2011 International Symposium on INnovations in Intelligent SysTems and Applications

Organized by

In cooperation with

Sponsors

15-18 June 2011, İstanbul-Kadıköy, TURKEY

Anomaly Detection in Temperature Data Using DBSCAN Algorithm

Mete ÇELİK
Dept. of Computer Engineering
Erciyes University
38039 Kayseri, Turkey
mcelik@erciyes.edu.tr

Filiz DADAŞER-ÇELİK
Dept. of Environmental Engineering
Erciyes University
38039 Kayseri, Turkey
fdadaser@erciyes.edu.tr

Ahmet Şakir DOKUZ
Dept. of Computer Engineering
Erciyes University
38039 Kayseri, Turkey
dokuz.a.sakir@gmail.com

Abstract—**Anomaly detection is a problem of finding unexpected patterns in a dataset. Unexpected patterns can be defined as those that do not conform to the general behavior of the dataset. Anomaly detection is important for several application domains such as financial and communication services, public health, and climate studies. In this paper, we focus on discovery of anomalies in monthly temperature data using DBSCAN algorithm. DBSCAN algorithm is a density-based clustering algorithm** 基于密度的聚类算法 **that has the capability of discovering anomalous data. In the experimental evaluation, we compared the results of DBSCAN algorithm with the results of a statistical method. The analysis showed that DBSCAN has several advantages over the statistical approach on discovering anomalies.**

Keywords-anamoly detection in time series data;DBSCAN algorithm; temperature data

I. INTRODUCTION

Anomaly detection can be defined as “the problem of finding patterns in data that do not conform to expected normal behavior” [1]. Anomaly detection is important when the abnormal behavior in the dataset provides significant information about the system. Anomalies can be caused by malicious activities, instrumentation errors, changes in the environment (i.e., climate change), and human errors [1].

Anomaly detection is an important problem in several application domains such as credit card fraud detection in financial systems, intrusion detection in communication systems, and contagious disease detection in public health data. In recent years, anomaly detection has also become an important problem in climate studies for detecting abnormal climatic conditions caused by global warming. Anomalies in a climate data series can be explained by the following example. In a time series of temperature data, one can see fluctuations 波动 between high and low temperatures. The normal behavior for temperature data is to reach high values (e.g., temperature over 20 ^{o}C) in summer months and drop to low values (e.g., temperature below 10 ^{o}C) in winter months. In a temperature dataset, a winter month with average temperature above 20 ^{o}C or a summer month with average temperature below 10 ^{o}C can be identified as an anomaly. In Fig. 1, on a sample monthly temperature data series, the data point, marked as 1, can be called an anomaly.

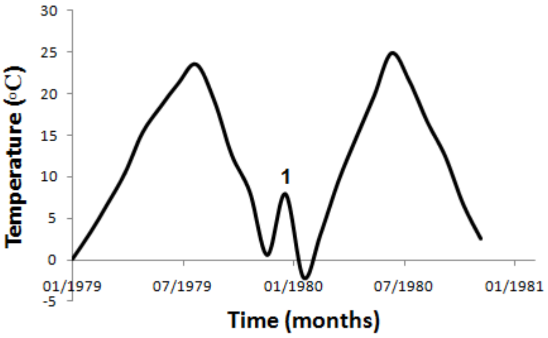


Figure 1. Representation of anomalies on sample monthly temperature data

One of the major challenges of anomaly detection is to characterize normal and abnormal behaviors. Other challenges can be listed as to develop techniques to discover abnormal data and to design computationally-efficient algorithms.

The aim of this study is to discover anomalies on temperature data. DBSCAN, a density-based clustering algorithm, is used as an anomaly detection technique and the results of DBSCAN were compared with the results of a statistical anomaly detection method. The dataset used in this study consists of daily average temperature data collected at station number 17836 (Develi Station in Kayseri, Turkey) operated by Turkish State Meteorological Service. The daily data were converted to monthly averages by averaging daily values for each month. The data were available for a 33-year period (from 1975 to 2008).

