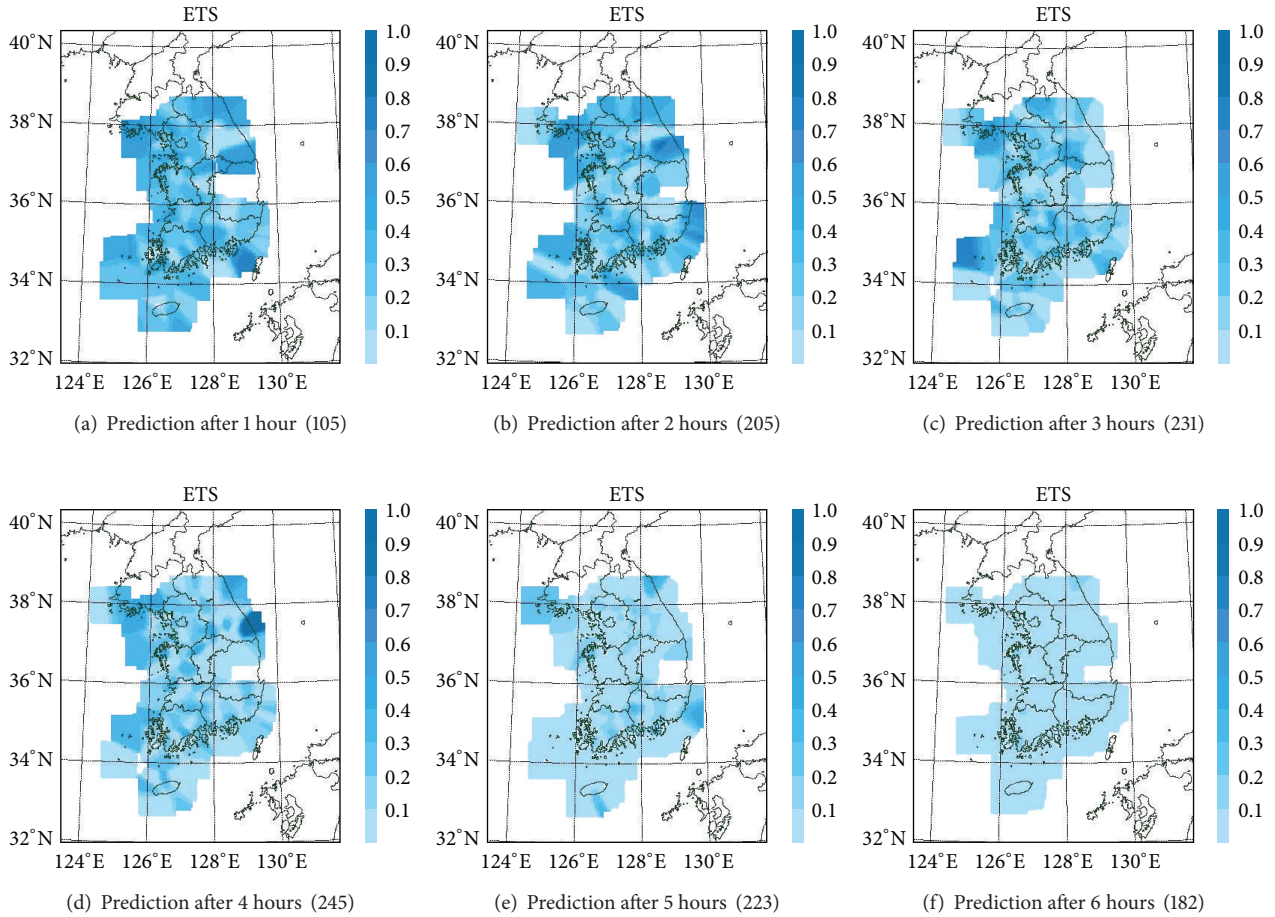


FIGURE 10: Individual maps, with AWS in blue dots, for GA heavy rainfall prediction after from 1 to 6 hours (ETS).

learning techniques, we expect to get better results. We have validated our algorithms with AWS data; however, it would be interesting to examine the performance with, for example, satellite data as another future work.

