

Article

AWMC: Abnormal-Weather Monitoring and Curation Service Based on Dynamic Graph Embedding

Yuxuan Gu ^{1,†} , Jiakai Gu ^{1,†} , Gen Li ^{1,†} , Heeseung Yun ^{1,†}, Jason J. Jung ^{1,*} , Sojung An ² and David Camacho ³ 

¹ Department of Computer Engineering, Chung-Ang University, Seoul 06974, Korea

² Korea Institute of Atmospheric Prediction Systems, Seoul 07071, Korea

³ Department of Computer Systems Engineering, Universidad Politécnica de Madrid, 28040 Madrid, Spain

* Correspondence: j3jung@cau.ac.kr

† These authors contributed equally to this work.

Abstract: This paper presents a system, namely, the abnormal-weather monitoring and curation service (AWMC), which provides people with a better understanding of abnormal weather conditions. The service can analyze a set of multivariate weather datasets (i.e., 7 meteorological datasets from 18 cities in Korea) and show (i) which dates are mostly abnormal in a certain city, and (ii) which cities are mostly abnormal on a certain date. In particular, the dynamic graph-embedding-based anomaly detection method was employed to measure anomaly scores. We implemented the service and conducted evaluations. Regarding the results of monitoring abnormal weather, AWMC shows that the average precision was approximately 90.9%, recall was 93.2%, and F1 score was 92.1% for all the cities.

Keywords: abnormal weather visualization system; anomaly detection; graph embedding



Citation: Gu, Y.; Gu, J.; Li, G.; Yun, H.; Jung, J.J.; An, S.; Camacho, D. AWMC: Abnormal-Weather Monitoring and Curation Service Based on Dynamic Graph Embedding. *Appl. Sci.* **2022**, *12*, 10444. <https://doi.org/10.3390/app122010444>

Academic Editor: Arcangelo Castiglione

Received: 3 October 2022

Accepted: 14 October 2022

Published: 17 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Abnormal weather is the occurrence of extreme weather conditions. In recent years, the frequent occurrence of abnormal weather has affected people's life. Therefore, climate changes on Earth are an important concern for meteorologists [1]. The World Meteorological Organization reported in 2020 regarding the state of global climate that many weather-related indicators are changing dramatically, which could damage the global economy and ecosystems [2].

Anomaly detection is the problem of mining patterns that do not match the expected patterns [3]. As an important research area in data science, anomaly detection is widely applied in several fields, such as fraud detection, network monitoring, and medicine. Anomaly detection also plays an important role in many fields, and many researchers are investigating various anomaly detection methods. It can be divided into supervised and unsupervised anomaly detection. Supervised anomaly detection methods include k-nearest neighbor, supervised neural networks, decision trees, support vector machines; unsupervised anomaly detection includes k-means, fuzzy c-means, and unsupervised niche clustering [4].

However, with the development of artificial intelligence and the emergence of deep learning, some research has applied deep-learning models to detect an anomaly. For example, Li et al. [5] combined a stacked autoencoder, and long- and short-term memory for anomaly detection in mechanical systems. The accuracy of their method was 99%. Scott et al. [6] combined a convolutional neural network and gated recurrent unit for water-level anomaly detection. Their results showed that the mean absolute error of this model was lower than that of other models.

Existing anomaly detection methods fit a model through labeled data and detect the anomalies. However, these methods all require prior data labeling and model training.

For example, Liu et al. [7] proposed a model based on long- and short-term memory, and an autoencoder for detecting heart-rate anomalies. However, the used dataset comprised data labeled by cardiologists in advance. To solve this problem, Li and Jung [8] proposed an anomaly detection method based on dynamic graph embedding.

In addition, many studies in anomaly detection on multiple climate time series are devoted to improving the accuracy of anomaly detection. However, it is hard for people who do not work in computer science to understand abnormal weather through these studies. In addition, existing weather systems do not have a function that can show the extent of weather anomalies. To solve this problem and allow for people to understand the extent of weather anomalies in their city, we designed and provide an abnormal-weather monitoring and curation system. This system monitors the extent of weather anomalies and shows weather data changes for each city. At the same time, this system compares the anomalies for each city. It shows the weather data of a user's city and the average weather data of all cities to enable people to understand why the weather in their cities is abnormal.

Contribution

The key contributions of this research are summarized as follows.

- We propose an anomaly detection method that constructs dynamic graphs from spurious relationships between weather data, and uses graph entropy to measure the similarity between two graphs. Then, we propose a dynamic graph embedding model to construct an embedding space for anomaly detection. Lastly, existing anomaly detection methods can be applied in the embedding space for anomaly detection.
- We designed an abnormal-weather monitoring and curation system (AWMC) that shows abnormal changes in the weather.
- We provide a system that visualizes weather data and anomalies. This system can help people in understanding the relationship between weather changes and the extent of anomalies. It can also help people in understanding the reasons for abnormal weather.

The rest of this paper has the following structure. In Section 2, we describe the related work. Section 3 discusses an anomaly detection method based on dynamic graph embedding. In Section 4, the architecture and implementation of AWMC are explained. Section 5 is the evaluation of the AWMC system. Section 6 presents the conclusion and future work.