II. RELATED WORK

In the literature, there are a number of extensive reviews discussing anomaly detection approaches. Kumar et al. [1] provided a recent review of the anomaly detection problems, techniques, and application areas. Hodge et al. [2] provided a review of anomaly detection techniques. The

978-1-61284-922-5/11/$26.00 ©2011 IEEE

advantages and disadvantages, and the motivation of these techniques were discussed in a comparative approach. Tang et al. [3] reviewed the different solution methods of anomaly detection problems, i.e., density-based, connectivity-based methods, and the theoretical structure of the anomaly detection problems. Petrovskiy [4] discussed the anomaly detection algorithms, basic approaches, the advantages and disadvantages of these approaches, and proposed a new fuzzy-based anomaly detection algorithm.

The anomaly detection techniques can be classified as statistical approaches and distance-based approaches. Statistical approaches aim to develop a statistical model of the data and identify data that do not fit into the model. In distance-based approaches, the distances between data are considered in detection of anomalies: the data at a distance greater than a pre-defined distance is called an anomaly. This study focuses on detecting outliers using distance-based approaches.

离群值

DBSCAN, which was developed by Ester et al. [5], is one of the distance-based approaches, which has been widely used for solving anomaly detection problems. Ester et al. [5] showed that DBSCAN is much more powerful algorithm than the other algorithms on mass and dense datasets for finding anomalies.

In the study, the DBSCAN algorithm is used for detection of anomalies in monthly temperature data and this method is compared with a statistical anomaly detection method.

III. DBSCAN Algorithm

DBSCAN is a density-based spatial clustering algorithm that can also define anomalies in the data series. It requires two user-defined parameters, which are neighborhood distance *epsilon* (*eps*) and minimum number of points *minpts*. For a given point, the points in the *eps* distance are called neighbors of that point. If the number of neighboring points of a point is more than *minpts*, this group of points is called a cluster.

DBSCAN labels the data points as core points, border points, and outlier (anomalous) points. Core points are those that have at least *minpts* number of points in the *eps* distance. Border points can be defined as points that are not core points, but are the neighbors of core points. Outlier points are those that are neither core points nor border points.

In DBSCAN, the clustering approach is different from typical clustering approaches. DBSCAN can define outlier (anomalous) points that do not fit to any clusters. In Fig. 2, the groups represent results of a clustering approach that takes distance threshold *eps* as a cluster metric. In addition to distance metric, DBSCAN requires minimum number of points *minpts* inside a group to label it as a cluster. For example, if the *minpts* value is selected as 3, *group 1, group 2, group 4*, and *group 6* will be marked as clusters using DBSCAN. In contrast, *group 3* and *group 5* will be defined as outliers since they do not contain sufficient number of points to form a cluster. Similarly, if the *minpts* value is selected as 5, *group 3, group 5,* and *group 6* will be assigned as outliers using DBSCAN.

距离度量

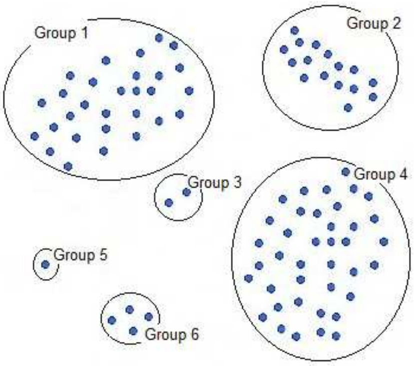


Figure 2. A sample dataset for distance-based anomaly detection approach

The pseudo-code of DBSCAN algorithm is given in Algorithm 1. The inputs of the algorithm are dataset and user-defined *eps* and *minpts* parameter values.

Algorithm 1. The pseudo code of DBSCAN algorithm

```
Inputs:
D: the dataset
Eps: the neighborhood distance
Minpts: the minimum number of points
Output:
Discovered outliers and clusters
Variables:
m, n: row and column values of D matrix, respectively
Dist: distance vector
indices: indices that distance of points is lower than Eps
class_no: indicates the clusters – default 1

Algorithm:
import the data-set into D
for i = 1 to m    //row counter
    Dist = distance(i, D)
    neighbors= find(Dist =< Eps)
    neighbor_count = count(neighbors)
    core_neig=check_core_neighbor (neighbors)
    if (neighbor_count >=minpts)
        class(i) = class_no    //clustered point
        while(more points near i)
            class(point) = class_no
        end while
        class_no += 1
    else if(neighbor_count<minpts & core_neig==True)
        class(i) = 0          //border point
    else if (neighbor_count<minpts)
        class(i) = -1         //outlier point
    end if
end for
return class
```

