Detection of Outliers in Time Series Data

Samson Sifael Kiware
Marquette University

DETECTION OF OUTLIERS IN TIME SERIES DATA

by

Samson Kiware, B.A.

A Thesis Submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

May 2010

ABSTRACT
DETECTION OF OUTLIERS IN TIME SERIES DATA

Samson Kiware, B.A.

Marquette University, 2010

This thesis presents the detection of time series outliers. The data set used in this work is provided by the GasDay Project at Marquette University, which produces mathematical models to predict the consumption of natural gas for Local Distribution Companies (LDCs). Flow with no outliers is required to develop and train accurate models. GasDay is using statistical approaches motivated by normally distributed samples such as the $3-\sigma$ rule and the $5-\sigma$ rule to aid the experts in detecting outliers in residuals from the models. However, the Jarque-Bera statistical test shows that the residuals from the GasDay models are not normally distributed.

We present an explanation of Density Based Spatial Clustering of Applications with Noise (DBSCAN) and how it is used to detect time series outliers. We have introduced a new application for the DBSCAN algorithm by adapting it to detect outliers in natural gas flow. The performance of DBSCAN is compared with GasDay's existing technique. Five data sets from temperature-sensitive operating areas with identified outliers and 1000 data sets with synthetic outliers are used in the evaluation process. The 1000 synthetic data sets are prepared using the same empirical distribution as one of the identified data set. This work indicates that DBSCAN has shown some improvement in detecting outliers over GasDays existing technique and merits further exploration.

ACKNOWLEDGEMENTS

Samson Kiware

I wish to express my gratitude to many people who have provided me with crucial help and support while carrying out the research for this thesis. Without the help and encouragement that my advisor, Dr. George Corliss has provided, none of this would of been possible. He has spent countless hours in discussion of this project and provided me with ideas while keeping me on the right track. Dr. Corliss, Dr. Craig Struble, Dr. Stephen Merrill, and Dr. Praveen Madiraju, my committee members, have contributed a lot in different ways to the accomplishments of this work. I thank them for hours they spent in discussion of this project and the ideas they provided. The project carried on this work originated from a simple idea Dr. Struble and I discussed when I took his Data Mining course. I thank Dr. Ronald Brown, The GasDay Project Director, for providing me with some of the data sets and for spending his countless hours in discussion of this project. They have been my teachers and mentors, by providing much help, not only through this work, but also in and out of the classroom.

I also thank Mr. Steve Vitullo, Mr. Sakauchi Tsuginosuke, Mr. Nathan Wilson, Ms. Navneet Dhillon, and other GasDay student researchers for sharing ideas and technical help during the course of this project. Students who work at the GasDay

research Lab highly recognize the help and encouragement we get from Mr. Thomas Quinn, Business Director, and Ms. Paula Gallitz, Project Cordinator. I want to say thanks to both of them for their help through my graduate studies.

I must also thank my blood family in Tanzania and my American family (Pr. Viviane and Pr. Fred) for their motivational and financial support throughout. Finally, I'm thankful to Dr. Ronald Brown, Dr. George Corliss, Mr. Thomas Quinn, and Marquette University for giving me the opportunity to be a part of the graduate student body and for their financial support, without which none of this work could have been completed.

DEDICATION

This work is dedicated to my family, Kiware, Eliaita, Lilian, Frida, and Ndelilio for your love, motivation, inspiration, and support throughout this passage. Especially my late sister Neema, I will always love you, we miss you a lot.

TABLE OF CONTENTS

TABLE OF CONTENTS — *Continued*

LIST OF FIGURES

LIST OF FIGURES — *Continued*

CHAPTER 1

INTRODUCTION TO NATURAL GAS FLOW

Chapter one presents the GasDay project (1; 2) at Marquette University, which has provided the data set (natural gas flow) used by this thesis. It provides a background for outlier detection in gas flow and discusses GasDay's mathematical models and residuals from the models. We give mathematical and business statements of the problem to be considered by this work. The chapter introduces the performance evaluation of an outlier detection technique, and the organization of the rest of the thesis is provided.

1.1 The GasDay Project

GasDay Project at Marquette University produces mathematical models to predict the consumption of natural gas for Local Distribution Companies (LDCs). Using inputs such as past weather data, previous flow, and current weather forecasts, GasDay models make accurate gas flow forecasts that save LDCs time, money, and effort (1). The Gasday project receives daily flow files from regional sets of customers. We call these regional areas operating areas. The GasDay project works with two types

of regional areas: temperature-sensitive and non-temperature-sensitive operating areas. Temperature-sensitive operating areas use natural gas primarily for heating space, while customers in non-temperature-sensitive operating areas use natural gas primarily for other purposes, especially commercial or industrial processes. In this thesis, we refer to temperature-sensitive and non-temperature-sensitive operating areas by the generic names of JOTO and BARIDI, respectively.

We present several figures to help visualize the relationship between temperature and flow for both types of operating areas. First, consider temperature and flow time series. Figure 1.1 shows a temperature time series, showing higher temperatures in the summer and lower temperatures in the winter. Figure 1.2 shows corresponding time

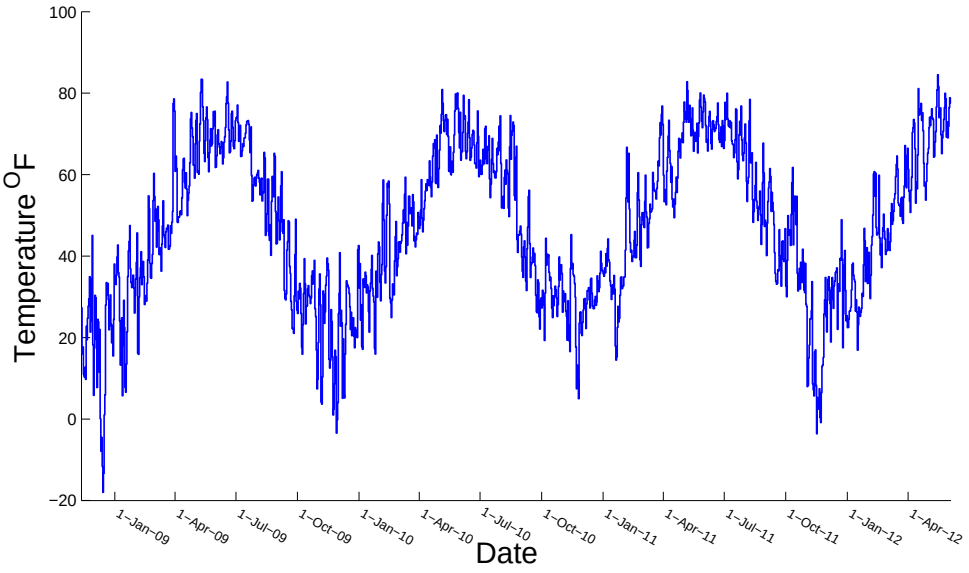

Figure 1.1: Temperature time series plot.

series for natural gas flow, with higher consumption during the winter and lower consumption during the summer. Figure 1.3 shows flow versus 65°F minus daily temperature (Heating Degree Day, HDD) for a JOTO (temperature sensitive). It shows significant variation of gas consumption versus HDD for an operating area. Flow is nearly constant and then starts to increase linearly with increasing HDD.

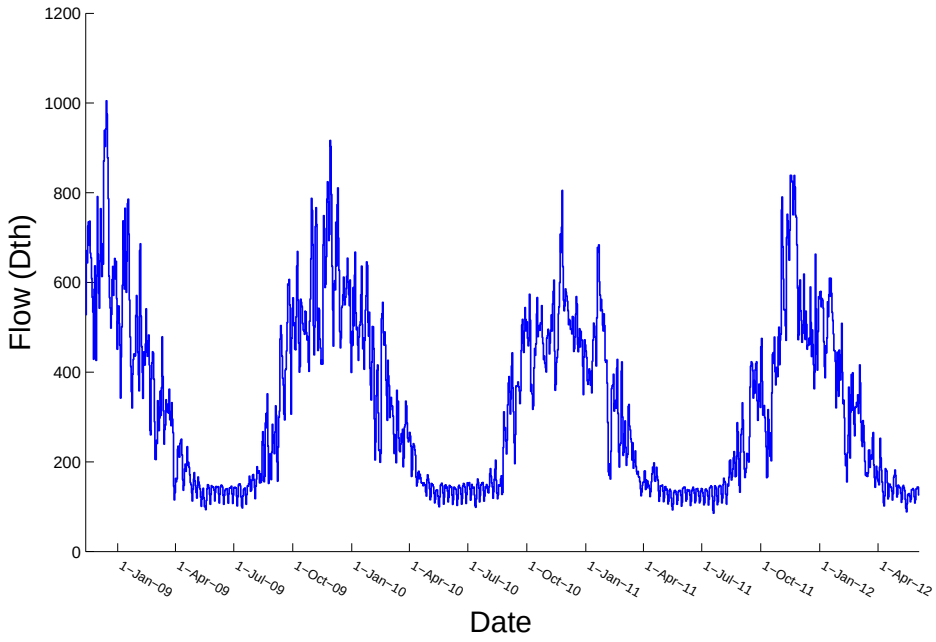

Figure 1.2: Gas flow for a JOTO.