2. Related Work

Shirakawa et al. [9] built a web application that displays climate changes, natural disaster risks, and socioeconomic conditions. The aim of this web application is to reduce the risk of natural disasters and improve economic efficiency while adapting to weather changes. The application collects data from natural disasters and economic aspects. These data include flood, landslide, and drought risks, and crop production value. In this web application, 2D and 3D views are included. The 2D view displays disaster risk map information and socioeconomic statistics that include the gross regional product and the output of various industries. People can understand the relationship between natural disaster risk and economic output through their changes and develop relevant policies. In addition, the 3D view of the site enhances people's understanding of natural disasters. However, this web application was only on the statistical level and did not use artificial-intelligence techniques to process the data, but just provided a visual interface. In our system, we visualize the changes in weather data and use graph embedding to detect weather anomalies so that people can better understand weather conditions.

Chow et al. [10] proposed and built a data visualization tool for treatment-process monitoring to solve the problem of traditional water quality assessment being retrospective and intermediate. The functions of data visualization tools include water quality data aggregation, prediction, and anomaly detection. In data aggregation, the authors first corrected problematic data in the original dataset, including negative, null, and extreme values. They changed extreme values to triple the hourly data's variance, and negative and null values to the mean of the hourly data. The purpose of data aggregation is to ensure visit efficiency and filter peaks to show the average trends of the data. In their tool, two main working modes were included: query mode and analysis mode. The query mode mainly visualizes data. Data visualization includes data series, anomalies, and predictions. The prediction function of this data visualization tool builds a prediction model through the display of time series and model learning. Predicted values can also be provided for the anomaly detection function. If a predicted value exceeds the normal range, it is shown as abnormal. In addition, the density-based anomaly detection method is applied to the anomaly detection function of this tool, which can allow for staff to identify problems and resolve them immediately. However, in the anomaly detection function, this visualization tool just uses the Pearson product moment correlation method to detect anomalies. This anomaly detection method did not extract feature from data and map the original data into a low-dimensional space. In our system, we build graphs from the relationships between multiple time series and embed the graphs into a low-dimensional space to extract features from weather data.

Lumley et al. [11] compared and analyzed 41 weather data visualization tools from the five aspects of purpose, data content, visual presentation, interactivity, and web technology. Regarding purpose, just over 66% of weather visualization sites supported exploration, and only 17% of sites supported user-simulated weather models. In the data content aspect, environmental change was the most common data type in most systems, with 78% of weather visualization tools showing data on air temperature, precipitation, and extreme weather events. In the data visualization aspect, 88% of the tools visualized weather data on the map. Therefore, the map was the core technology of weather visualization, and it was the most different between weather visualization tools and other visualization tools. For each tool interaction, the exploratory weather visualization tool statistically had 1.6 more operators than those of the interpretive weather visualization tool. Lastly, for each tool's site technology, most tools drew from external libraries to support visualization in their website. In addition, 80% of them use JavaScript-based open web maps on their website.

We compare the AWMC system with existing weather systems regarding the four aspects of function, method, feature extraction, and database. The functions of existing weather systems are focused on weather forecasting and monitoring. However, compared with existing systems, the AWMC system not only monitors changes in weather data, but also provides a function for detecting abnormal weather, giving people a better understanding of weather conditions. In addition, existing weather forecasting systems do not extract features from weather data, which may lead to a decrease in the accuracy of forecasts. In the AWMC system, we extract relational features between weather time series. More details of the comparison are shown in Table 1.

Table 1. Comparison of existing weather systems and AWMC (✗: the system does not use the method, feature extraction, or database.)

System	Function	Method	Feature Extraction	Database
Greer et al. [12]	Weather data visualization	✗	✗	PostgreSQL, Hadoop
Granville et al. [13]	Fire weather index interpolation	Interpolation method	✗	✗
Kumar et al. [14]	Weather forecasting	✗	✗	✗
Gahlot et al. [15]	Weather monitoring	Zigbee	✗	✗
Latha et al. [16]	Weather forecasting	Support vector machine	✗	MySQL
Srinivasan et al. [17]	Weather forecasting	✗	✗	MySQL
Zubov et al. [18]	Weather forecasting	Analogue complexing algorithm	✗	✗
Wica et al. [19]	Temperature and rainfall prediction	Neural network	✗	✗
Beeharry et al. [20]	Weather forecasting	K-nearest Neighbors	✗	IBM Cloudant
Bendre et al. [21]	Weather forecasting	MapReduce algorithm	✗	✗
Baste et al. [22]	Low Cost Weather Monitoring	✗	✗	MySQL
Munandar et al. [23]	Real-time weather monitoring	✗	✗	✗
Albatli et al. [24]	Weather data interpolation	Linear interpolation	✗	MySQL
Hartung et al. [25]	Wildland fire monitoring	✗	✗	✗
Thouta et al. [26]	Historical weather data visualization	Online analytical processing	✗	MySQL
Steiner et al. [27]	Automatic weather information retrieval	✗	✗	MySQL
Goldstein et al. [28]	Real-time flood threat estimation	Subsurface hydrologic analysis	✗	MS-SQL
Bala et al. [29]	Weather monitoring	ID3 algorithm	✗	MySQL
AWMC	Abnormal-weather monitoring and curation	Dynamic graph embedding	Relational feature	MySQL

3. Method

This section describes the anomaly detection method for the AWMC system. The system uses an anomaly detection method based on the dynamic graph embedding proposed by Li [8]. The method can be divided into four steps. The first step is to find spurious relationships between multiple time series and build dynamic graphs. The second step calculates the graph entropy. The third step is to propose a dynamic graph embedding model on the basis of graph entropy to build the embedding space. Lastly, existing anomaly detection methods are applied to the embedding space to detect anomalies. The details of each step are as follows.

