Journal of IoT in Social, Mobile, Analytics, and Cloud (ISSN: 2582-1369)
www.irojournals.com/iroismac/

Anomaly Detection in Weather Forecasting System

Mrs. Arul Jothi S[1], Ms. Nandikaa G[2]

CSE Department, PSG College of Technology, Coimbatore, Tamil Nadu, India.

E-mail: [1]saj.cse@psgtech.ac.in, [2]21z325@psgtech.ac.in

Abstract

Predicting rainfall remains a challenging job in weather forecasting until now. The primary focus of weather forecasting is the prediction of the weather at a specific future period. The objective of this research is to predict the anomaly in multivariate weather prediction dataset. To predict the future weather condition, the variation in the conditions in past years must be utilized. The proposed model uses supervised anomaly detection techniques for weather forecasting system and then compares the results from each technique

Keywords: Anomaly detection, Weather forecasting, Neural network, Bayesian model

1. Introduction

Finding data points, items, observations, or events that deviate from the predicted pattern of a given group is known as anomaly detection. These anomalies are extremely rare, yet they could indicate a serious concern, such fraud or cyber breaches. In order to help in learning about the detection, identification, and prediction of the occurrence of these abnormalities, anomaly detection is heavily utilized in behavioral analysis and other forms of analysis. Outlier detection is another name for anomaly detection. To identify the many types of anomalies, present in a given data collection and to learn more about their occurrences, anomaly detection is primarily a data-mining technique. It can be used in areas like fault detection, fraud detection, weather forecasting system, and health monitoring system using wireless sensor networks. One of the most scientifically and technically difficult challenges in the globe over the past century has been weather forecasting, a crucial application of meteorology. The dynamic nature of the atmosphere makes it difficult to predict weather conditions with any degree of accuracy. Thus, this study focuses on a detection approach for identifying weather trends based on data mining techniques.

DOI: https://doi.org/10.36548/jismac.2023.1.004
Received: 11.03.2023, received in revised form: 11.04.2022, accepted: 26.04.2023, published: 06.05.2023

2. Design of System

A. Existing System

Many studies have been conducted using various methods to enhance our understanding of weather change patterns, and they have made significant contributions. The use of data mining techniques in this research area hasn't been widely adopted, though. The current system required and still requires a significant investment of time, money, talent, and technology. It entails using sophisticated models that produced incorrect predictions. As a result, the model implemented in this work is simple, and provides some valuable information. It is more accurate when compared to the existing ones.

B. Proposed system

The dataset is first taken and then analysed.Then, the dataset is pre-processed to make it easier for further processing. Once this is done, the dataset is implemented with a supervised anomaly detection algorithm to detect anomalies if any. The supervised algorithms include Bayesian network and Neural network.

3. Requirements

The hardware and software requirements of the system and the functional and non-functional requirements of the system are summarized below.

A. Hardware Requirements

TABLE I. HARDWARE REQUIREMENTS

Processor	1 GHz to 2 GHz
Hard drive	1 GB
Memory (RAM)	Minimum 10 GB, recommended 15 GB or more

B. Software Requirements

TABLE 2. SOFTWARE REQUIREMENTS

Operating system	Windows 7 or above
Environment	Python IDLE (version 3.6 or above)

C. Functional requirements

Data pre-processing requirements: The dataset obtained/downloaded should be pre-processed (i.e., it removes all the unwanted attributes from the dataset).

Training requirements: The system should ensure that the neural network and Bayesian model chosen should provide results with highest accuracy.

D. Non-Functional Requirements

a) Usability: The system should be easy to use, because the user should be able to understand algorithms and train the dataset.

b) Reliability: This software is developed with machine learning and deep learning techniques. So, the data is used to compare the result and measure reliability. The maintenance period should not be a matter of concern because the reliable version always runs on the server. The users can use the system at any time, hence the maintenance is guaranteed.

c) Performance: The amount of wrongly predicted weather should be minimal

d) Supportability: The system should require Python knowledge for maintenance. If a problem arises in machine learning methods, it requires code knowledge and machine learning background to solve

4. Literature Review

In [1], by examining gathered physiological data from medical sensors, a novel method for detecting sensor abnormality was provided. The technique's goal was to properly

discriminate between false alerts and actual alarms. Wireless Sensor Networks (WSN) are subject to different sensor failures and incorrect measurements. This flaw prevents quick and effective responses in different WSN applications, including healthcare. For instance, inaccurate readings can result in false alarms that may necessitate unnecessary medical assistance. As a result, a strategy to distinguish between legitimate medical issues and false alarms will enhance remote patient monitoring systems and the standard of healthcare provided by WSN. The methods presented were:

1) Sequential minimal optimization regression (SMO regression) for predicting sensor value.

2) Dynamic threshold for error calculation.

3) Majority voting for decision on whether to generate alarm.

The accuracy of the suggested sensor anomaly and true/false alarm detection methods was investigated using real medical datasets in a Java context. The regression features of the WEKA tool were employed for the prediction portion. Each parameter's sensed value and forecasted value were compared. The SMO regression approach was used to generate the prediction model, which is based on historical data.