In the algorithm, between steps 2 and 18, there is a for-loop running for number of points in the dataset. In step 2, DBSCAN assigns *D[i]* as a center point. In step 3, the distances between center point *D[i]* and remaining points are calculated, and then in step 4 the points whose distances are less than or equal to *eps*, are accepted as neighbors of the center point. In step 5 the number of neighbors of the center point is calculated. Step 6 of the algorithm checks if there is any core point in the neighbor list. In step 7, the algorithm checks the number of neighbor points, *neighbor_count*, to test if it is greater than or equal to *minpts*. If it is, the center is determined as a core point. Between steps 8 and 12, a unique class number is assigned to the core point *D[i]* and its neighbors. If the center point is not a core point but near a core point, it is defined as a border point between steps 13 and 15. If the center point is neither a core point nor a border point, and distance of the points are greater than *eps*, it is labeled as outlier. Finally, in step 19 the algorithm outputs the results.

IV. Deseasoning Time Series Data

In this study, the aim is to discover anomalies out of monthly temperature data. As temperature data have strong seasonal component, the series should be deseasoned in a pre-processing step before applying any of the anomaly detection techniques.

In the pre-processing step, monthly z-score technique, explained by Kumar et al [6], was applied (Algorithm 2). The aim of monthly z-score technique is to eliminate the seasonal component from the dataset to obtain a deseasoned dataset. The *monthly_z_score* function first splits the dataset into months. For each month, the mean and the standard deviation 标准偏差 are calculated. The value of each month is than normalized using the mean and standard deviation for that month. Fig. 3 shows the original and deseasoned temperature data used in this study. The mean and standard deviation of the original data were 10.96 °C and 8.73 °C, respectively. The mean and standard deviation of the deseasoned data were changed to 0 °C and 1 °C, respectively. The temperature values in the original data were in the range of -8.25 – 26.55 °C. The temperature values in the deseasoned data were in the range of -2.6 – 3.11 °C.

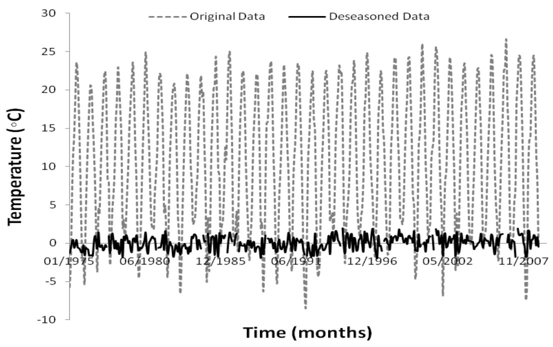


Figure 3. The original data and deseasoned temperature data

Algorithm 2. The pseudo code of the pre-processing step

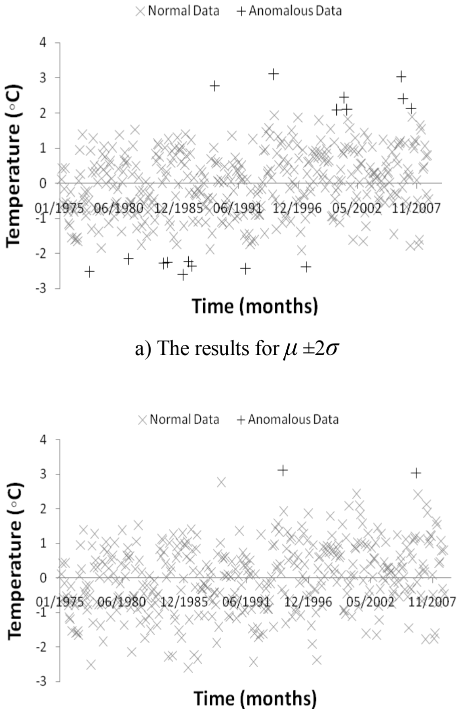
```
Input:
X: The dataset
Output:
Deseasoned data vector deseasoned_X
Variables:
Deseasoned_X: The deseasoned data vector.
Pre-Processing function:
1.      initialization
2.      deseasoned_X = monthly_z_score(X)
```

V. Experimental Evaluation

To evaluate the experimental results of the DBSCAN algorithm, we compared it with a statistical anomaly detection method. In the experiments, we used the deseasoned temperature time series data.