Appendices

A. Spatial and Temporal Distribution of Heavy Rainfall over South Korea

We calculated the rainfall duration, which meets the criterion of heavy rainfall, from each automatic weather station for the period from 2007 to 2010. We divided the rainfall duration by 100 and let the result be depicted in the map. Figure 11 shows the distribution of heavy rainfall for the whole seasons. Figure 12 shows the distribution of heavy rainfall by seasons. Most heavy rainfalls have been concentrated in summer, and they have a wide precipitation range regionally. Also, their frequencies are quite different from region to region.

B. k -Nearest Neighbors Classifier

In pattern recognition, the k -nearest neighbors algorithm (k -NN) [29] is a method for classifying objects based on the

closest training examples in the feature space. k -NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally, and all computation is deferred until classification. The k -NN algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k -nearest neighbors (k is a positive integer, typically small). The k -NN classifier is commonly based on the Euclidean distance between a testing sample and the specified training samples.

Golub et al. [30] developed a procedure that uses a fixed subset of informative genes and makes a prediction based on the expression level of these genes in a new sample. Each informative gene casts a weighted vote for one of the classes, with the magnitude of each vote dependent on the expression level in the new sample, and the degree of that gene's correlation with the class distinction in their class predictor. We made a variant k -nearest neighbors algorithm (k -VNN) that the degree (ρ') of genes' correlation was applied to a majority vote of its neighbors. Box 2 shows the equation calculating correlation between feature and class. In Box 2, g means a feature (i.e., a weather element) and C means a class (i.e., heavy-rain or no-heavy-rain). The test results of k -VNN were better than those of k -NN. We set k to be 3 in our

$$\begin{aligned} \mu_1(g) &\leftarrow \text{average of } g \text{ for the samples in 1} \\ \mu_0(g) &\leftarrow \text{average of } g \text{ for the samples in 0} \\ \sigma_1(g) &\leftarrow \text{standard deviation of } g \text{ for the samples in 1} \\ \sigma_0(g) &\leftarrow \text{standard deviation of } g \text{ for the samples in 0} \\ \rho'(g, C) &\leftarrow (\mu_1(g) - \mu_0(g)) / (\sigma_1(g) + \sigma_0(g)) \end{aligned}$$

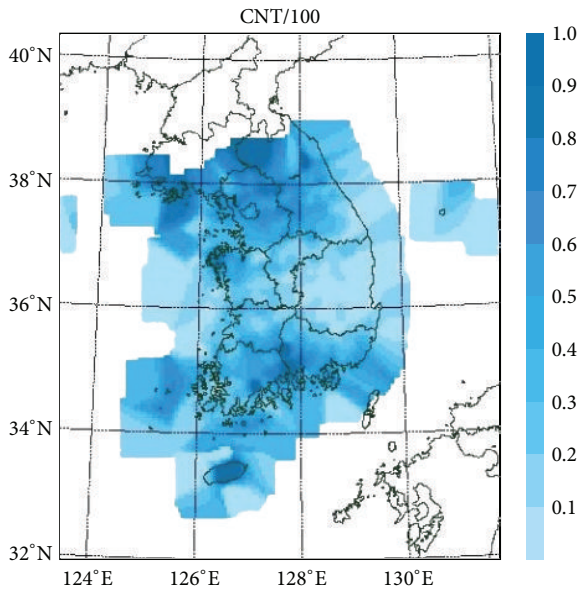
Box 2: Correlation ρ' between feature g and class C (0 or 1) [30, 31].

FIGURE 11: The distribution of heavy rainfall for the whole seasons (2007–2010).

experiments because it is expected that the classifier will show low performance if k is just 1 and it will take a long computing time when k is 5 or more.