Figure 1.4 shows gas flow for a typical BARIDI (non-temperature sensitive) composed primarily of industrial customers. The consumption is not higher during the winter and lower during the summer, but varies throughout the year. Figure 1.5 does not show significant variation of gas consumption versus HDD. We cannot observe a

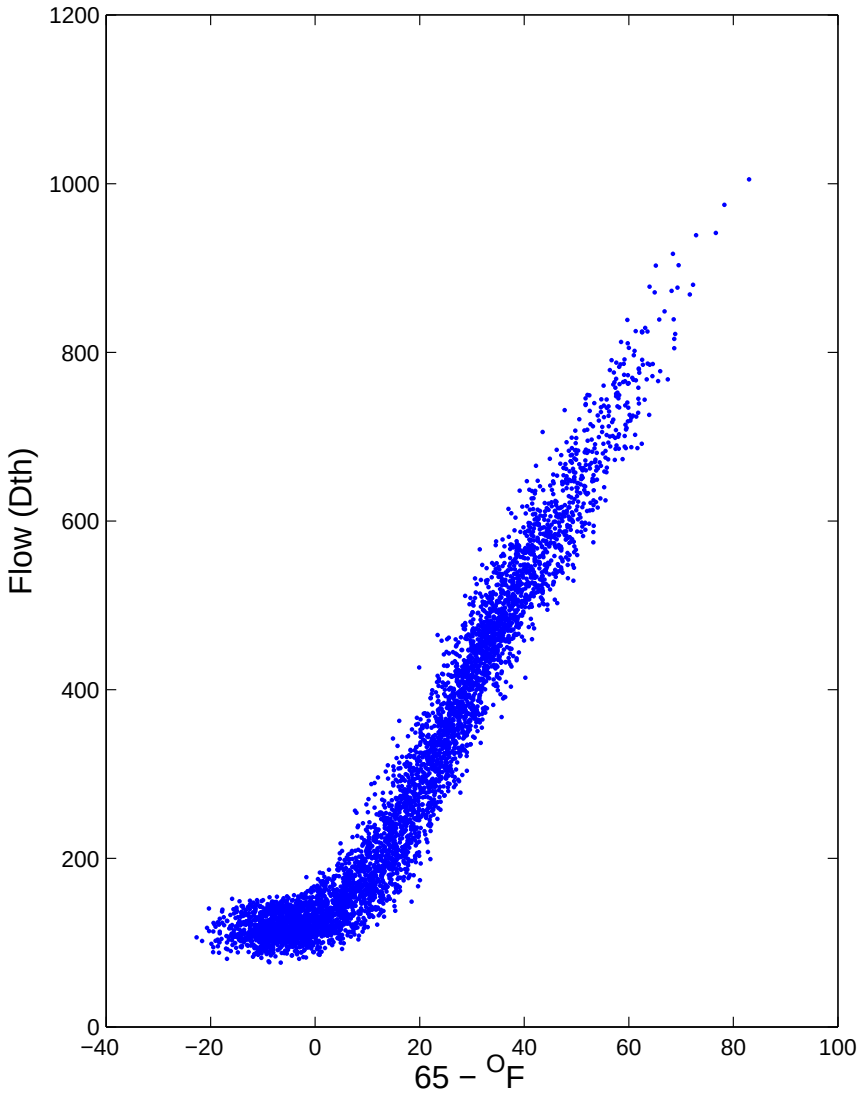

Figure 1.3: Scatter plot of flow consumption vs, HDD for a typical JOTO

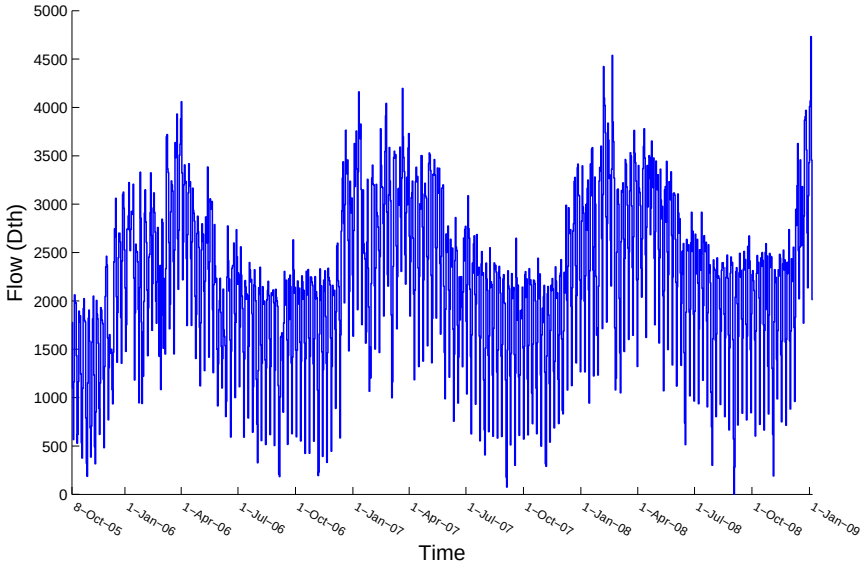

Figure 1.4: Gas flow for a BARIDI

linear relationship similar to that in Figure 1.3 because natural gas consumption is not highly affected by temperatures for industrial customers comprising a BARIDI. These plots help us understand the data sets used in this work as provided by the GasDay project.

This thesis presents techniques used to detect outliers in time series data. Daily flow (real-time data) and weather are required as inputs into GasDay models (explained in Section 1.3) when packaged and ready to use by customer. We develop a technique that can be implemented by the GasDay project to detect incorrect data in both historical and real-time data. The focus is natural gas flow data for operating areas showing significant variation of gas consumption with temperature as shown in Figure 1.3.

The rest of this chapter is organized as follows. In Section 1.2, we provide the

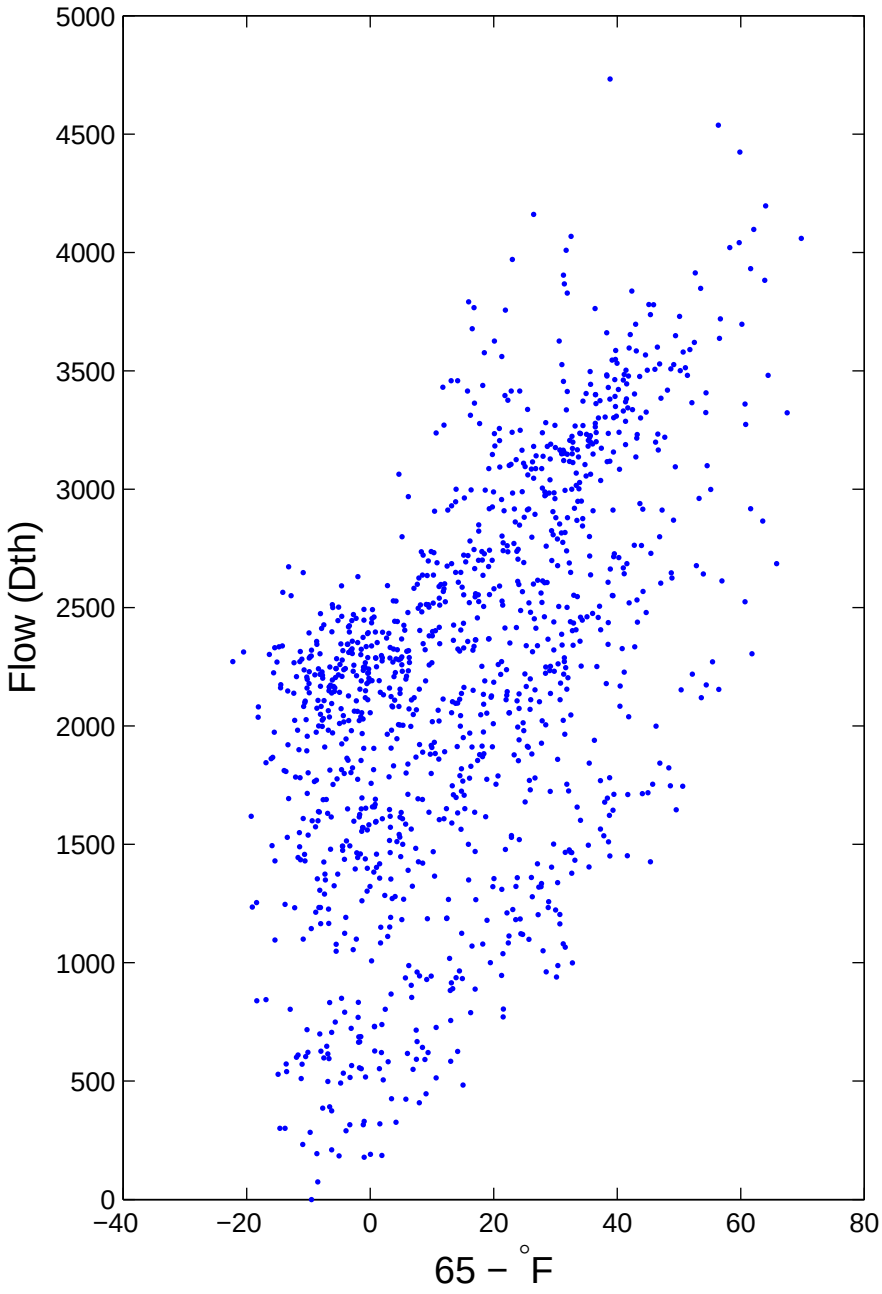

Figure 1.5: Scatter plot of flow consumption vs, HDD for a BARIDI

background of outlier detection in gas flow. GasDay's mathematical models are presented in Section 1.3. The statement of the problem addressed by this research is stated in Section 1.4. Section 1.5 introduces the evaluation data set used to evaluate the performance of outlier detection techniques.

1.2 Outlier Detection in Gas Flow

In this Section, we discuss the problem of outlier detection in natural gas consumption time series. An outlier is an entry in a data set that is anomalous with respect to the behavior seen in the majority of the other entries in the data set (3; 4; 5). The data sets used in this thesis are provided by the GasDay project. Correct data is required to develop and train accurate models. There is not a clear way of knowing correct flow to be able to give a clear definition of a true outlier in flow data processed by the GasDay project. For example, suppose there is a flow value (s) in a data set that is high compared to the rest of the flow values. Using the statistical definition of an outlier, (s) is considered an outlier. However, it might be that on that day, it was very cold. Since consumption of natural gas is highly affected by temperature, we expect a cold day to have higher flow than days with higher temperatures. Therefore in this thesis, we assume that flow close to historical patterns followed by the majority of the data are the correct data. Those data points that lie sufficiently far from their immediate neighbors are outliers after considering all factors affecting consumption of natural gas.