A. Detecting Anamolies using Statistical Method

The basic assumption of the statistical method is that the data are from a normal 正态分布 distribution. The normal distribution has two parameters, the mean (μ) and the standard deviation (σ). According to Tan et al. [7] the chance that a value is located at the tails of the distribution is very low. Based on that, it is a general practice to identify the data points greater than "μ+2σ" or "μ+3σ" and smaller than "μ-2σ" or "μ-3σ as anomalies.



a) The results for $\mu \pm 2\sigma$

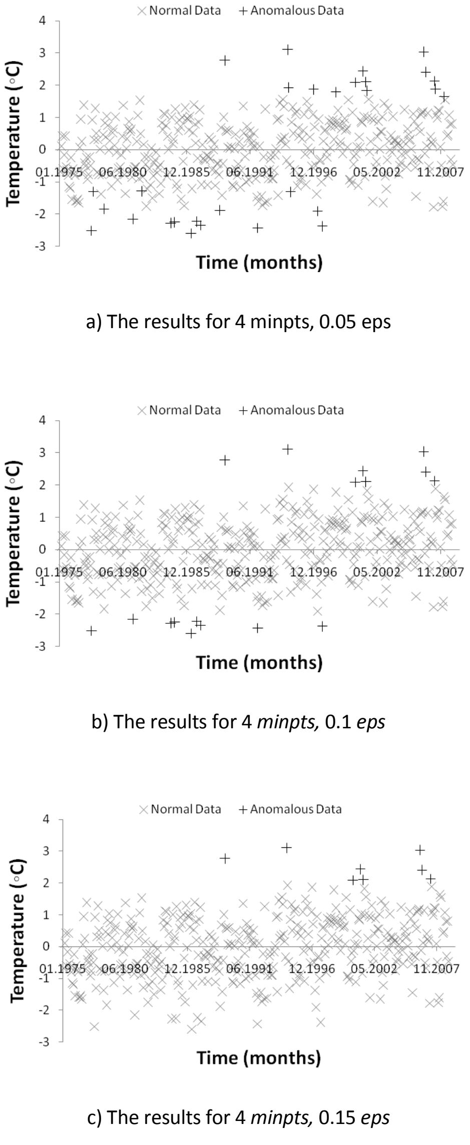
b) The results for $\mu \pm 3\sigma$

Figure 4. Anomalies detected by the statistical approach. ((+) represents anomolous points and (x) represents normal points)

As can be seen from Fig. 4, the statistical method detects anomalous points as the points which are smaller or greater than certain temperature values.

B. *Detecting Anamolies using DBSCAN Algorithm*

The aim of the DBSCAN algorithm is to discover abnormal points that do not fit any of the clusters. In the experiments we evaluated the effects of the values of the *eps* and *minpts* parameters using the deseasoned dataset.



a) The results for 4 minpts, 0.05 eps

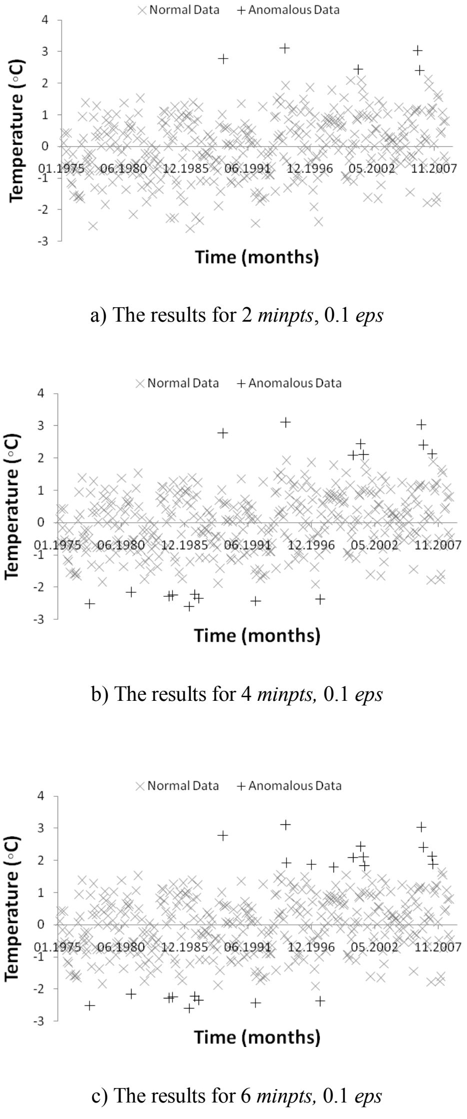
b) The results for 4 *minpts,* 0.1 *eps*