C. Support Vector Machine

Support vector machines (SVM) [32] are a set of related supervised learning methods that analyze data and recognize patterns and are used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a nonprobabilistic binary linear classifier. Since an SVM is a classifier, it is then given a set of training examples, each marked as belonging to one of two categories, and an SVM training algorithm builds a model that assigns new examples into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category, based on which side of the gap they fall on.

D. Evolutionary Computation

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution, and this heuristic is routinely used to generate useful solutions to optimization and search problems [33]. In the process of a typical genetic algorithm, the evolution starts from the population of completely random individuals, and the fitness of the whole population is determined. Each generation consists of several operations, such as selection, crossover, mutation, and replacement. Some individuals in the current population are replaced with new individuals to form a new population. Finally, this generational process is repeated, until a termination condition has been reached.

Differential evolution (DE) is an evolutionary (direct-search) algorithm, which has been mainly used to solve optimization problems. DE shares similarities with traditional evolutionary algorithms. However, it does not use binary encoding as a simple genetic algorithm, and it does not use a probability density function to self-adapt its parameters as an evolution strategy. Instead, DE performs mutation, based on the distribution of the solutions in the current population. In this way, search directions and possible step sizes depend on the location of the individuals selected to calculate the mutation values [34].

E. Differences between Adopted Methods

In applied mathematics and theoretical computer science, combinatorial optimization is a topic that consists of finding an optimal object from a finite set of objects. In many such problems, exhaustive search is not feasible. It operates on the domain of those optimization problems, in which the set of feasible solutions is discrete or can be reduced to discrete, and in which the goal is to find the best solution [33].

Feature selection is a problem to get a subset among all features, and it is a kind of combinatorial optimization. Genetic algorithms (GAs) and differential evolutions (DEs) use a random element within an algorithm for optimization or combinatorial optimization, and they are typically used to solve the problems of combinatorial optimization such as feature selection, as in this paper.

Machine learning techniques include a number of statistical methods for handling classification and regression. Machine learning mainly focuses on prediction, based on known properties learned from the training data [33]. It is not easy to use general machine learning techniques for feature selection. In this paper, machine learning techniques