The outliers in flow are mostly caused by errors in data file processing or because of faulty meter measurements. Some of the outliers observed in natural gas flow received at the GasDay project include (see Figures 1.6 and 1.7):

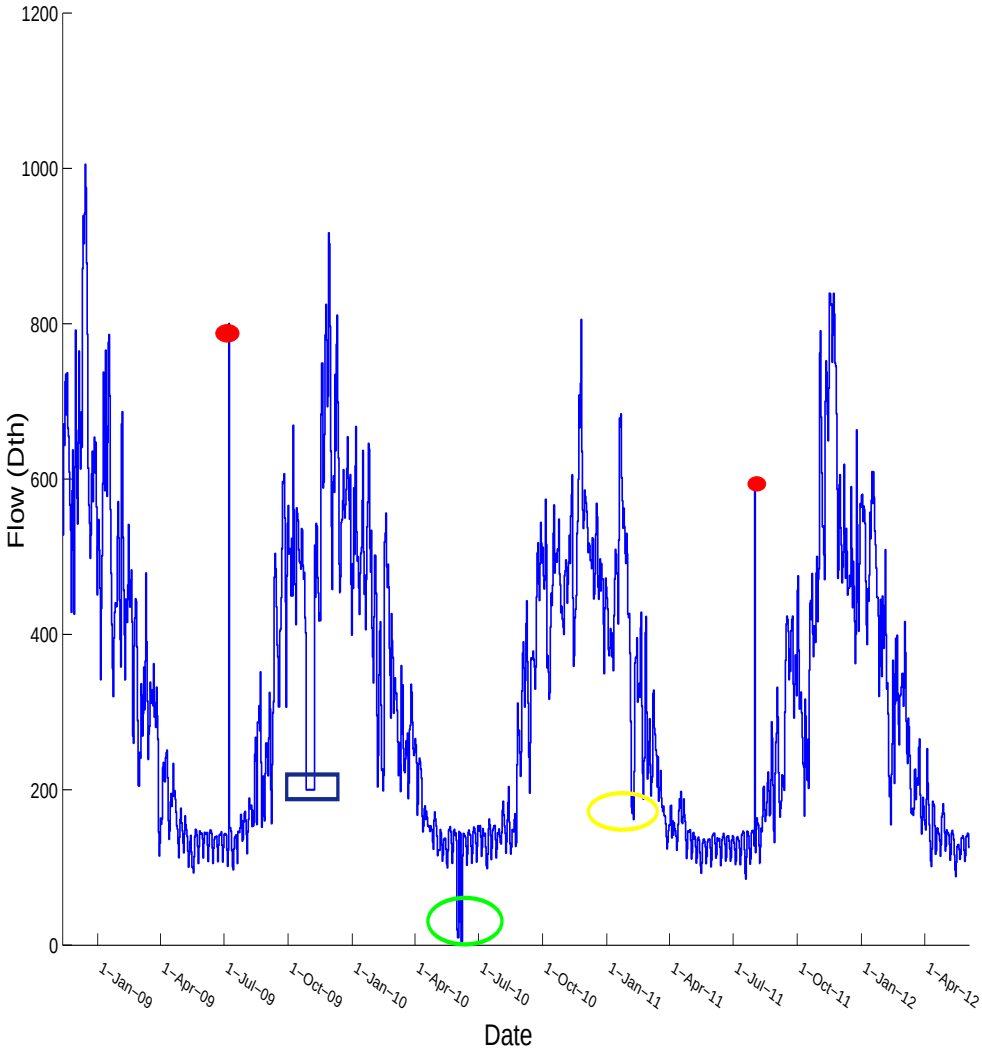

Figure 1.6: Time series flow outliers as observed by the GasDay project

- Single abnormal flow measurement, points circled in red;
- Multiple abnormal flow measurements even up to a month, points circled in

yellow;

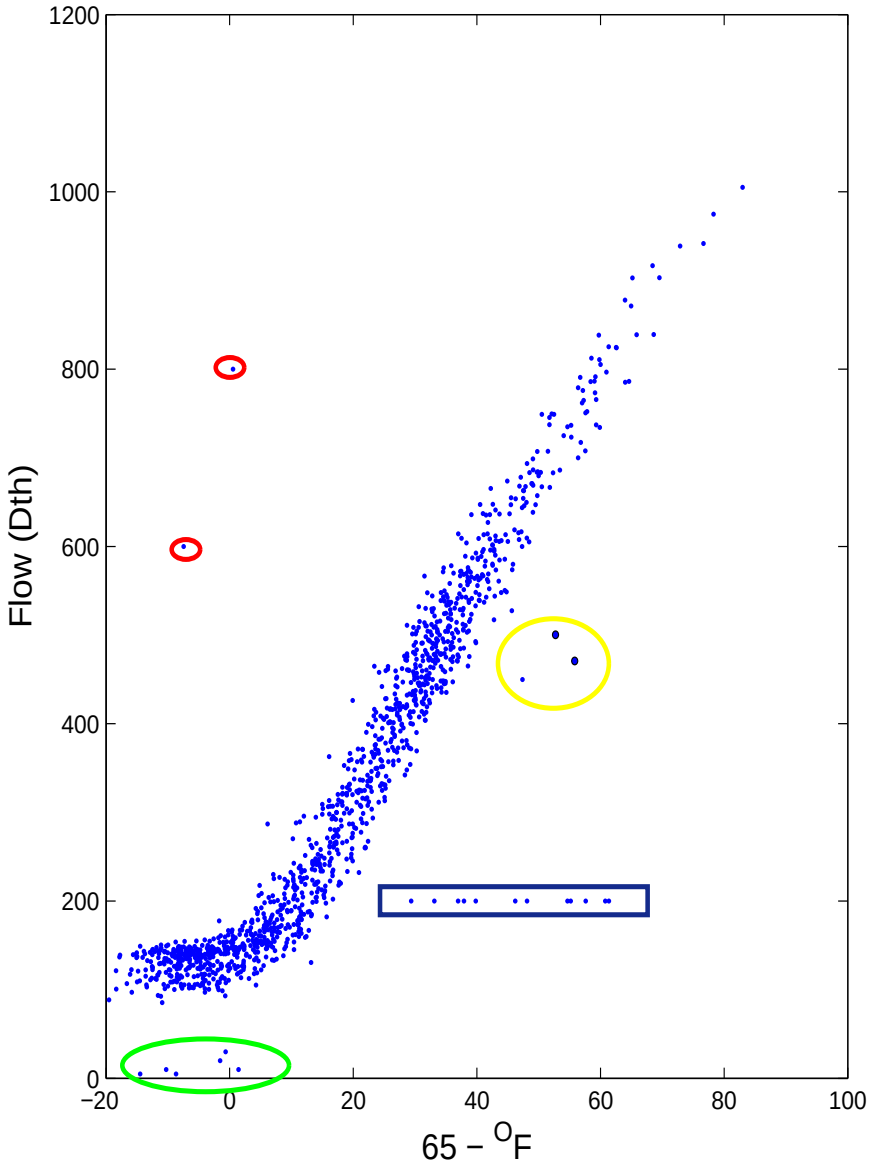

Figure 1.7: Flow outliers as observed by the GasDay project

- Same abnormal repeated flow values, points inside the green rectangle;
- Flow value at zero, points circled in green; and

- Abnormal flow values as a result of events (hurricane, storms).

We have presented temperature and flow time series. The presentation has also shown flow consumption in a JOTO is highly affected by temperature. The next section presents mathematical models with an explanation of other factors that affect the consumption of natural gas flow.

1.3 GasDay's Mathematical Models

This section provides a brief discussion of the mathematical models used by the GasDay project to predict the consumption of natural gas flow. There are two types of models. One type deals with numeric and nonnumeric data types known as logical models. The other type, mathematical models, only deals with numeric data types (6). Mathematical models are described using mathematical operators to relate inputs and desired outputs by mathematical equations (7). Mathematical model types include fixed models, parametric models, and nonparametric models. Parameters are unknown quantities that characterize a model. The parametric model explicitly uses mathematical equations to characterize the structure of the relationship between inputs and outputs. The GasDay projects uses parametric mathematical models to predict the consumption of natural gas (2). Some of the inputs used include Heating Degree Days with a reference temperature of 65 (HDD65), HDD with a wind correction (HDDW65), Cooling Degree Days (CDD65), and the base load (β_o). The indices for

the cosine and the sine of the day of the week (DOW) and the day of the year (DOY) also are used (6). For example, Equation (1.1) shows the relationship between the model parameters for the multiple linear regression modeling technique as used by the GasDay project. Let each β_j be a parameter that specifies how the output is related to the k^{th} input, and let $x_{k,j}$ represent the k^{th} input factor on day k. Then estimated flow

$$\widehat{s}_k = \beta_o + \sum \beta_k x_{kj}. \tag{1.1}$$

Equation (1.2) is said to be simple, linear in the parameters (β_o), and linear in the predictor variable (X_k), with an error term ε_k;

$$s_k = \beta_o + \beta_1 X_k + \varepsilon_k. \tag{1.2}$$

The error term represents the residuals from the models as explained in the next subsection.

1.3.1 Model Residuals

One approach to outlier detection is to fit a model of the desired form to the data and then examine the residuals, looking for points that are poorly predicted by the model (6; 8). We use the same terms used by GasDay's models to define the residual, defined as the difference between the flows estimated by the GasDay models

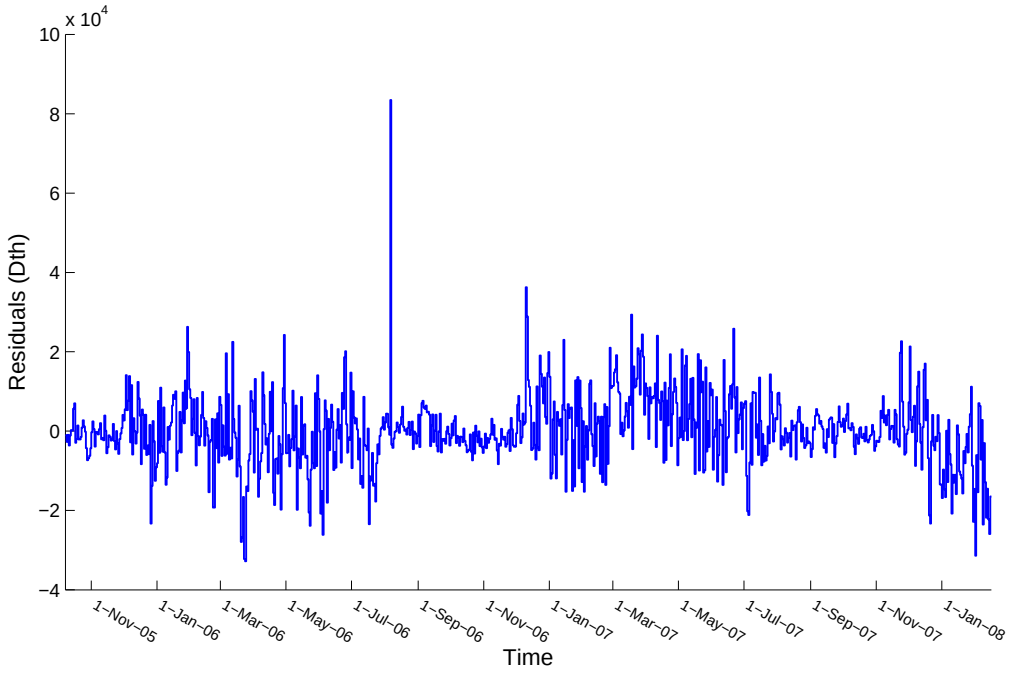

Figure 1.8: Residuals from the models for a JOTO

and measured flows. Let $\hat{s}_k$ be the flow estimated by the GasDay model and s_k be the measured flow for k^{th} day. The residual (or error) is

$$r_k = \hat{s}_k - s_k.$$

Figure 1.8 shows time series of residuals for a JOTO from the models that estimates flow using inputs including HDD65, HDDW65, CDD65, β_o, Cosine and Sine of DOW, Cosine and Sine of DOY, and Holidays. In the next section, we argue that the residuals from the GasDay models are not normally distributed.