3.1. Construction of Dynamic Graph

Our idea was to discover the relationship among weather time series to detect abnormal weather conditions, and the graph structure was better than other data structures for modeling relationships between multiple items. Therefore, this study obtained the changing relationships among weather time series using a dynamic graph. In the graph, the vertex denotes a time series, and the edge represents the relationship among the time series. The weight of the edge is determined by causality and correlation between different time series. Causality is defined as the extent to which one series improves the prediction of another series and is expressed as follows.

$$C(x, y) = \begin{cases} 1 & \text{if } p \geq 0.05 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where x and y are two time series, $C(x, y)$ indicates the causality between x and y , and p is the probability that the two time series are not causally related.

The Granger causality test [30] was used to calculate the short-run causality between two time series, x and y . The Granger causality test first used the value of the previous time

interval of x to predict the current x . Then, it used values of the previous time interval of x and y to predict the value of the current x . Lastly, we determined whether the y time series helped in predicting the x time series by comparing it with two predict results. In addition, p is the probability that the two time series are not causally related. If p is greater than 0.5, the two time series are not causally related.

Pearson’s correlation coefficient (PCC) [31] was used to express the correlation between time series. Additionally, the weight of the edge was calculated with causality and PCC, as follows.

$$R(x, y) = \begin{cases} 1 & \text{if } C(x, y) = 0 \\ C(x, y) - |PCC| & \text{otherwise} \end{cases} \quad (2)$$

where $R(x, y)$ indicates the spurious correlation coefficient between two time series, and $C(x, y)$ indicates the causality between two time series. If $C(x, y) = 0$, the two time series are uncorrelated. Otherwise, the spurious correlation coefficient is the difference between causality and the absolute value of PCC. Therefore, the dynamic graph is denoted as $\mathcal{G} = \{\langle G_t \rightarrow G_{t+1} \rangle | G_i \in \mathcal{G}, t \in [0, T]\}$ where T is the number of time intervals, G_i is the graph at time interval i , and G_t indicates the precedent graph of $G_t + 1$.

3.2. Calculation of Graph Entropy

Graph entropy in this method is based on information entropy [32]. Information entropy describes the uncertainty of the occurrence of each possible event in the information source. The formula for calculating the information entropy of events X is as follows.

$$e(X) = - \sum_{i=1}^N P(x_i) \log_2 P(x_i)$$

where X indicates the set of events, x_i indicates the i th event, and N is the number of events. $P(x_i)$ indicates the probability of the x_i event.

To calculate graph entropy, vertex entropy that is based on the weight between vertices in the graph is defined and calculated as follows.

$$e(v_i) = \sum_{j=0, j \neq i}^N -w_{i,j} \log_2 w_{i,j} \quad (3)$$

where v_i indicates the i th vertex in the graph, $w_{i,j}$ indicates the weight between vertices v_i and v_j , and N indicates the number of vertices in the graph. The weight value between vertices v_i and v_j is equal to the value of $R(v_i, v_j)$, which is the spurious correlation coefficient between vertices v_i and v_j .

The definition of graph entropy that is the sum of the entropy of each vertex in the graph is proposed. Therefore, graph entropy is calculated as follows.

$$e(G) = \sum_{i=1}^N e(v_i) \quad (4)$$

where N indicates the number of vertices in the graph. In addition, the entropy of the dynamic graph is the set of graph entropies for each time interval $t \in [0, T]$. Then, the dynamic graph entropy is as follows.

$$\mathcal{E} = \{e(G_t) | t \in [0, T]\} \quad (5)$$

If the values in the time series change, the weights between the vertices and thereby graph entropy also change. Therefore, we can detect the anomalies on the basis of graph entropy.

3.3. Dynamic Graph Embedding Based on Entropy

The graph similarity algorithm to find the most similar graph corresponding to each graph is proposed. The graph similarity algorithm is as follows.

$$d(e(G_i), e(G_j)) = \sqrt{\|e(G_i) - e(G_j)\|_2^2} \quad (6)$$

Then, we can obtain the set of graphs that are most similar to each graph. This set is called the supervised dynamic matrix that contains the most similar graph for each graph. The set is formulated as $\mathcal{S} = \{(S_t \rightarrow S_{t+1}) | S_i \in \mathcal{S}, t \in [0, T]\}$, where T is the number of time intervals, S_i is the graph at time interval i , and S_t indicates the precedent graph of S_{t+1} . The embedding model includes two autoencoders. An autoencoder is used to reconstruct the dynamic weather graph, and another autoencoder is used to reconstruct the dynamic supervised graph. The embedding vector is calculated by the autoencoder. More specifically, the encoder extracts the features of the graph for mapping to the embedding space, and the embedding vector is calculated by reversing the encoder's computation. The loss function is based on graph entropy similarity to shorten similar graphs in the distance in the embedding space. The formula of the loss function based on calculating graph entropy similarity in the embedding space is as follows.

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T \|e(g_t) - e(s_t)\|_2^2 + \frac{1}{T} \sum_{t=1}^T \|G_t - \widehat{G}_t\|_2^2 + \frac{1}{2} \sum_{i=0}^I (\|W\|_2^2 + \|\widehat{W}\|_2^2) \quad (7)$$

where W^i and \widehat{W}^i indicate the weight in the i -th layer of the encoder and decoder, respectively, and g_t and s_t indicate the embedding vectors of graphs G_t and S_t , respectively.