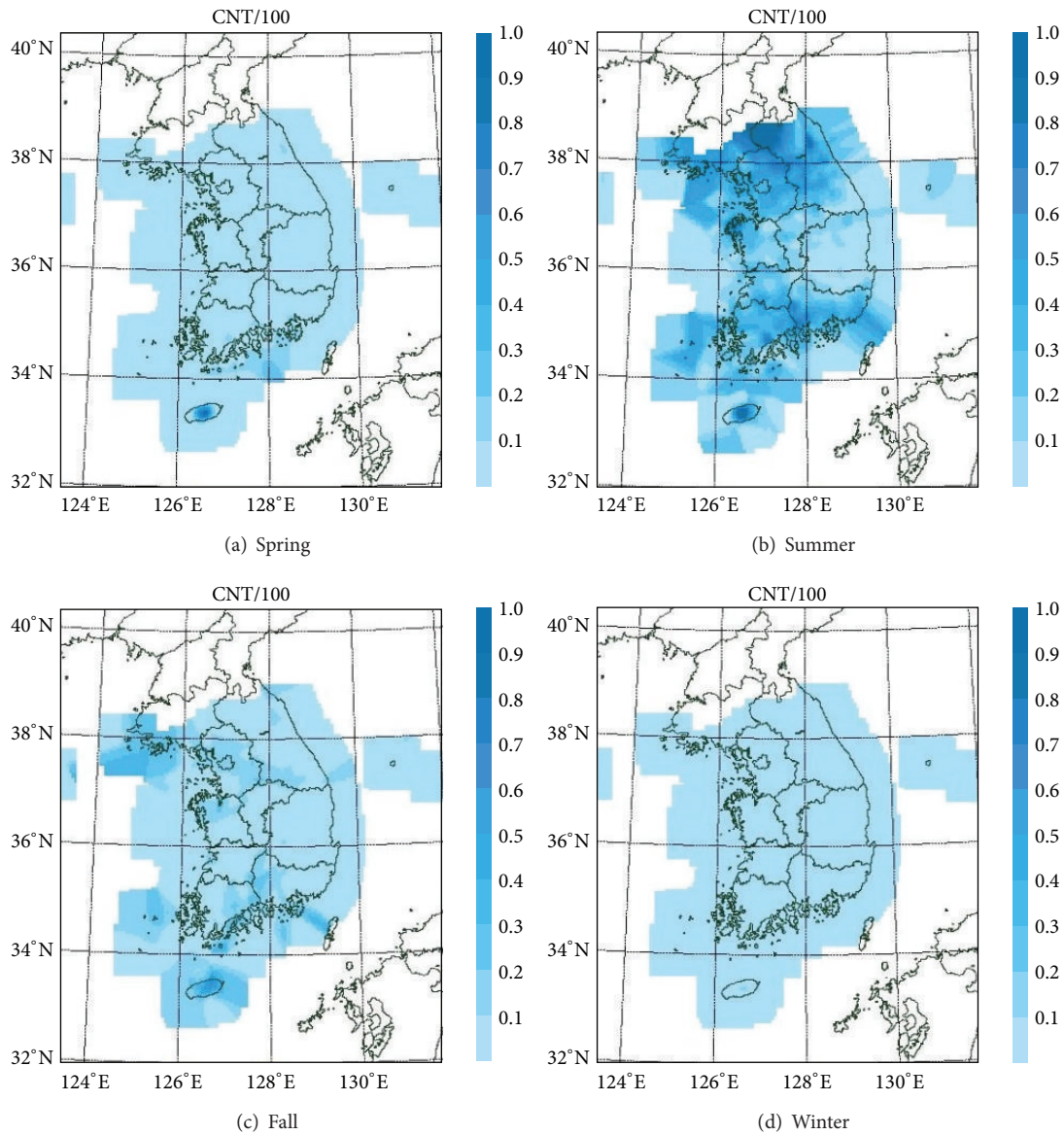


FIGURE 12: The distribution of heavy rainfall by seasons (2007–2010).

were used for classification. GA and DE could be used for regression, but they have a weakness in handling regression because these algorithms will take longer computing time than other regression algorithms.

F. Detailed Statistics of Experimental Results

Tables 8–11 show SVM (with polynomial kernel) test results for GA and DE. As shown in the contingency Table 6, the test results show ETS and other scores. We defined CE and WP to be forecasting the coming event and the whole process of heavy rainfall, respectively. The test results include the number of used automatic weather stations by each prediction hour, and the number of those is equally set, in the same prediction hour of each experiment. As a result, GA was considerably superior to DE.

Conflict of Interests

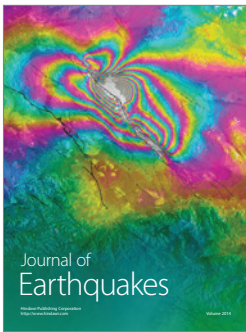
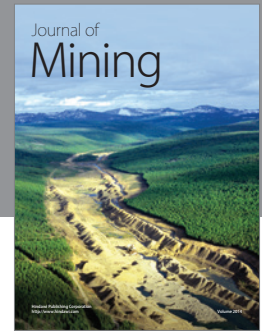
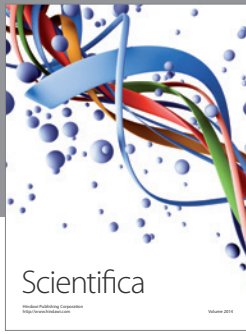
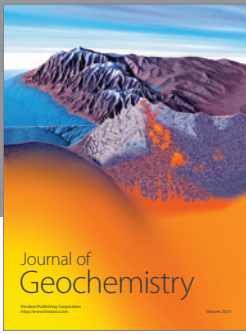
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