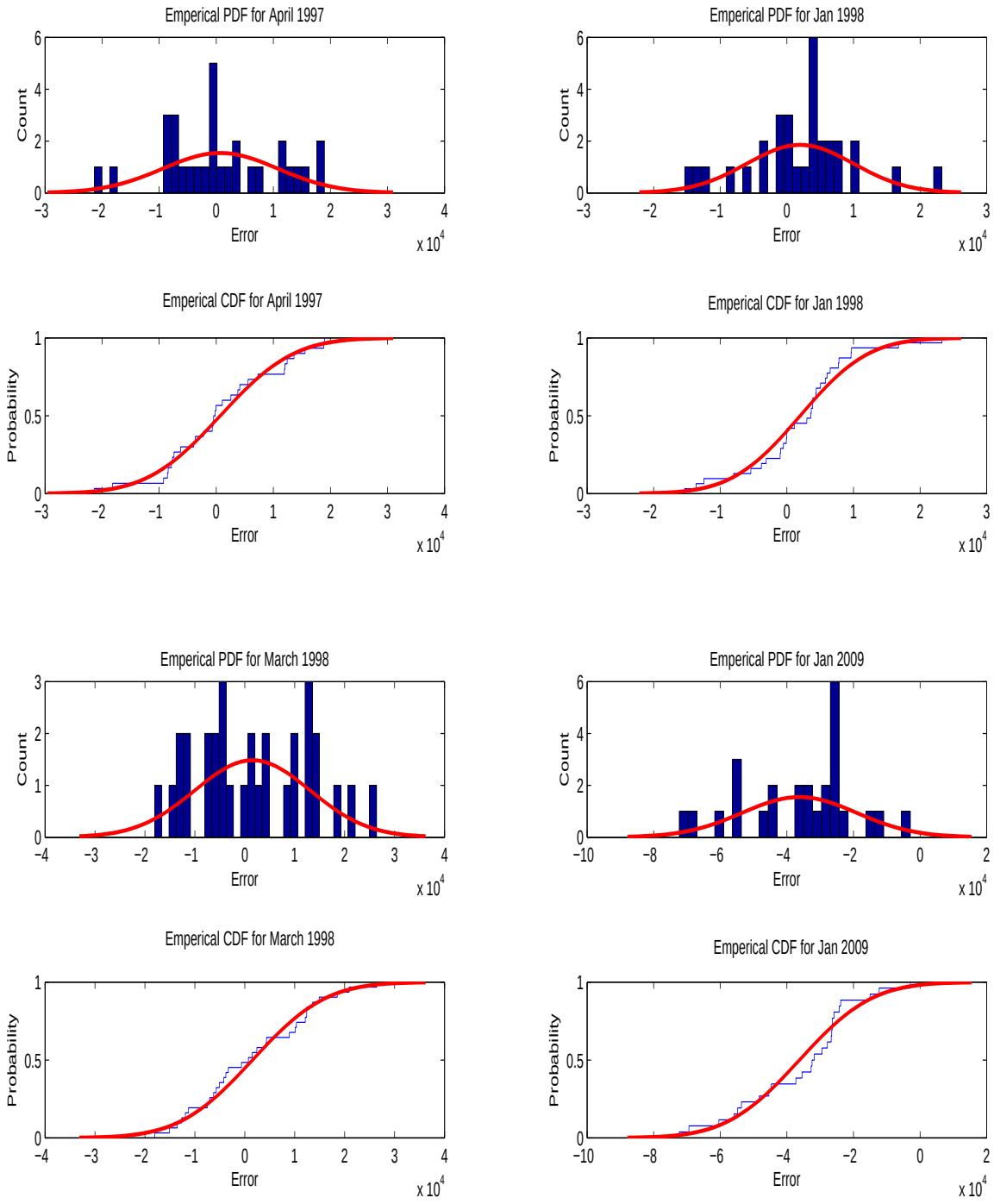

Figure 1.9: Histograms showing distribution of residuals for four JOTO compared with a normal distribution.

1.3.2 Statistical Test

This subsection explains statistical tests applied to the residuals from GasDay models.

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable (9). The skewness of a set of values measures the degree to which the values are symmetrically distributed around the mean. We know the i^{th} moment about the mean (or i^{th} central moment) of a real-valued random variable X is the quantity $\mu_i := E[(X - E[X])^i]$, where E is the expectation operator (9). If μ_i is the third moment about the mean μ, and σ is the standard deviation, skewness of a distribution is (10)

$$\gamma_1 = \frac{\mu_3}{\sigma^3}. \tag{1.3}$$

Kurtosis is a measure of the peakedness of the probability distribution of a real-valued random variable (11). Kurosis is a normalized form of the fourth central moment μ_4 of a distribution (10),

$$g_2 = \frac{\mu_4}{{\mu_2}^2}. \tag{1.4}$$

The Jarque-Bera (JB) statistical test is used to test if a given set of samples come from a normal distribution (12). Bera et al. (13) define the Jarque-Bera test as a measure of departure from normality, based on the sample kurtosis and skewness. If we

let n be the number of observations, γ_1 be the sample skewness, and g_2 be the sample kurtosis (13),

$$JB = \frac{n}{6}\left(\gamma^2 + \frac{(g_2 - 3)^2}{4}\right). \tag{1.5}$$

MATLAB has implemented the Jarque-Bera test in a function called "jbtest" whose null hypothesis is that the sample X comes from a normal distribution. The test returns the value of 1 if it rejects the null hypothesis at the 5% significance level and the value of 0 if it cannot (14).

Chapter 2 explains the existing GasDay technique for detecting outliers. It uses techniques that are motivated by normally distributed data sets. However, the Jarque-Bera test shows that the residuals from the GasDay models are not normally distributed. The MATLAB function "jbtest" returns a value of 0 for the model residuals of all the operating areas shown in Figure 1.9.

Histograms plots also can be used to illustrate the distribution of a data set. For example, Figure 1.9 illustrates that residuals for operating areas often are not normally distributed. The red lines are normal distributions. The empirical distributions do not fit the red lines.

The *JB* statistical test and visualization from histograms conclude that the residuals of the GasDay flow models are not normally distributed. One motivation of

this work is to find techniques to detect outliers in time series that are not motivated by normally distribution samples. The next section gives the statement of the problem considered by this research in both mathematical and business forms.

1.4 Statement of the Problem

This work addresses a problem which can be stated in a business or in a mathematical form:

- **Business statement:** Develop techniques that can be implemented in GasDay to detect outliers in both historical and real-time data. The focus is natural gas flow for temperature-sensitive operating areas.
- **Mathematical statement:** Let $x_1, ..., x_n$ be the points of a time series data set X, and let K be a set of points ($K \subset X$) that follows the historical pattern. We define an outlier as a point p in X not belonging to K. Develop a technique to find p from X.

1.5 Introduction to Performance Evaluation

An outlier detection technique presented in this thesis is evaluated against GasDay's existing technique. For this evaluation approach to work, we need data sets for which outliers are known. In practice, we never know for sure because of faults in

flow measurements. Also, when asked, sometimes operating area personnel cannot say for sure which flow values are true outliers.

Two strategies are used to generate the evaluation data sets. First, we use real data with outliers identified by experts. Second, we use empirical distributions of observed outliers to make synthetic outliers. More details on the use of these strategies are provided in Chapter 3. We construct evaluation data sets that contain a combination of a single outlier, multiple outliers, repeated outliers, and flow values at or near zero.

The following metrics are used in the fields of science, engineering, industry, and statistics to evaluate the performance of a classification technique (3):

True Positive (TP) - an outlier is classified correctly as an outlier.

False Positive (FP) - correct value is classified as an outlier.

True Negative (TN) - correct value is classified as a correct value.

False Negative (FN) - an outlier is wrongly classified as a correct data.

Using metrics TP, FP, TN, and FN, we define four performance metrics (3; 15):

- Accuracy is the degree of closeness of measurements of a quantity to its actual value.

$$A = \frac{TP + TN}{TP + TN + FN}. \tag{1.6}$$

- Precision is a measure of exactness.

$$P = \frac{TP}{TP + FP}. \tag{1.7}$$

- Recall is a measure of completeness.

$$R = \frac{TP}{TP + FN}. \tag{1.8}$$

- F1 measures the balance between precision and recall; it is a harmonic mean between them.

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}. \tag{1.9}$$

Using both the proposed and the existing GasDay techniques, Accuracy, Precision, Recall, and F1 measures are computed for each of the evaluation data sets. The performance of each technique is presented in Chapter 4.

1.6 Organization of Thesis and Summary

Chapter 2 surveys the literature discussing various time series outlier detection techniques. Chapter 3 presents Density Based Spatial Clustering of Applications with Noise (DBSCAN) applied specifically to natural gas flow. In Chapter 4, results are presented to show the performance of GasDay's existing technique and our DBSCAN technique. Chapter 5 serves as the conclusion to the thesis and describes future work

involving time series outlier detection techniques.

CHAPTER 2

TIME SERIES OUTLIER DETECTION TECHNIQUES LITERATURE SURVEY

This chapter provides a summary of the literature discussing various outlier detection techniques. It covers statistical and clustering-based outlier detection techniques. It also outlines different Density Based Spatial Clustering of Applications with Noise (DBSCAN) applications. The chapter starts with the discussion of time series outliers.

2.1 Time Series Outliers

Figure 2.1 shows daily natural flow consumption in Decatherms (Dth) over a period of one year for one temperature-sensitive operating area (JOTO) with 369 daily flow points. This data set is part of an operating area selected randomly among other operating areas. Flow starts on Jan 6, 2008, and ends on Jan 09, 2009. All data values fall between 80,000 Dth and 580,000 Dth. There are isolated flow points on Feb 15th, 2008, and Aug 15th, 2008, (points marked in red) that lie sufficiently far from their immediate neighbors to qualify as time series outliers. Looking at the total range of

data variation, these points are not extreme relative to the range of variation, but they are extreme relative to the variation observed by immediate neighbors (*locally*). Specifically, time series outliers are data points that do not follow the general (historical) pattern of regular variation seen in the data sequence (3; 4; 5).