In addition, the gradient-descent [33] and backpropagation [34] algorithms were used to train the model, and update the weight and basis. Then, we obtained the final dynamic graph embedding model.

4. AWMC Description

AWMC is a system that can detect weather anomaly scores. In addition, to help people in understanding the causes of weather anomalies, the system allows for selecting any time interval to see changes in weather anomalies and corresponding weather data during that time interval. Next, we discuss the AWMC's design and implementation.

4.1. Design

4.1.1. Back-End Design

The architecture of the back-end is shown in Figure 1. The back-end of the AWMC system comprises three components: a data processor, a graph embedding model, and an anomaly detector. The data processor compensates for missing values in the original dataset, and the graph embedding model builds the embedding space and maps each day's weather data as an embedding vector. Lastly, the anomaly detector applies the existing anomaly detection algorithm to the embedding space and calculates the anomaly score for each day.

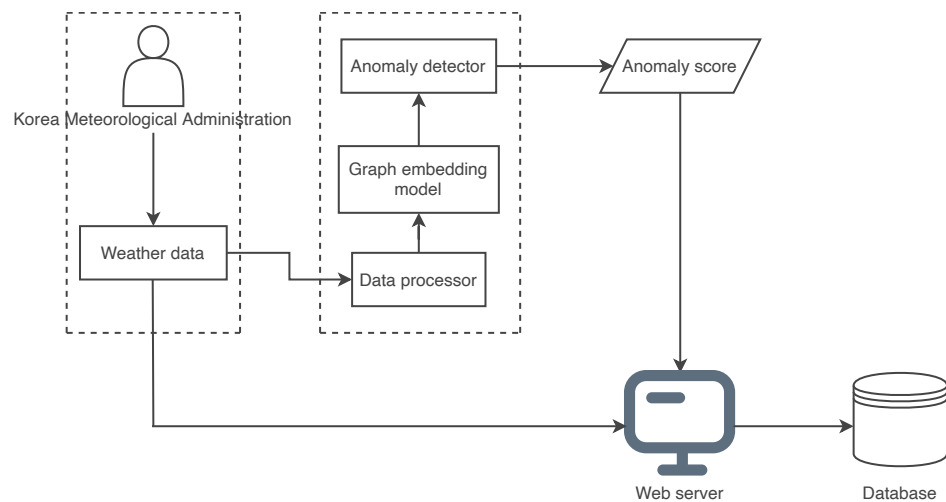


Figure 1. Back-end architecture.

4.1.2. Front-End Design

In the front-end design, the homepage includes two parts. The first part is the map of Korea, and the second part is the visualization of weather anomalies and weather data. In the second part, people can select the time interval to see changes in weather anomalies and weather data in that time interval. In addition, we designed box plots to show the dispersion of weather data. The second part includes some items as follows.

- **Time selector:** can be selected the time interval.
- **Anomaly-score line chart:** shows changes in weather anomalies during the selected time interval.
- **Temperature line chart:** shows changes in temperature during the selected time interval.
- **Humidity line chart:** shows changes in humidity during the selected time interval.
- **Vapor-pressure line chart:** shows changes in vapor pressure during the selected time interval.
- **Dew-point temperature line chart:** shows changes in dew-point temperature during the selected time interval.
- **Local air pressure line chart:** shows changes in local air pressure during the selected time interval.
- **Sea-surface pressure line chart:** shows changes in sea surface pressure during the selected time interval.
- **Ground-temperature line chart:** shows changes in ground temperature during the selected time interval.
- **Weather data box plot:** shows the dispersion of weather data that includes temperature, humidity, vapor pressure, dew-point temperature, local air pressure, sea-surface pressure, and ground temperature.

4.1.3. Database Design

The database of the AWMC system is based on the Relational Database Management System (RDBMS). The database has three tables, namely, 'coordinates', 'weather', and 'anomaly score'. The 'coordinates' table stores the city name, latitude, and longitude. The 'weather' table stores the date and the weather data of each city. Weather data include temperature, humidity, vapor pressure, dew-point temperature, local air pressure, sea-surface pressure, and ground temperature. The 'anomaly score' table stores the date and score of each day for each city. The score represents the weather anomaly extent of each day for each city. In addition, we set and store the KID of each city in each table. KID is the label of each city. With KID as the primary key of table coordinates, and the foreign

key of the table weather and anomaly score, we can build the relationship of each table. The relationship between the tables is shown in Figure 2.

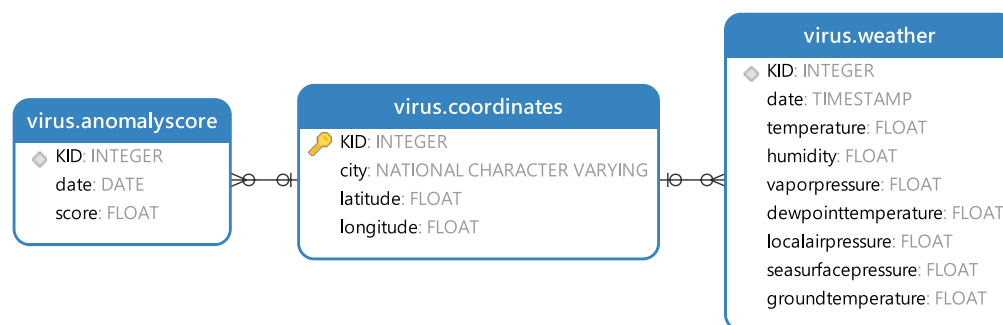


Figure 2. Relationship between the tables.