A preliminary version of this paper appeared in the Proceedings of the International Conference on Convergence and Hybrid Information Technology, pp. 312–322, 2012. The authors would like to thank Mr. Seung-Hyun Moon for his valuable suggestions in improving this paper. The present research has been conducted by the Research Grant of Kwangwoon University in 2014. This work was supported by the Advanced Research on Meteorological Sciences, through the National Institute of Meteorological Research of Korea, in 2013 (NIMR-2012-B-1).

References

- [1] J. Bushey, "The Changma," <http://www.theweatherprediction.com/weatherpapers/007>.
- [2] G. E. Afandi, M. Mostafa, and F. E. Hussieny, "Heavy rainfall simulation over Sinai peninsula using the weather research and forecasting model," *International Journal of Atmospheric Sciences*, vol. 2013, Article ID 241050, 11 pages, 2013.
- [3] J. H. Seo and Y. H. Kim, "A survey on rainfall forecast algorithms based on machine learning technique," in *Proceedings of the KIIS Fall Conference*, vol. 21, no. 2, pp. 218–221, 2011, (Korean).
- [4] Korea Meteorological Administration, <http://www.kma.go.kr>.
- [5] M. N. French, W. F. Krajewski, and R. R. Cuykendall, "Rainfall forecasting in space and time using a neural network," *Journal of Hydrology*, vol. 137, no. 1–4, pp. 1–31, 1992.
- [6] E. Toth, A. Brath, and A. Montanari, "Comparison of short-term rainfall prediction models for real-time flood forecasting," *Journal of Hydrology*, vol. 239, no. 1–4, pp. 132–147, 2000.
- [7] S. J. Burian, S. R. Durrans, S. J. Nix, and R. E. Pitt, "Training artificial neural networks to perform rainfall disaggregation," *Journal of Hydrologic Engineering*, vol. 6, no. 1, pp. 43–51, 2001.
- [8] M. C. Valverde Ramírez, H. F. de Campos Velho, and N. J. Ferreira, "Artificial neural network technique for rainfall forecasting applied to the São Paulo region," *Journal of Hydrology*, vol. 301, no. 1–4, pp. 146–162, 2005.
- [9] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi, "An artificial neural network model for rainfall forecasting in Bangkok, Thailand," *Hydrology and Earth System Sciences*, vol. 13, no. 8, pp. 1413–1425, 2009.
- [10] V. M. Krasnopolsky and Y. Lin, "A neural network nonlinear multimodel ensemble to improve precipitation forecasts over continental US," *Advances in Meteorology*, vol. 2012, Article ID 649450, 11 pages, 2012.
- [11] L. Ingsrisawang, S. Ingsriswang, S. Somchit, P. Aungsuratana, and W. Khantiyanan, "Machine learning techniques for short-term rain forecasting system in the northeastern part of Thailand," in *Proceedings of the World Academy of Science, Engineering and Technology*, vol. 31, pp. 248–253, 2008.
- [12] W.-C. Hong, "Rainfall forecasting by technological machine learning models," *Applied Mathematics and Computation*, vol. 200, no. 1, pp. 41–57, 2008.
- [13] C. M. Kishitwal, S. Basu, F. Patadia, and P. K. Thapliyal, "Forecasting summer rainfall over India using genetic algorithm," *Geophysical Research Letters*, vol. 30, no. 23, pp. 1–9, 2003.
- [14] J. N. K. Liu, B. N. L. Li, and T. S. Dillon, "An improved Naïve Bayesian classifier technique coupled with a novel input solution method," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 31, no. 2, pp. 249–256, 2001.
- [15] S. Nandargi and S. S. Mulye, "Relationships between rainy days, mean daily intensity, and seasonal rainfall over the koyna catchment during 1961–2005," *The Scientific World Journal*, vol. 2012, Article ID 894313, 10 pages, 2012.