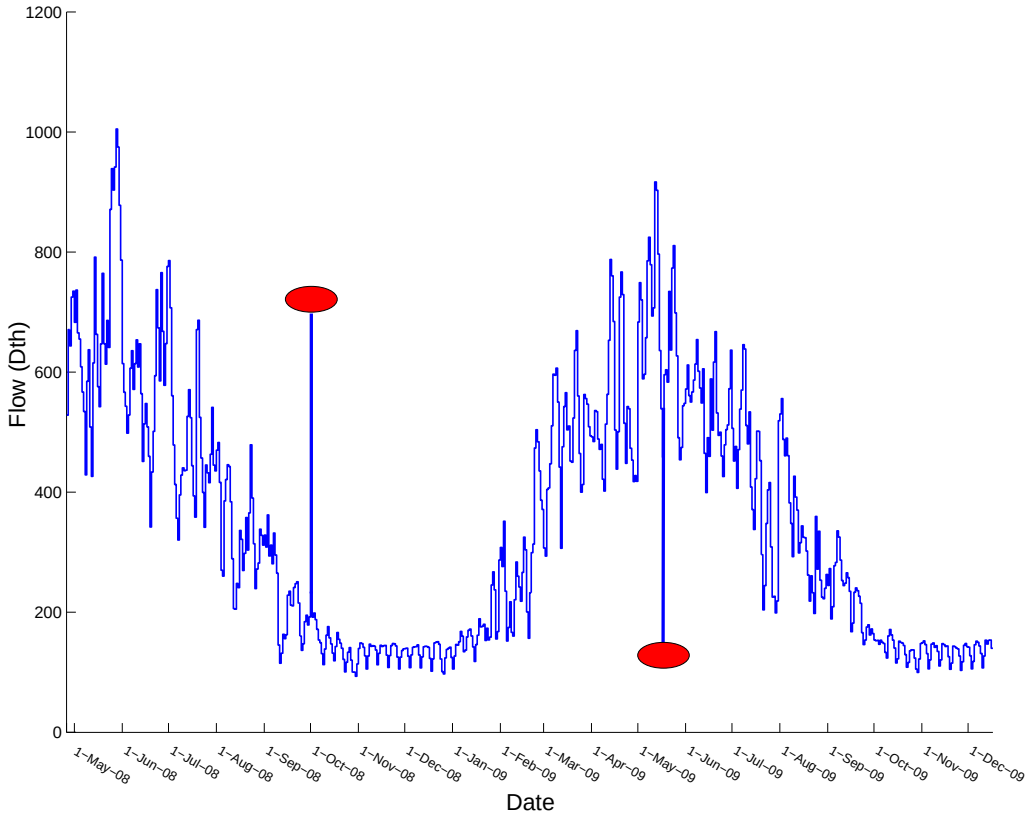

Figure 2.1: Daily flow illustrating the phenomenon of time series outliers

GasDay wants to detect errors in their data sets. A true flow value might be 1000 Dth, and the value reported can be 1001 Dth, which is erroneous. We have no hope of detecting that. We settle for detecting outliers. Most outliers observed by GasDay result from human error during manual entry and manual intervention or from file processing error and equipment data recording errors. One consequence of outliers

in a data set is a cost incurred by not detecting the outliers.

Data mining techniques are used to remove or replace outliers from the data set to make it clean. Clean, correct data is required to train high quality models. The presence of outliers affects the training process resulting in poor models (45). Therefore, it is important that these outliers are detected and removed or replaced with modeled values from the data sets. The next section presents GasDay's existing technique for detecting outliers using approaches motivated by normally distributed samples.

2.2 Detecting Outliers Using Approaches Motivated by Normally Distributed Samples

The GasDay project uses approaches that are motivated by normally distributed samples to detect outliers in residuals from models. The Gaussian (normal) distribution frequently is used in statistics and analysis. We use it to describe a simple approach to statistical outlier detection used by the GasDay project. The normal distribution $\mathrm{N}(\mu, \sigma)$ has two parameters, the mean (μ) and the standard deviation (σ) (3; 38). Figure 2.2 shows the density function of the distribution with ($\mu = 1$) and ($\sigma = 0$). In (3), Tan states "that there is little chance that an object (value) from an $\mathrm{N}(0, 1)$ distribution will occur in the tails of the distribution." Given a constant value c such that prob($|x| \geq c$), Tan defines an outlier as an object with attribute value x from

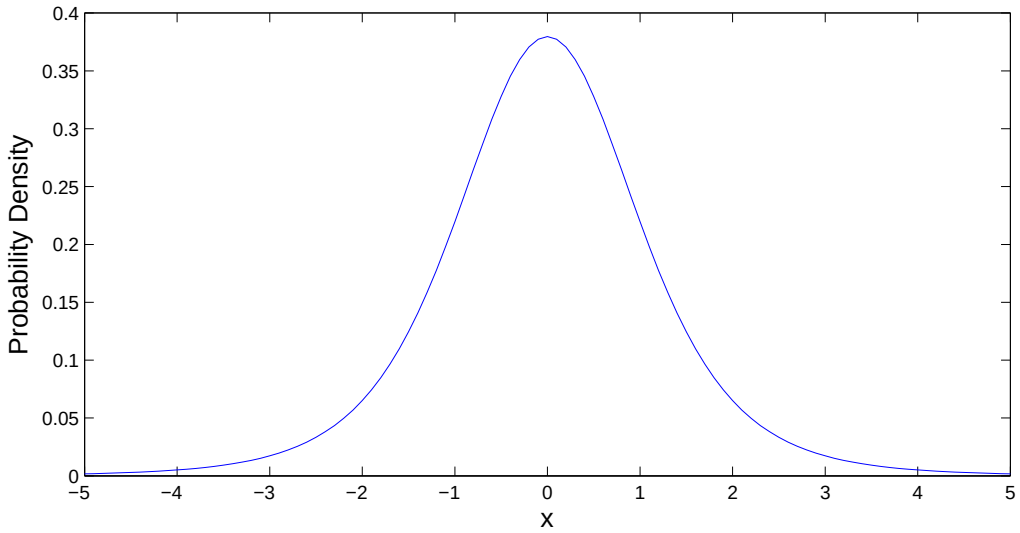

Figure 2.2: Probability density function of a Gaussian distribution N(1,0)

a Gaussian distribution with $\mu = 0$ and $\sigma = 1$ if

$$|x| \geq c. \tag{2.1}$$

In general, prob($|x| \geq c$) decreases rapidly as c increases. An object that lies beyond the central area between ± 3 standard deviations often is considered to be an outlier. The GasDay project uses an absolute measure $\pm 3\sigma$ or $\pm 5\sigma$ thresholds to detect outliers in residuals from the models as an initial step to help an expert. The GasDay expert(s) visualize the points indicated as outliers and decide whether they should be considered as outliers. For example, Figure 2.3 shows flow points marked as outliers (red X) and flow modeled points (red circle) as displayed by GasDay's existing technique. Using visualization, an expert decides which red Xs should be considered as outliers.

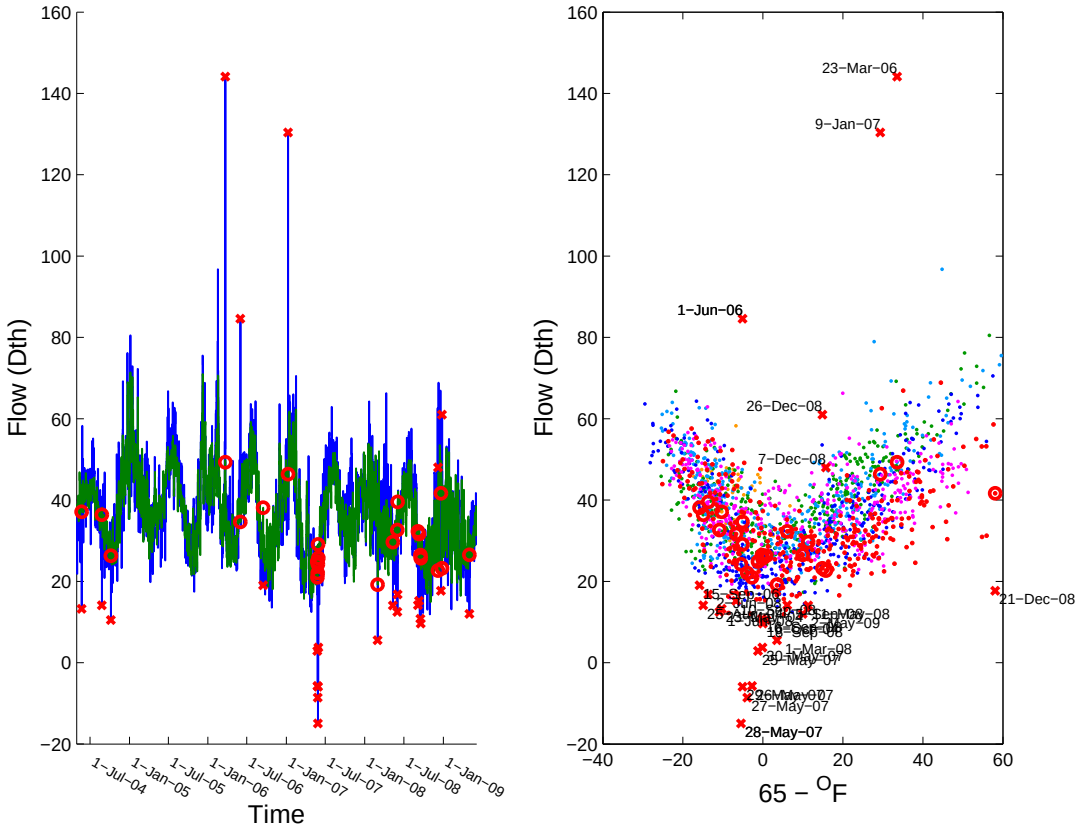

Figure 2.3: Time series and scatter plots display outliers detected by the existing GasDay technique

Although the approaches used are those motivated by normally distributed data sets, as explained in Chapter **??**, residuals from the models are not normally distributed. That is why the outliers detected might not really be outliers. Also, most operating areas do not have negative flows. For example, if the actual flow value for a given day is 500 Dth, the lowest the model can predict is 0, making an error of -500 Dth, while positive error made can be unbounded. The error made in the negative direction is not the same as the error made in the positive direction. Hence, the use of $\pm 3\sigma$ or $\pm 5\sigma$ for both tails in the distribution of residuals also leads to false positive and false negative

classifications. The identified and approved outliers are replaced with values estimated by the model. Also, a relative measure approach is used where flow value less than half the modeled value or more than twice the modeled value is flagged as an outlier. Even if GasDay uses statistical approaches motivated by normally distributed samples to detect outliers, it does not depend entirely on results obtained by those approaches. The GasDay expert(s) are required to approve the results.

There several robust statistical methods used to detect outliers as explained by Pearson in (40). The Hampel identifier is regarded as one of the most robust outlier identifiers (40). By replacing the mean (μ) with the median and the standard deviation (σ) with the Median Absolute Deviation (MAD), the Hampel identifier is obtained (6; 40). GasDay lab uses a variant of Hampel outlier detection in one of its customer-specific services (6).