c) The results for 4 *minpts,* 0.15 *eps*

Figure 5. Anomalies detected by the DBSCAN algorithm for 4 *minpts* and 0.05, 0.1, and 0.15 *eps*. ((+) represents anomolous points and (x) represents normal points)

To evaluate the effect of the *eps* parameter on discovering anomalies, we set *minpts* parameter to 4 and conducted experiments with 0.05, 0.1, and 0.15 *eps* values. Fig. 5 shows the results of the DBSCAN algorithm for different *eps* values. As can be seen in Fig. 5, as the value of the *eps* parameter increases, the number of anomalies detected by the algorithm decreases. The results reported in Fig. 5b and Fig. 5c are somewhat similar with the results found with the statistical method. That is, anomalies were discovered above or below a certain temperature values. In the case of Fig 5a, DBSCAN algorithm discovered anomalies in between normal points and so there is no linear separation between normal points and abnormal points as the statistical method showed.

To evaluate the effect of the *minpts* parameter on discovering anomalies, we set *eps* to 0.1 and conducted experiments with *minpts* of 2, 4, and 6. In Fig. 6, an increase in the number of anomalous points can be observed as the number of *minpts* increases. This situation occurs because DBSCAN assigns a point to a cluster only if there are at least *minpts* number of neighbor points in the *eps* distance. When *minpts* is higher, the chance of having a neighbor point within the *eps* distance is lower.



a) The results for 2 *minpts*, 0.1 *eps*

b) The results for 4 *minpts,* 0.1 *eps*

c) The results for 6 *minpts,* 0.1 *eps*

Figure 6. Anomalies detected by the DBSCAN algorithm for 0.1 eps and 2, 4, and 6 *minpts*. ((+) represents anomolous points and (x) represents normal points)

Overall, the experimental results showed that the anomalous points detected by the statistical method are based on their value. In other words, the statistical method can detect anomalous points which are above and below a certain threshold (extremes). However, anomalous points are not only extreme points but also they are the data that do not occur frequently. DBSCAN algorithm discovers these kinds of anomalies as well as extremely low and high values.

VI. CONCLUSION AND FUTURE WORKS

We applied DBSCAN algorithm for detecting anomalies in time series data and compared this method with a statistical anomaly detection method. Because of the temporal nature of dataset we applied a pre-processing step to remove seasonality before applying DBSCAN algorithm on the dataset. The results show that DBSCAN algorithm can discover anomalies even if they are not extreme values.

In the future, we plan to apply other anomaly detection techniques and to test the performances of the algorithm with different parameter values.

ACKNOWLEDGMENT

This is study was partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) (Project no: CAYDAG 110Y110) and Research Fund of the Erciyes University (Project no: FBA-09-866).

REFERENCES

[1] V. Kumar, A. Banerjee, and V. Chandola, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, July 2009.

[2] V. Hodge and J. Austin, "A survey of outlier detection methodologies," Artificial Intelligence Review, vol. 22, pp. 85-126, October 2004.

[3] J. Tang, Z. Chen, A. W.-c. Fu, and D. Cheung, "A robust outlier detection scheme for large datasets," in *Knowledge Discovery and Data Mining*, Pacific-Asia Conf., vol. 6, pp. 6-8, 1996.

[4] M. Petrovskiy, "Outlier detection algorithms in data mining systems," Programming and Computing Software, vol. 29, pp. 228-237, July-August 2003.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density based algorithm for discovering clusters in large spatial databases with noise," in *KDD-96 Proceedings*, pp. 226-231, 1996.

[6] V. Kumar, P.-N. Tan, and M. Steinbach, "Finding spatio-temporal patterns in earth science data," in *KDD 2001 Workshop on Temporal Data Mining, August 2001*.

[7] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston Addison-Wesley, April 2005.