In [2], the solution, which is centered on detecting anomaly and covers the detection of faults and events, was presented. WSNs have become widely used in a variety of areas recently, including manufacturing, the military, healthcare, environmental monitoring, and defense. They are typically constrained by the availability of energy, computing power, storage space, and the communication bandwidth because they are made up of numerous nodes dispersed across a vast geographic region. Detecting abnormalities and determining the occurrence of an event is a fundamental challenge in WSN applications. The hostile environment, inadequate node energy, software and hardware failure, etc. are a few examples of this instability. The study proposed a Fault-Tolerant Anomaly Detection (FTAD) method based on sensor network spatial-temporal correlation. The methods were:

1) Temporal Correlation

2) Spatial Correlation

In this research, for event detection in WSNs, a FTAD based on spatial-temporal correlation is proposed. The strategy attempts to enhance event detection quality and provide

a fix for the issue that occurs when the sensor defect probability reaches a critical level: event detection ability quickly diminishes. The experiment demonstrates that FTAD performs well even when there is a large failure rate and that it is more effective at detecting events. Both low-density heterogeneous networks and high-density sensor networks can use this technique. Although this system is capable of accurately detecting a single event, further research is required to see how well it performs in multi-modality event detection. Future work will concentrate on perfecting the suggested technique to satisfy multi-modality event detection in WSNs.

5. Analysis and Design

A. Overview

The study consists of the following phases:

1) Dataset pre-processing
2) Feature selection
3) Anomaly detection with Neural network
4) Anomaly detection with Bayesian model

Initially, the dataset is pre-processed to make it easier for further processing. This can be done only when the dataset is analyzed. Once this is done, the dataset is implemented with a supervised anomaly detection algorithm to detect anomalies if any. The study is implemented with various supervised algorithms like Bayesian network and Neural network.

B. Architecture of the system

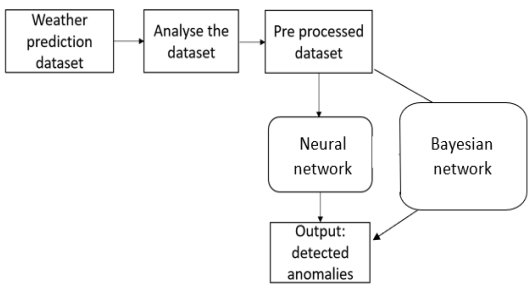

Figure 1. Bayesian Model

C. Dataset pre-processing

The study's dataset is weather AUS, which includes daily observation from numerous Australian weather stations in CSV file format. This dataset contains 24 attributes. The attributes are:

Date - The observation date

Location – Weather station location

Min-Temp – Temperature (Minimum) in Celsius

Max-Temp – Temperature (Maximum) in Celsius

Rainfall – Amount of rainfall recorded in mm

Evaporation – PAN evaporation (class A in mm)

Sunshine – Sunshine hours

Wind-GustDir - The direction of wind gust (strongest) in 24 hours

Wind-GustSpeed - The speed of wind gust (strongest) in 24 hours in kmph

WindDir-9am – wind direction at 9am

WindDir-3pm – wind direction at 3pm

WindSpeed9am – The average wind speed over 10 mins, prior to 9 am, in kmph

WindSpeed3pm - The average wind speed over 10 mins, prior to 3pm, in kmph

Humidity9am - Humidity at 9am in percentage

Humidity3pm - Humidity at 3pm in percentage

Pressure9am - Atmospheric pressure (hpa) reduced to mean sea level at 9am

Pressure3pm - Atmospheric pressure (hpa) reduced to mean sea level at 3pm

Cloud9am - Fraction of sky obscured by cloud at 9 pm (in "oktas": eighths)

Cloud3pm - Fraction of sky obscured by cloud at 3pm (in "oktas": eighths)

"Oktas," a unit of eights, are used to measure this. It shows how many sky eights are blocked by clouds. A score of 0 denotes a totally clear sky, while an 8 denotes a completely clouded sky.

Temp9am - Temperature at 9am in Celsius

Temp3pm - Temperature at 3pm in Celsius

RainTodayBoolean - 1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0.

RISK_MM - The amount of next day rain in mm. Used to create response variable RainTomorrow. A kind of measure of the "risk".

RainTomorrow - The target variable; denotes if it will rain tomorrow or not.

The pre-processing of the dataset starts with dropping the rows with null values. The unwanted columns dropped are date, location, windDir9am, and windDir3pm. Next, the categorical values are converted to numeric values. WindGustDir contains various values (W, WNW, NNW, SSE, N, S, E, etc.). These values are converted into meaningful and useful data by extrapolating into 7 other different columns.

fN – indicating ‘From North’

fS – indicating ‘From South’

fW – indicating ‘From West’

fE is ignored since if all the other columns are 0, it is automatically assumed From East.

The same is repeated to extrapolate meaning from the end direction.

tN – indicating ‘to North’

tS – indicating ‘to South’

tW – indicating ‘to West’

tE- indicating ‘to East’. This column is not ignored here, since sometimes there is no end direction.