The next section provides an overview of non-statistical clustering techniques used to detect outliers. One of these techniques is used by this work as a different and more effective approach that can be used by the GasDay project to detect outliers.

2.3 Clustering Algorithms

In this section, we provide a brief overview of clustering algorithms. Cluster analysis is the process of assigning a set of observations into clusters so that observations in the same cluster have similar features (43). Clustering is the task of

grouping similar points together with respect to distance or, equivalently, a similarity measure (25). Most clustering algorithms are based on one of the following:

- Hierarchical techniques organize data in a nested sequence of groups displayed in a form of a tree structure called a dendrogram. It is divided into two types; agglomerative, in which one starts at the leaves and successively merges clusters together; and divisive, in which one starts at the root and recursively splits the clusters (31; 32).

- Grid-based techniques quantize the object space into a finite number of cells that form a grid structure. Each object falls into a grid cell whose corresponding attribute intervals contain the values of the object (3; 36).

- Model-based techniques assume that the data were generated by a model and tries to recover the original model from the data. The model recovered from the data then defines clusters and an assignment of data to clusters (3).

- Graph-based techniques represent data objects using nodes. The proximity between two objects is represented by the weight of the edge between the corresponding nodes (3).

- Density-based algorithms typically regard clusters as dense regions of objects in the data space that are separated by regions of low density. They find and separate regions of high density from low-density regions. Density-based algorithms make it easy to discover arbitrary clusters (25; 35).

2.3.1 Clustering-Based Techniques for Outlier Detection

Clustering finds groups of strongly related objects. Outlier detection finds objects that are not strongly related to other objects. Thus, an object is a cluster-based outlier if the object does not belong strongly to any cluster (3). In detecting outliers, small clusters that are far from other clusters are considered to contain outliers. This approach is sensitive to the number of clusters selected. It requires thresholds for the minimum cluster size and the distance between a small cluster (with outliers) and other clusters. If a cluster is smaller than the minimum size, it is regarded as a cluster of outliers.

This thesis presents a density-based clustering technique known as Density Based Spatial Clustering of Applications with Noise (DBSCAN). The detection of time series outliers using the DBSCAN technique is discussed in detail in the next section.

2.4 Density Based Spatial Clustering of Applications with Noise (DBSCAN)

In this section, we present a density-based clustering technique which estimates similarities between points from a data set with respect to distance and partitions them into subsets known as clusters, so that the points in each cluster share some common trait (42). We use the clustering technique used in data mining known as Density Based Spatial Clustering of Applications with Noise (DBSCAN).

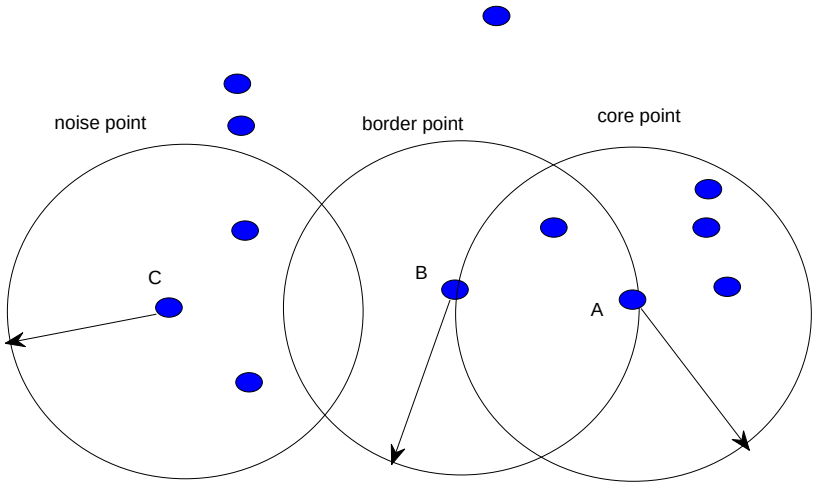

Figure 2.4: Illustrates DBSCAN's key concepts: core (A), border (B), and noise (C) points

DBSCAN is designed to discover the clusters and the noise from a given set of points by classifying a point (i) inside of a cluster (core point), (ii) in the edge of a cluster (border point), or (iii) as neither a core point nor a border point (noise) (3). DBSCAN requires two important parameters; *Eps*, which is a specified radius around a point to other points, and $MinPts$, which is the minimum number of points required to form a cluster. Further discussion how these parameters are selected is given in Section 2.3.2. In the next subsection, we provide the key concepts to help understand the DBSCAN algorithm.

2.4.1 Key Concepts

The following definitions are the key concepts in understanding the DBSCAN algorithm::

Definition 1 (16) The $Eps - neighborhood$ of a point p, denoted by N_{Eps}(p), is defined by N_{Eps}(p) = {q ∈P | dist(p,q) <= Eps }.

A point is a *core point* if it has more than a specified minimum number of points required to form a cluster ($MinPts$) within an $Eps - neighborhood$. These are points that are in the interior of a cluster. A *border point* has fewer than $MinPts$ within an Eps, but it is in the $Eps - neighborhood$ of a core point. A *noise point* is any point that is neither a core point nor a border point. For example, given $MinPts = 4$ and $Eps = 1$ as illustrated in Figure 2.4 (3), A is a core point, B is a border point, and C is a noise point (16).

Definition 2: (3) The *density-based approach* is an approach that regards clusters as regions in the data space in which the objects are dense and separated by regions of low object density (outliers).

Definition 3: (16) Considering points p_1 and p_2 from Figure 2.5, p_1 is *directly density reachable* from p_2 if

1. Points are close enough to each other such that dist(p_1,p_2) < Eps, as measured using Euclidean distance or using any other distance measure.

2. There are at least $MinPts$ points in its neighborhood. For example, if $MinPts = 6$, then p_1 must have at least 6 points as its neighbors.

This concept of direct density-reachability is shown by Figure 2.5. In this case, the figure shows that p_1 is density reachable from p_2 because the dist(p_1,p_2) < Eps, and p_1 has enough points as its neighbors.

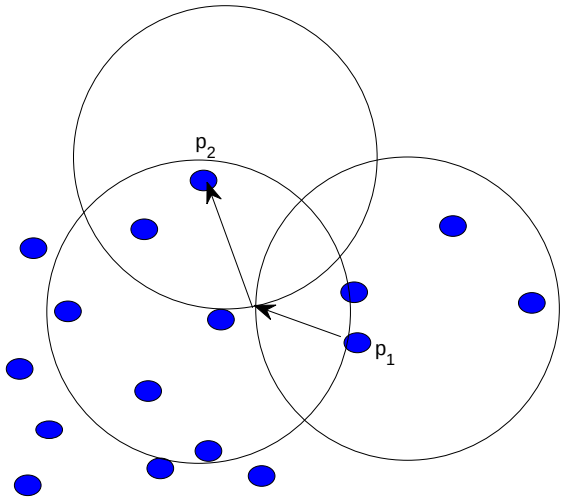

Figure 2.5: Point p_1 is density reachable from p_2

Definition 4: (16) A point p_1 is *density reachable* from a point p_2 wrt. Eps and $MinPts$ if there is a chain of points p_1, ..., p_n, such that p_{i+1} is directly density-reachable from p_i.

Definition 5: (16) A point p_0 is *density-connected* to a point p_n wrt. Eps and $MinPts$ if there is a point q such that both p_0 and p_n are density-reachable from q *wrt.* Eps and $MinPts$.

The scatter plot of consumption vs. HDD in Figure 2.6 illustrates the density-connectivity concept. A cluster is a set of all density-connected points. In the next section, we present the DBSCAN algorithm.

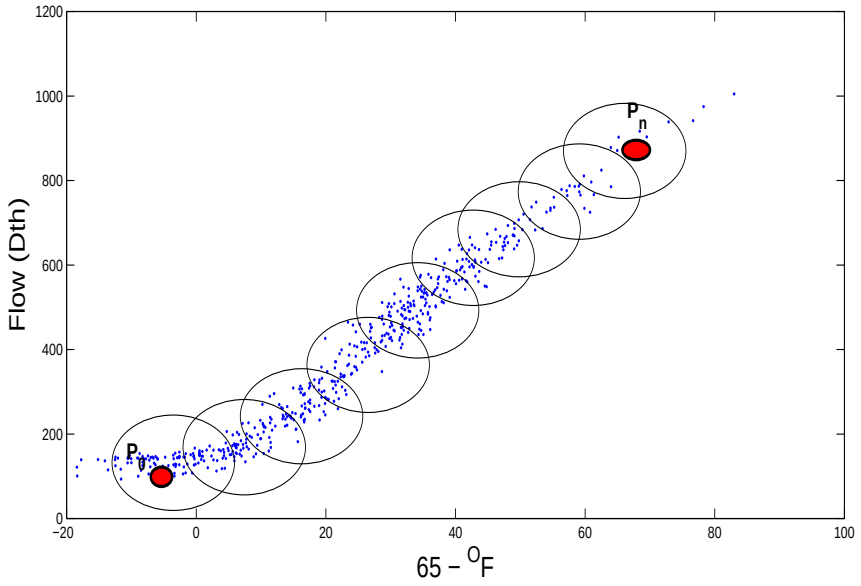

Figure 2.6: A point p_0 is density-connected to a point p_n

2.4.2 The Algorithm

In general, using parameters Eps and $MinPts$, DBSCAN finds a cluster by starting with an arbitrary point p from a set of points and retrieves all points density-reachable from p wrt. Eps and $MinPts$. Suppose p is a core point, if p has neighboring points greater than or equal to the value of $MinPts$, a cluster is started. Otherwise, the point is labeled as an outlier, and a new unvisited point is retrieved and

processed leading to the discovery of a further cluster of core points (3). A point can be in a cluster, and it can be an outlier. After all the points have been visited, any points not belonging to any clusters are considered outliers. The formal details are given in Table 2.1, and Algorithm 1 presents MATLAB-like pseudocode for the DBSCAN algorithm (16).

Table 2.1: DBSCAN Algorithm

0. Select the values of Eps and $MinPts$ for a data set P to be clustered.
1. Start with an arbitrary point p and retrieve all points density-reachable.
2. If p is a core point that contains at most $MinPts$ points
2.1 A cluster is formed,
2.2 Otherwise, label p as an outlier.
3. A new unvisited point is retrieved and processed leading to the discovery of further clusters of core points.
4. Repeat step 3 until all the points have been visited.
5. Label any points not belonging to any cluster as outliers.