4.2. Implementation

4.2.1. Server

Our server uses Intel(R) Xeon(R) Platinum 8167M CPU @ 2.00GH, Ubuntu 22.04.1 LTS, Python 3.10.4 for the back end, Pytorch 1.8 for training the embedding vector, Apache 2.4.46 used in web server, Mysql 8.0.16 is used for database.

4.2.2. Data Source

The source of weather data is from the Korean Meteorological Administration. The dataset included weather data for each day from 2018 to 2021 for 18 cities in Korea: Seoul, Busan, Incheon, Wonju, Chungju, Seosan, Boryeong, Daejeon, Jeonju, Gwangju, Mokpo, Yeosu, Daegu, Gumi, Andong, Yeongdeok, Ulsan, and Jeju. In the dataset, weather data were of seven types: temperature, humidity, vapor pressure, dew-point temperature, local air pressure, sea-surface pressure, and ground temperature. In addition, the weather data for each hour from 0:00 to 23:00 were included in the data for each day. This means that the dataset recorded every hour of weather data from 2018 to 2021.

4.2.3. Data Processing

The data processing of our system is divided into five steps. The first step is to clean the dataset. In the original dataset, some of the weather data were missing. Therefore, we used the `pd.fillna()` function in the pandas package to fill in the missing values. The `pd.fillna()` function fills in missing data and includes six parameters: value, method, axis, inplace, limit, and downcast. We chose `ffill` in the method parameter, which uses previous nonmissing values to fill the missing values. The second step is to build dynamic weather graphs, calculate the graph entropy of each graph, and build dynamic supervised graphs. To build a dynamic weather graph, since the weather data included seven types, we built the graph matrix that transformed the data of every day into a 7×7 matrix. The rows and columns in the graph matrix represent the seven types of weather data, and the values inside the matrix represent the value of *PCC* among the seven types of weather data. Therefore, the values on the diagonal of the graph matrix were all 1. Then, we calculated the graph entropy of each graph by using Equations (3)–(5). Lastly, to build a dynamic supervised graph, we calculated the most similar corresponding graph to each graph based on Equation (6).

The third step builds a dynamic graph embedding model. We built two autoencoders, one to train the dynamic weather graphs, and another to train training dynamic supervised graphs.

The fourth step is to apply the local outlier factor (LOF) algorithm [35] to the embedding space to calculate the anomaly scores of each day. LOF is an unsupervised anomaly detection method based on density that calculates the local outlier factor for each point in the space; if the local outlier factor is much greater than 1, the point is judged to be an anomaly, and if the local outlier factor is close to 1, the point is judged to be a normal point.

For calculating local outlier factors, we needed to first calculate the reachability distance. The formula of reachability distance is defined as follows.

$$\text{reachability} - \text{distance}_k(o, p) = \max\{k - \text{distance}(p), d(o, p)\} \quad (8)$$

where o and p are data points in the space, $k - \text{distance}(p)$ is the distance of point p to the k th nearest neighbor. and $d(o, p)$ is the distance between points o and p . Next, we calculated the local reachability density. The formula of local reachability density of point o is defined as follows.

$$\text{lrd}_k(o) = 1 / \left(\frac{\sum_{p \in N_k(o)} \text{reachability} - \text{distance}_k(o, p)}{|N_k(o)|} \right) \quad (9)$$

where $N_k(o)$ is the set of the k nearest neighbors of point o . Then, we could calculate the local outlier factor by using local reachability density. The formula of the outlier factor is as follows.

$$\text{LOF}_k(p) = \frac{\sum_{o \in N_k(p)} \text{lrd}_k(o)}{|N_k(p)|} / \text{lrd}_k(p) \quad (10)$$

In our system, we set the 10 nearest neighbors, and set the anomaly rate to 0.1 by using LOF function in the sklearn package. Lastly, we could calculate the local outlier factor of each point as the anomaly score for each day.

In the last step, since the weather data of each day included the data of each hour from 0:00 to 23:00, we calculated the average of each type of weather data for each day to understand the relationship between changes in weather anomalies and weather data.

4.2.4. Visualization

First, we used the Google API to fetch Google Maps from around the world. Then, each city was marked on the map by its latitude and longitude. People could click on these markers to see the weather conditions in each city. In the left part of the web, we produced a line chart to visualize the changes of anomalies for each city in a certain time interval. We also produced the line charts to visualize weather data changes in the same time interval. People could find the relationship between the anomaly scores and the weather data via changes in these lines and understand the causes of weather anomalies. In addition, we produced box plots to visualize the dispersion of weather data. A box plot is a statistical chart used to show the dispersion of a set of data. In a box plot, there is a rectangular box with the upper and lower ends of the box corresponding to the upper and lower quartiles of the dataset. The upper and lower quartiles are defined as $Q3$ and $Q1$, and the distance between $Q3$ and $Q1$ is called the interquartile distance (IQR). In addition, we drew a line in the rectangular box, which is the median of the dataset, and two lines at $Q3 + 1.5IQR$ and $Q1 - 1.5IQR$ that are outlier truncation points. The outlier truncation points were split into upper and lower. If the data are larger than the upper or smaller than the lower, then they are marked as an outlier in the box plot. People can also analyze the causes of weather anomalies by using the box plots and line charts of anomaly scores. Lastly, we added a click function to the anomaly-score line chart. We clicked on each point in the anomaly-score line chart. Then, the map visualized the extent of anomalies for each city by circles. The size of the circle is based on the anomaly score. If the anomaly score is larger, then the circle is larger. People can use this visualization to compare the extent of anomalies in their city with that in other cities to further understand their weather conditions.