- [16] A. Routray, K. K. Osuri, and M. A. Kulkarni, "A comparative study on performance of analysis nudging and 3DVAR in simulation of a heavy rainfall event using WRF modeling system," *ISRN Meteorology*, vol. 2012, no. 21, Article ID 523942, 2012.
- [17] Y. K. Kouadio, J. Servain, L. A. T. Machado, and C. A. D. Lentini, "Heavy rainfall episodes in the eastern northeast Brazil linked to large-scale ocean-atmosphere conditions in the tropical Atlantic," *Advances in Meteorology*, vol. 2012, Article ID 369567, 16 pages, 2012.
- [18] Z. Wang and C. Huang, "Self-organized criticality of rainfall in central China," *Advances in Meteorology*, vol. 2012, Article ID 203682, 8 pages, 2012.
- [19] T. Hou, F. Kong, X. Chen, and H. Lei, "Impact of 3DVAR data assimilation on the prediction of heavy rainfall over southern China," *Advances in Meteorology*, vol. 2013, Article ID 129642, 17 pages, 2013.
- [20] H. D. Lee, S. W. Lee, J. K. Kim, and J. H. Lee, "Feature selection for heavy rain prediction using genetic algorithms," in *Proceedings of the Joint 6th International Conference on Soft Computing and Intelligent Systems and 13th International Symposium on Advanced Intelligent Systems (SCIS-ISIS '12)*, pp. 830–833, 2012.
- [21] J. H. Seo and Y. H. Kim, "Genetic feature selection for very short-term heavy rainfall prediction," in *Proceedings of the International Conference on Convergence and Hybrid Information Technology*, vol. 7425 of *Lecture Notes in Computer Science*, pp. 312–322, 2012.
- [22] N. V. Chawla, "Data mining for imbalanced datasets: an overview," *Data Mining and Knowledge Discovery Handbook*, vol. 5, pp. 853–867, 2006.
- [23] Y.-S. Choi and B.-R. Moon, "Feature selection in genetic fuzzy discretization for the pattern classification problems," *IEICE Transactions on Information and Systems*, vol. 90, no. 7, pp. 1047–1054, 2007.
- [24] K. A. de Jong, *An analysis of the behavior of a class of genetic adaptive systems [Ph.D. thesis]*, University of Michigan, Ann Arbor, Mich, USA, 1975.
- [25] R. N. Khushaba, A. Al-Ani, and A. Al-Jumaily, "Differential evolution based feature subset selection," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, December 2008.
- [26] R. N. Khushaba, A. Al-Ani, and A. Al-Jumaily, "Feature subset selection using differential evolution and a statistical repair mechanism," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11515–11526, 2011.
- [27] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.
- [28] Automatic Weather Stations, <http://www.automaticweatherstation.com>.
- [29] R. Chang, Z. Pei, and C. Zhang, "A modified editing k -nearest neighbor rule," *Journal of Computers*, vol. 6, no. 7, pp. 1493–1500, 2011.
- [30] T. R. Golub, D. K. Slonim, P. Tamayo et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–527, 1999.
- [31] Y. H. Kim, S. Y. Lee, and B. R. Moon, "A genetic approach for gene selection on microarray expression data," in *Genetic and Evolutionary Computation—GECCO 2004*, K. Deb, Ed., vol. 3102 of *Lecture Notes in Computer Science*, pp. 346–355, 2004.
- [32] Y. Yin, D. Han, and Z. Cai, "Explore data classification algorithm based on SVM and PSO for education decision," *Journal of Convergence Information Technology*, vol. 6, no. 10, pp. 122–128, 2011.
- [33] Wikipedia, <http://en.wikipedia.org>.
- [34] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference*, pp. 485–492, July 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