Algorithm 1

```
%Aim:
    Clustering the data with Density-Based Scan Algorithm with Noise (DBSCAN)
%Input:
    SetOfPoints (P) - data set (m,n); m-objects, n-variables;
    Eps             - neighborhood radius
    MinPts          - minimal number of objects required to form a cluster
-------------------------------------------------------------------------
%Output:
   A vector specifying assignment of a point to certain cluster.
   E.g 1st and 3rd points can be in cluster
   #1 and 2nd and 4th points can be in cluster #2, etc
-------------------------------------------------------------------------
function [IsPointAnOutlier] = DBSCAN(SetOfPoints, MinPts, Eps)
    SetOfPoints = Normalize(SetOfPoints)
    Clusterid = 0
     FOR each unvisited point p in a SetOfPoints
        mark p as visited
        PListOfNeigbors = getNeighbors(SetOfPoints, P, Eps)
         IF sizeof(PListOfNeigbors) < MinPts
            mark p as OUTLIER
         ELSE
            Clusterid = next cluster
            expandCluster(SetOfPoints, P, N, Clusterid, Eps, MinPts)
         ENDIF
     ENDFOR
ENDDBSCAN
function expandCluster(SetOfPoints, P, N, Clusterid, Eps, MinPts)
    add p to cluster Clusterid
    WHILE there is unvisited point p' in ListOfNeighbors
        mark p' as visited
        PListOfNeigbors' = getNeighbors(p', Eps)
        IF  PListOfNeigbors' <= MinPts
             PListOfNeigbors  = PListOfNeigbors joined with PListOfNeigbors
            add p' to cluster Clusterid
         ENDIF
    ENDWHILE
    RETURN
ENDEXPANDCLUSTER
function = getNeighbors (SetOfPoints, P, Eps)
    RETURN Eps-Neighborhood of p in SetOfPoints as a list of points
ENDGETNEIGHBORS
```

DBSCAN has several advantages including its ability to find arbitrarily shaped clusters. It does not require the user to know the number of clusters in the data in advance. DBSCAN is very robust to outliers and requires just two parameters, Eps and $MinPts$ (3; 16). However, DBSCAN is highly affected by the distance measure used in finding the distance between two points. Its effectiveness in clustering data points depends on the distance measure used. The Euclidean distance measure is commonly used, but any other distance measure can be used. Also, before computing the distances between two points with different units, the data points must be normalized (42).

2.4.3 Selecting the Parameters Eps and $MinPts$

The DBSCAN algorithm requires two user-defined parameters Eps and $MinPts$. The values of these parameters have a big impact on the performance of the DBSCAN (16; 25). For instance, if Eps is large enough, then all points form a single cluster, and no points are labeled as outliers. Likewise, if Eps is too small, majority of the points are labeled as outliers. There are several approaches that can be used to determine the values of Eps and $MinPts$.

The first approach uses the parameters specified by the experts. The parameters are provided by an expert who is very familiar with the data set to be clustered. An expert can provide the parameters and run the DBSCAN algorithm, which provides graphs showing which points from the data sets are considered to be

outliers. Using visualization, an expert looks at the graphs, adjusts the parameters, and runs the algorithm until he/she gets good results. Good results are determined by the expert knowledge of the data set. An expert selects parameters that can be used as default parameters for that data set.

The $k - dist$ approach looks at the behavior of the distance from a point to its k^{th} nearest neighbor. If k is not larger than the cluster size, the value of $k - dist$ is small for points that belong to the same cluster. The $k - dist$ for points not in the cluster is relatively large. The idea is to pick a value of k to be the $MinPts$. The following steps are performed to find the value of k:

- Compute the $k - dist$, (distance to its k^{th} nearest neighbor) for each of the data points.
- Sort $k - dist$ measures in increasing order.
- Plot the sorted $k - dist$ values. We expect to see a sharp change at the value of $k - dist$ that corresponds to a suitable value of Eps.

For example, Figure 2.7 shows sorted distances of the fourth nearest neighbor ($k = 4$) from the same operating area discussed in Section 2.1 with 369 two-dimensional normalized points. In this example, $MinPts$ is 4, and Eps is approximately 2 (the value corresponding to the knee of the curve) (16).

DBSCAN algorithm has not been used previously to detect outliers in natural

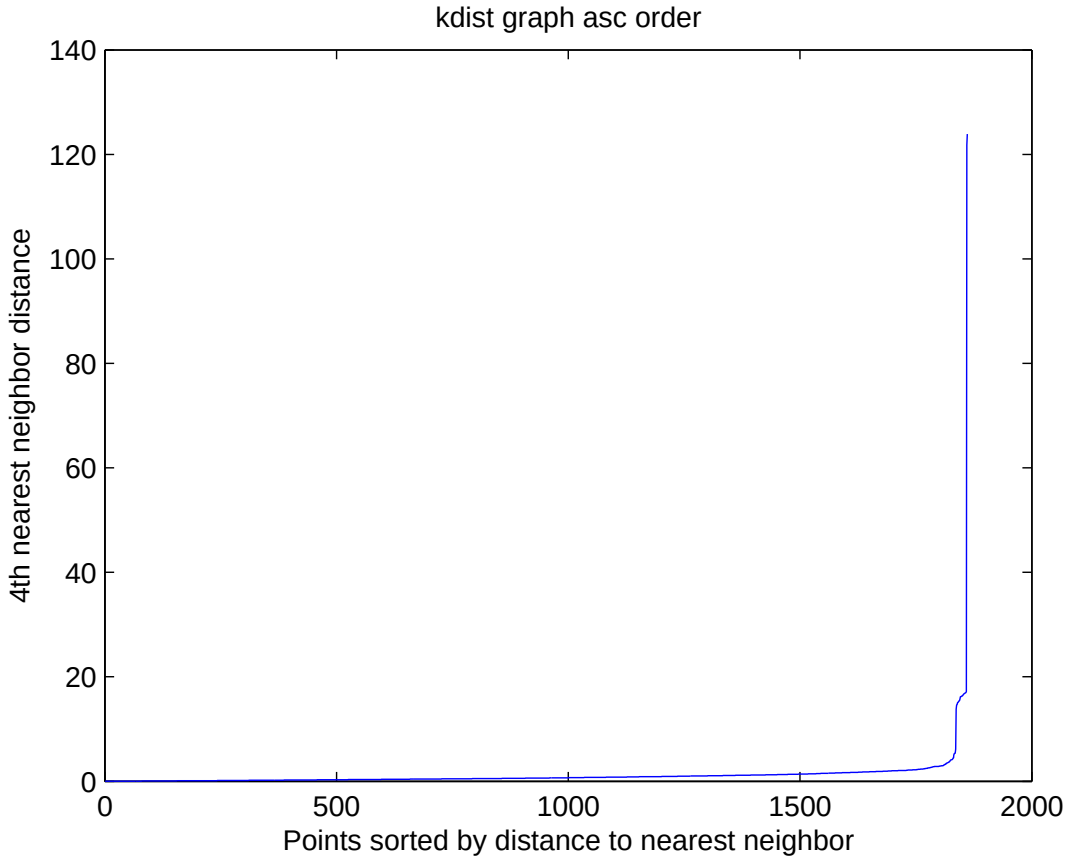

Figure 2.7: $k - dist$ plot for a JOTO with 369 two dimensional points

gas flow. Searching various academic databases do not yield any papers related to the use of DBSCAN in detecting outliers in natural gas flow. The last section in this chapter outlines some of the DBSCAN applications discussed in the literature.

2.5 DBSCAN applications

This section lists several DBSCAN applications discussed in the literature:

Internet traffic classification using DBSCAN (18). The authors apply DBSCAN

algorithm as a machine learning technique for Internet traffic classification. A technique which overcomes some short-comings of traditional classification technique which involves the security and privacy. Authors lists three merits of the DBSCAN algorithm: (1) minimal requirements of domain knowledge to determine the input parameters; (2) discovery of clusters with arbitrary shapes; (3) good efficiency on large data sets.

Evaluation of Fuzzy ARTMAP using DBSCAN in a VLSI Application (30). The authors present a new model for partitioning a circuit using DBSCAN and a fuzzy ARTMAP neural network. Analysis of the investigational results proved that the fuzzy ARTMAP with a DBSCAN model achieves greater performance than only a fuzzy ARTMAP in recognizing sub-circuits with the lowest amount of interconnections between them.

NET-DBSCAN: Clustering the nodes of a dynamic linear network (33). The authors presents a new DBSCAN method known as NET-DBSCAN, a method for clustering the nodes of a linear network whose edges may be temporarily inaccessible.

Although the applications presented are not related to the detection of flow time series outliers, they provide insights on how the DBSCAN algorithm can be adapted to natural gas flow. For example, the three merits outlined in (18) helped theoretically to believe that DBSCAN can be adapted to detect outliers in flow time series.

Chapter 2 has provided a literature survey for statistical and clustering-based outlier detection techniques. In Chapter 3, we present two strategies used to evaluate

the performance of DBSCAN and GasDay's existing outlier detection techniques. All the classes developed in MATLAB used by this work are presented in this Chapter. More important, we propose a new DBSCAN application by adapting it specifically for natural gas flow time series data.

CHAPTER 3

DENSITY BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE ADAPTED TO NATURAL GAS FLOW

Chapter 2 presented several outlier detection techniques, including the technique known as Density Based Spatial Clustering of Applications with Noise (DBSCAN). In Chapter 3, we present two strategies used to evaluate the performance of DBSCAN and GasDay's existing outlier detection techniques. More importantly we describe how DBSCAN is adapted specifically for natural gas flow time series data. The entire outlier detection process developed in this thesis is presented as well. The chapter starts with the discussion of evaluation of the performance of outlier detection algorithms.

Table 3.1: Definitions for notational used in Chapter 3

Notation	Definition
n	number of days of data
δ_T	inter-arrival times between outliers
x	a value of a uniform random variable
X	a data set with synthetic outliers
Y	a historical data set with identified outliers
Z	a real time data set
σ	standard deviation
MAD	mean absolute deviation
r	residual of JOTO
t	date
P	set of points t and r
d	distance

3.1 Evaluating an Outlier Detection Algorithm

In this Section, we discuss how we will evaluate the performance of DBSCAN and GasDay's existing techniques. For this evaluation approach to work, we need data sets for which outliers are known so we can assess how well each technique finds outliers. To compare the performance of both techniques in detecting outliers, two strategies are used, real and synthetic evaluation data sets. The data used in both sets are daily residuals from the GasDay models of natural gas flow as discussed in Chapter 2.