Table 1. Sample attributes' conversion to binary values

	WindGustDir	fN	fS	fW	tN	tE	tS	tW
0	SSW	0	1	0	0	0	0	1
1	S	0	1	0	0	0	0	0
2	NNE	1	0	0	0	1	0	0
3	WNW	0	0	1	0	0	0	1
4	WNW	0	0	1	0	0	0	1

D. Feature selection

Feature selection is a process where the most important feature that determines the target variable is selected. Feature selection is done using Pearson correlation. Correlation of every feature in the dataset with the target variable is calculated. The most correlated features with the target variable are selected. The selected features are Humidity3pm, Cloud9am and Cloud3pm.The correlations are:

Humidity3pm – 0.455

Cloud3pm – 0.388

Cloud9am – 0.323

E. Anomaly Detection with Neural Network

Neural network model is a better prediction model when compared to others. The model used is the built-in MLP classifier from the package neuralnetwork in sklearn. It is one of the supervised learning algorithms. It is a class of feedforward artificial neural network. It calls a function detectanomaly() to detect the anomaly for a given row. The detectanomaly() function calculates the threshold from the mean value of the features. Anomaly is detected for the three selected features.

F. Anomaly Detection with Bayesian Model

The model used is the built-in Gaussian Naive Bayes classifier from the package naïve_bayes in sklearn. It calls a function detectanomaly() to detect the anomaly for a given row. The same detectanomaly() function calculates the threshold from the mean value of the features. Anomaly is detected for the three selected features.

6. Results

The accuracy of the two models show that Neural network is better than Gaussian network although the accuracy is more or less the same. And the mean squared error proves that using neural network is the best way.

```
Neurl Network

--------------------------------

The geometric mean is 0.9999435267541552
mean_squared_error
0.009413875735476427
Index(['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am',
       'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm',
       'Temp9am', 'Temp3pm', 'RainToday', 'RISK_MM', 'fN', 'fS', 'fW', 'tN',
       'tE', 'tS', 'tW', 'TargetVariable'],
      dtype='object')
Accuracy: 0.999911378943637
```

Figure 2. Image depicting accuracy and mean squared error for Neural network

```
Bayesian Network

--------------------------------

The geometric mean is 0.961654699234709
mean_squared_error
0.2429436633002651
Index(['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am',
       'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm',
       'Temp9am', 'Temp3pm', 'RainToday', 'RISK_MM', 'fN', 'fS', 'fW', 'tN',
       'tE', 'tS', 'tW', 'TargetVariable'],
      dtype='object')
Accuracy: 0.9409783764622475
```

Figure 3. Image depicting accuracy and mean squared error for Bayesian network

7. Conclusion

In this research, both the Gaussian and Neural network models are implemented for anomaly detection and the results are compared. The experimental results have proved that Neural network is better than the Gaussian model with more accuracy and less mean squared error. Using two completely different anomaly detection methods, a very robust method to detect anomalies has been proposed. These methods have been implemented in python using built-in classifiers in COLAB. This work can be further improved using deep learning and other supervised methods. With access to more features, deep learning could give more accurate results. The detected anomalies can also be replaced with a much meaningful value by using some data recovery techniques.

References

1) Haque SA, Rahman M, Aziz SM. " Sensor anomaly detection in wireless sensor networks for healthcare", Sensors (Basel), Apr15;15(4):8764-86, 2015.

2) Peng N, Zhang W, Ling H, Zhang Y, Zheng L, "Fault-Tolerant Anomaly Detection Method in Wireless Sensor Networks", Information, 9(9):236, 2018.

3) Guo, J., Takahashi, N., Nishi, T, "A Novel Sequential Minimal Optimization Algorithm for Support Vector Regression", Lecture Notes in Computer Science, vol 4232. Springer, 2006.

4) W. Gong et al., "A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion," Sensors 19(7), 1693 (2019).

5) Jaouher Ben Ali, Nader Fnaiech, LotfiSaidi, Brigitte Chebel-Morello, and Farhat Fnaiech. Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. Applied Acoustics, 89:16–27, 2015.

6) Turker Ince, Serkan Kiranyaz, LeventEren, Murat Askar, and MoncefGabbouj. Real-time motor fault detection by 1-d convolutional neural networks. IEEE Transactions on Industrial Electronics, 63(11):7067– 7075, 2016.

7) Jiedi Sun, Changhong Yan, and Jiangtao Wen. Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning. IEEE Transactions on Instrumentation and Measurement, 67(1):185–195, 2017.

8) RoozbehRazavi-Far, Maryam Farajzadeh-Zanjani, ShokoofehZare, Mehrdad Saif, and JafarZarei. One-class classifiers for detecting faults in induction motors. In 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), pages 1–5. IEEE, 2017.

9) Feng Jia, Yaguo Lei, Jing Lin, Xin Zhou, and Na Lu. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data.Mechanical Systems and Signal Processing, 72:303–315, 2016.