We used Echart [36] from the Apache open-source project to draw these line charts and box plots. In addition, we used the AntV-L7 API to visualize the anomalies of each city on the map. AntV-L7 API is an open-source WebGL-based visual analysis engine for large-scale geospatial data launched by the AntV data visualization team of Ant Group.

4.3. Functionalities and Illustrative Example

The functionalities of the AWMC system are as follows.

- The system shows the anomaly scores of each day for each city in a certain time interval and changes in the anomaly scores.
- The system shows weather data of the temperature, humidity, vapor pressure, dew point temperature, local air pressure, sea-surface pressure, and ground temperature of each day for each city in a certain time interval, and changes in these weather data.
- The system shows the dispersion of seven types of weather data and marks outlier data points.
- The system shows the anomaly score of each city for one day on the map.

Next, we discuss an illustrative example for our system. First, we click the marker on the map and select the city we want to check. Then, a box that includes the time selector pops up on the left side of the page. We can select the time interval that we want to check. Then, the line charts of anomaly scores and weather data for this city in this time interval appear. At the same time, the box plots of the weather data in this time interval also appear. Lastly, by clicking on any point in the anomaly-score line chart, the map shows the weather anomaly extent of all cities on the chosen day.

4.4. User Interface

The user interface of the AWMC system is shown in Figure 3. The example of the user interface is from 3 March to 8 May 2021 in Wonju. The figure shows that the weather anomaly score on 4 April was the highest in this time interval. Then, among changes in the relevant weather data below, vapor pressure and dew-point temperature had relatively high values on 4 April. At the same time, the size of the circles on the map also shows that Andong had the highest weather anomalies among all cities on 4 April.

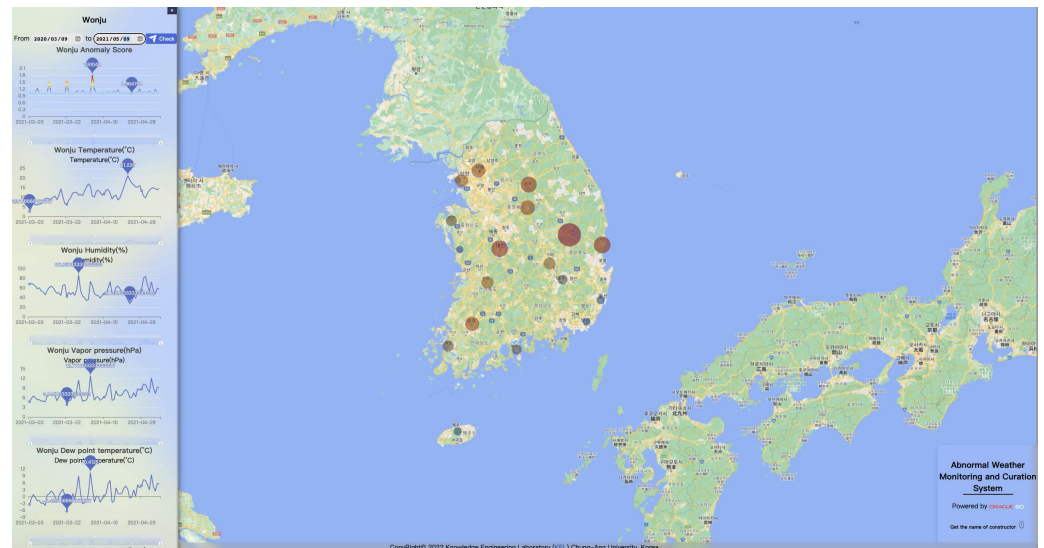


Figure 3. Example of the user interface.

5. Evaluation

Since AWMC is an abnormal-weather detection system, the evaluation of our system is an evaluation of our anomaly detection method. In our method, we detected anomaly points in the embedding space by the LOF algorithm. In our evaluation, we selected some anomaly points in the embedding space as our labels. These points were the top 10% of points with the greatest distance to the mean point, which is defined as the average of all data points in the embedding space. The mean point is denoted with $c = \frac{1}{T} \sum_{t=1}^T g_t$, where T is the total number of data points. Therefore, we compared the Euclidean distance [37]

from each data point to the mean point in the embedding space and found the anomaly points, which were then labeled. Then, we evaluated our anomaly detection method using precision, recall, and F1 score. Precision is calculated as follows.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

where TP is the number of points that were detected to be abnormal and in fact were normal, FP is the number of points that were detected to be abnormal, but in fact were normal. Recall is calculated as follows.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

where TP is the number of points that were detected to be abnormal and in fact were normal, and FN is the number of points that were detected to be normal, but in fact were abnormal. The F1-score is calculated as follows.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

We selected five cities for evaluation: Seoul, Incheon, Busan, Daejeon, and Daegu. The evaluation results are shown in Table 2.

Table 2. Evaluation results.

	Precision	Recall	F1 Score
Seoul	0.904	0.928	0.916
Incheon	0.918	0.938	0.928
Busan	0.901	0.924	0.913
Daejeon	0.908	0.935	0.922
Daegu	0.915	0.937	0.926

Table 1 shows that Incheon's precision was 0.918, recall was 0.938, and F1 score was 0.928. Three metrics were all higher than those of the four other cities because our model detects anomalies by finding relationships between different time series. If there is a large change in the relationship between the different time series, it represents an anomaly. Incheon had a larger change in the relationship between weather time series data than other cities did. So, the abnormal patterns of Incheon data points in the embedding space were more obvious than other cities' data points. Therefore, when using the LOF anomaly detection method in the embedding space of Incheon data, it could be easier to detect abnormal data points and obtain a higher F1 score than for other cities.

6. Conclusions and Future Work

In this work, we proposed an anomaly detection method based on dynamic graph embedding, and designed the AWMC system. In the anomaly detection method, we first constructed the dynamic graph by finding the relationship among multiple time series, then calculated the graph entropy, and lastly constructed a graph embedding model on the basis of graph entropy for anomaly detection. We discussed the AWMC system's design, implementation, functionalities, an illustrative example, and user interface. The system can help people in understanding abnormal weather and its causes by visualizing changes in weather anomalies and corresponding weather data. In addition, the system visualizes the weather anomalies of each city on a map to compare anomalies for each city. We applied Korean weather data to the system. The result shows that the proposed method reached average precision of approximately 90.9%, recall of 93.2%, and F1 score of 92.1% for all the cities. Lastly, the AWMC system was implemented via the AWMC website (<http://awmc.caucse.club>, accessed on 2 October 2022).

However, due to limitations, we could not detect anomalies in the weather data in real time. Therefore, to improve this system, we aim to find ways that would allow for this system to be able to detect anomalies in weather data in real time. In addition, to understand global weather changes, we will apply more countries.

Author Contributions: Conceptualization, Y.G., J.G., G.L. and J.J.J.; methodology, G.L. and J.J.J.; software, Y.G., J.G. and G.L.; validation, Y.G., J.G. and G.L.; formal analysis, J.J.J. and D.C.; investigation, Y.G. and H.Y.; resources, J.J.J., S.A. and D.C.; data curation, Y.G., J.G. and G.L.; writing—original draft preparation, Y.G.; writing—review and editing, J.J.J., S.A. and D.C.; visualization, Y.G., J.G. and G.L.; supervision, J.J.J.; project administration, J.J.J. and D.C.; funding acquisition, J.J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Chung-Ang University Research Grants in 2021. This work was also supported in part by Oracle Cloud credits and related resources provided by the Oracle for Research program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by Chung-Ang University Research Grants in 2021. This work was also supported in part by Oracle Cloud credits and related resources provided by the Oracle for Research program.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shao, X.; Liao, Y.; Liu, Y.; Ye, D.; Si, D.; Wang, Y.; Yu, N. Global major weather and climate events in 2015 and possible cause. *Meteorol. Mon.* **2016**, *42*, 489–495.
2. WMO. *State of the Global Climate 2020*; World Meteorological Organization (WMO): Geneva, Switzerland, 2021.
3. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–58. [[CrossRef](#)]
4. Omar, S.; Ngadi, A.; Jebur, H.H. Machine learning techniques for anomaly detection: An overview. *Int. J. Comput. Appl.* **2013**, *79*. [[CrossRef](#)]
5. Li, Z.; Li, J.; Wang, Y.; Wang, K. A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 499–510. [[CrossRef](#)]
6. Miao, S.; Hung, W.H. River Flooding Forecasting and Anomaly Detection Based on Deep Learning. *IEEE Access* **2020**, *8*, 198384–198402. [[CrossRef](#)]
7. Liu, P.; Sun, X.; Han, Y.; He, Z.; Zhang, W.; Wu, C. Arrhythmia classification of LSTM autoencoder based on time series anomaly detection. *Biomed. Signal Process. Control* **2022**, *71*, 103228. [[CrossRef](#)]
8. Li, G.; Jung, J.J. Entropy-based dynamic graph embedding for anomaly detection on multiple climate time series. *Sci. Rep.* **2021**, *11*, 1–10. [[CrossRef](#)]
9. Shirakawa, H.; Suanpaga, W. Development of a web application for climate change adaptation in Thailand. In *Proceedings of the IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2022; Volume 1016, p. 012023.
10. Chow, C.W.; Liu, J.; Li, J.; Swain, N.; Saint, C.P. A Data Visualisation Tool for Treatment Process Monitoring in Web Browsers. *Water Conserv. Sci. Eng.* **2022**, 1–11. [[CrossRef](#)]
11. Lumley, S.; Sieber, R.; Roth, R. A framework and comparative analysis of web-based climate change visualization tools. *Comput. Graph.* **2022**, *103*, 19–30. [[CrossRef](#)]
12. Greer, M.; Rodriguez-Martinez, M.; Seguel, J. Open source cloud computing tools: A case study with a weather application. In *Proceedings of the IEEE Open Source Cloud Computing*, Miami, FL, USA, 5–10 July 2010.
13. Granville, K.; Woolford, D.G.; Dean, C.; Boychuk, D.; McFayden, C.B. On the selection of an interpolation method with an application to the Fire Weather Index in Ontario, Canada. *Environmetrics* **2022**, e2758. [[CrossRef](#)]
14. Kumar, S.; Renukadevi, P.; Suguna, M.; Jeyakumar, D. The Performance Analysis of a Location based Weather Identification Device. In *Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 22–24 June 2022; pp. 362–367. [[CrossRef](#)]
15. Gahlot, N.; Gundkal, V.; Kothimbire, S.; Thite, A. Zigbee based weather monitoring system. *Int. J. Eng. Sci.* **2015**, *4*, 61–66.
16. Latha, C.; Paul, S.; Kirubakaran, E.; Sathianarayanan, A. A service oriented architecture for weather forecasting using data mining. *Int. J. Adv. Netw. Appl.* **2010**, *2*, 608–613.

17. Srinivasan, K.; Nema, A.; Huang, C.H.; Ho, T.Y. Weather Forecasting Application Using Web-Based Model-View-Whatever Framework. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taichung, Taiwan, 19–21 May 2018; pp. 1–2. [[CrossRef](#)]
18. Zubov, D. Development of Web Application Structure for Weather Inductive Forecasting. In Proceedings of the 4th International Workshop on Inductive Modelling (ICIM'2011), Kyiv, Ukraine, 4–10 July 2011; pp. 123–127.
19. Wica, M.; Witkowski, M.; Szumiec, A.; Ziebura, T. Weather forecasting system with the use of neural network and backpropagation algorithm. In Proceedings of the International Conference on Data Engineering and Communication Technology; Springer: Singapore, 2019; Volume 2468, pp. 37–41.
20. Beeharry, Y.; Fowdur, T.P.; Sunglee, J.A. A Cloud-Based Real-Time Weather Forecasting Application. In Proceedings of the 2019 14th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Niš, Serbia, 23–25 October 2019; pp. 294–297. [[CrossRef](#)]
21. Bendre, M.R.; Thool, R.C.; Thool, V.R. Big data in precision agriculture: Weather forecasting for future farming. In Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 4–5 September 2015; pp. 744–750. [[CrossRef](#)]
22. Baste, P.; Dighe, D. Low cost weather monitoring station using Raspberry Pi. *Int. Res. J. Eng. Technol.* **2017**, *4*, 3184–3189.
23. Munandar, A.; Fakhurroja, H.; Rizqyawan, M.I.; Pratama, R.P.; Wibowo, J.W.; Anto, I.A.F. Design of real-time weather monitoring system based on mobile application using automatic weather station. In Proceedings of the 2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), Jakarta, Indonesia, 23–24 October 2017; pp. 44–47. [[CrossRef](#)]
24. Albatli, A.M.; Alzamil, I.A. A prototype of an automated live weather interpolation system using a web application. In Proceedings of the 2011 7th International Conference on Next Generation Web Services Practices, Salamanca, Spain, 19–21 October 2011; pp. 18–23. doi: 10.1109/NWeSP.2011.6088147. [[CrossRef](#)]
25. Hartung, C.; Han, R.; Seielstad, C.; Holbrook, S. FireWxNet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In Proceedings of the 4th International Conference on Mobile Systems, Applications and Services, Uppsala, Sweden, 19–22 June 2006; pp. 28–41.
26. Thouta, N. Mining Weather Data: A Web Application for California Smart Grid Center. Ph.D. Thesis, California State University, Sacramento, CA, USA, 2015.
27. Steiner, J.J.; Minoura, T.; Xiong, W. WeatherInfo: A Web-based weather data capture system. *Agron. J.* **2005**, *97*, 633–639. [[CrossRef](#)]
28. Goldstein, S.; Oyekwe-Madumelu, C.; Regina, J.; Sainju, A.M. FloodImpact: A Web Application to Identify Flood Extent and Community Vulnerabilities for Real-time Weather Forecasts. In *National Water Center Innovators Program Summer Institute Report 2017*; Technical Report; Consortium of Universities for the Advancement of Hydrologic Science, Inc.: Arlington, MA, USA, 2017; p. 85.
29. Arjun, D.S.; Bala, A.; Dwarakanath, V.; Sampada, K.S.; Prahlada Rao, B.B.; Pasupuleti, H. Integrating cloud-WSN to analyze weather data and notify SaaS user alerts during weather disasters. In Proceedings of the 2015 IEEE International Advance Computing Conference (IACC), Bangalore, India, 12–13 June 2015; pp. 899–904. [[CrossRef](#)]
30. Akadiri, S.S.; Lasisi, T.T.; Uzuner, G.; Akadiri, A.C. Examining the causal impacts of tourism, globalization, economic growth and carbon emissions in tourism island territories: Bootstrap panel Granger causality analysis. *Curr. Issues Tour.* **2020**, *23*, 470–484. [[CrossRef](#)]
31. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. *Noise Reduction in Speech Processing*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009; Volume 2.
32. Pele, D.T.; Lazar, E.; Dufour, A. Information entropy and measures of market risk. *Entropy* **2017**, *19*, 226. [[CrossRef](#)]
33. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [[CrossRef](#)]
34. Cao, Q.; Parry, M.E. Neural network earnings per share forecasting models: A comparison of backward propagation and the genetic algorithm. *Decis. Support Syst.* **2009**, *47*, 32–41. [[CrossRef](#)]
35. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 93–104.
36. Li, D.; Mei, H.; Shen, Y.; Su, S.; Zhang, W.; Wang, J.; Zu, M.; Chen, W. ECharts: A declarative framework for rapid construction of web-based visualization. *Vis. Inform.* **2018**, *2*, 136–146. [[CrossRef](#)]
37. Danielsson, P.E. Euclidean distance mapping. *Comput. Graph. Image Process.* **1980**, *14*, 227–248. [[CrossRef](#)]