3.1.1 Real Evaluation Data Sets

Real evaluation data sets are created by experts from the GasDay project. From the raw flow files of different operating areas, there is no way of knowing which flow values are outliers. In practice, we never know for sure. An expert, (Dr. Ron Brown, Director), uses the existing technique discussed in Section 2.2 to specify which flow values are believed to be outliers. These outliers are not removed from the data set; we call them *identified outliers.* They are the outliers detected by the existing technique and approved by the expert. An advantage of using this data set is that we are working with real data. This data set provides more confidence than synthetic data sets because if a technique can detect outliers in this data set, same technique should work the same with any other real data sets. One disadvantage is not knowing for sure that the identified outliers are true outliers. Five operating areas with identified outliers are used to evaluate the performance of both techniques, and the results are presented in Chapter 4.

3.1.2 Synthetic Evaluation Data Sets

A second evaluation strategy uses synthetic evaluation data sets containing synthetic outliers. With these evaluation data sets, we know for sure which values are really outliers because we injected them. The synthetic outliers introduced in these sets have the same empirical distribution as identified outliers from operating areas. If a technique can detect these synthetic outliers, it also detects true outliers from operating areas. We can make as many data sets as we wish. Its disadvantage is that we are not working on exactly the same outliers coming from operating areas. We have developed a class in MATLAB to make 1000 different synthetic evaluation data sets that are used to evaluate the performance of DBSCAN and GasDay's existing techniques. The next subsection discusses the process of developing synthetic evaluation data sets using the same empirical distribution as identified outliers.

3.1.3 Developing a Synthetic Evaluation Data Set

We wish to create synthetic evaluation data sets with the same empirical distribution as real evaluation data sets (described in Section 3.1.1). To show both data sets have the same empirical distribution, we need to show they are similar in some sense. Similarity between the two is shown using selected statistics and graphs as presented in Section 3.1.6. The GasDay experts have two roles in making synthetic outliers. They provide a data set with identified outliers as explained in Section 3.1.1, and they use visualization approaches to approve synthetic evaluation data sets as discussed in Section 3.1.6. In making synthetic outliers, two questions must be addressed;

1. When to insert the next outlier?
2. What is the magnitude of the outlier?

Table 3.2: Inter-arrival times between identified outliers with CDF values

δ_T	1	2	3	5	14	21
CDF	0.27	0.33	0.39	0.40	0.43	0.46

We address each of those questions in turn in the following sub-sections.

3.1.4 When to Insert the Next Outlier?

In developing a synthetic evaluation data set, we need to provide the time intervals between synthetic outliers. We use an example to show how to insert the next outlier and then discuss the general case. As an example, we start with 369 points (same data set introduced in Section 2.1) of clean flow containing 24 identified outliers. We compute the cumulative distribution frequency (CDF) for the inter-arrival times (δ_T) between those outliers to get their distribution. Table 3.2 displays some of the intervals between identified outliers. For example, if $\delta_T = 1$, it means identified outliers arrive successfully to each other. If $\delta_T = 2$, the next outlier arrives 2 days later. Figure 3.1 shows the CDF plot generated. From this plot, we see that almost 60% of

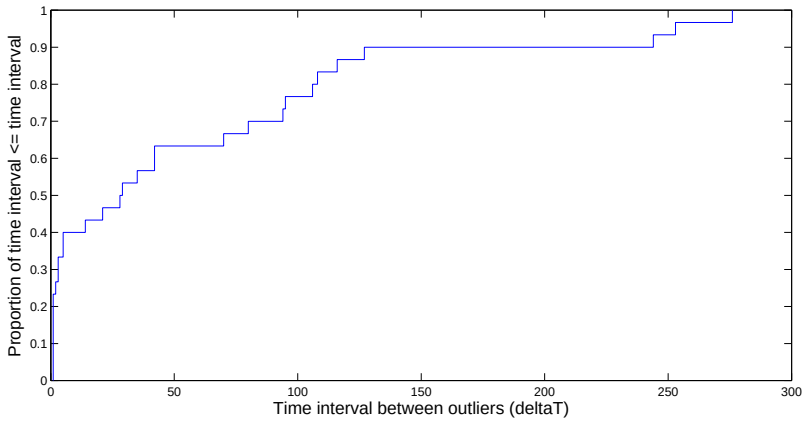

Figure 3.1: Displays (CDF) values for inter-arrival times between identified outliers.

all the points have their inter-arrival times between outliers less than or equal to 50. There are no points with inter-arrival times between 150 and 250 as indicated by a straight line between $\delta_T = 150$ and $\delta_T = 250$. Only 0.1% of the points have time intervals greater than 250. In this example, we have added the δ_T values and CDF values to allow for more time between outliers. The last δ_T value is multiplied by two, and a special equation is used to generate a smoother CDF function as shown by the red line in Figure 3.2. This CDF plot is the one used to find the next time to insert a synthetic outlier.

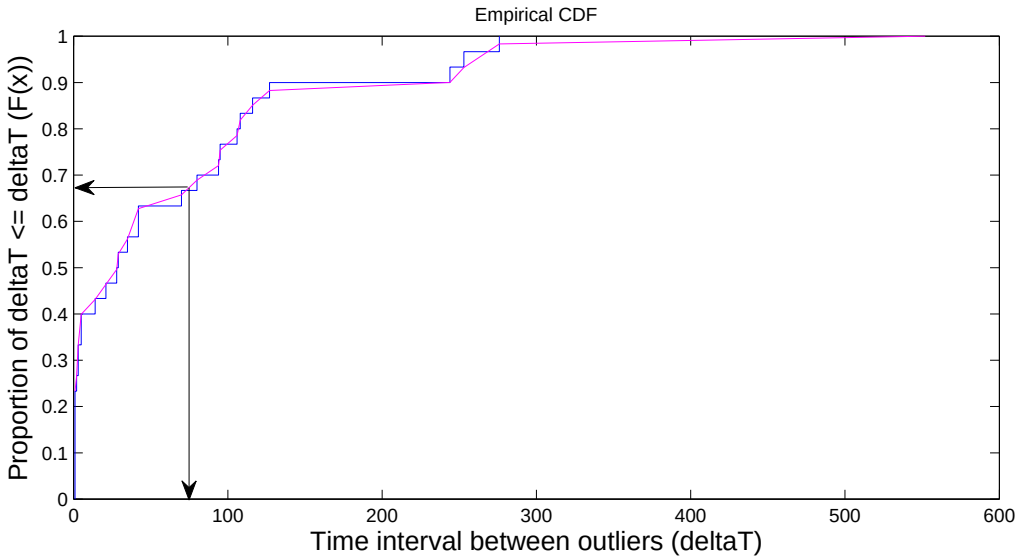

Figure 3.2: Displays a smoother CDF function indicated by a red line.

Suppose, from JOTO with 369 daily residual points with no known outliers (from daily flow operating area discussed in Section 2.1), we want to find when to insert the next outlier. To find the time to the next outlier, we generate a random number from a uniform distribution on [0,1] and compare it with the CDF value in Table 3.2. In this example, the first random number generated was 0.35. From Table 3.2, 0.35 is greater than 0.33 but less than 0.39. So, $\delta_T = 5$ is the time to the next outlier. The first outlier was inserted on March 23rd since data starts on March

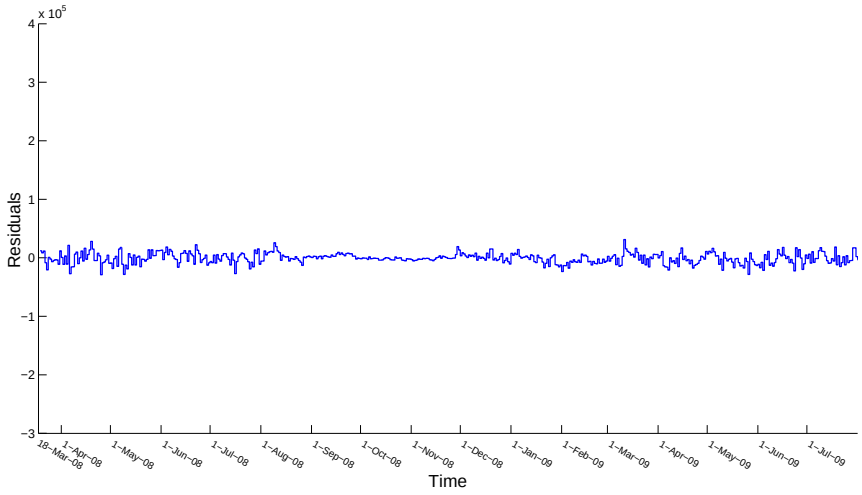

Figure 3.3: Displays a position of the first outlier in a residual time series.

18th. Our new initial point became March 23rd. 0.30 was the next random number generated. So, the new δ_T was 2, and the next outlier was inserted on March 25th. We repeated the generation of random numbers until we got to the last date corresponding to the last point of 1861. At the end, we had inserted 18 outliers.

The algorithm we have outlined for inserting outliers assumes that each outlier is an independent event. However, we know that the times of arrival between outliers in natural gas flow can be dependent on each other. For instance, if a meter is stuck, the same readings of flow values are expected from the meter until it is fixed. In general, we

Assume the time of arrival between outliers is independent of each other.

We can find the next time to insert an outlier as follows:

1. Start with uncleaned flow with identified outliers marked.

2. Generate a Cumulative Distribution Frequency (CDF) plot for inter-arrival times (δ_T) between identified outliers. Generate a smoother CDF function as explained previously using Figure 3.2.

3. Starting at the first day of the data as the starting point, generate a uniform random number x between 0 and 1. Compare the value of x with the CDF values from a smoother CDF function. If the value of x is less than or equal to a CDF value, then its corresponding δ_T value becomes the time to the next synthetic outlier. The start point plus δ_T becomes the new start point.

4. Repeat step 3 until the start point is less than or equal to n.

A sample flow with synthetic outliers and its residuals is presented in Section 3.1.6.

3.1.5 What is the magnitude of the outlier?

The steps in Section 3.1.4 have described when to insert the next outlier. Next, we need to find the magnitude of that outlier, how much the residual should be modified. This implies how much the flow should be modified. The steps to find the magnitude of an outlier are very similar to those used to find the inter-arrival times between outliers. In finding the magnitude, we use identified residual outlier values instead of intervals between the identified outliers.

At this point, we have time series for flow (synthetic data set) with outliers introduced artificially into a cleaned real flow. In the next Section, we argue that both synthetic and identified data sets are similar.

3.1.6 Similarities between Synthetic and Identified Outliers

We want natural gas flow time series with inserted synthetic outliers to be "similar" to actual gas flow time series GasDay receives from customers. That is impossible because we do not know the true outliers in the actual data from customers. We settle for asking that the synthetic flow be similar to operating areas flow with outliers identified by GasDay experts. This subsection discusses two ways that can be

used to show similarity between synthetic outliers and those identified by GasDay experts. We use statistics and graphs to show similarity.

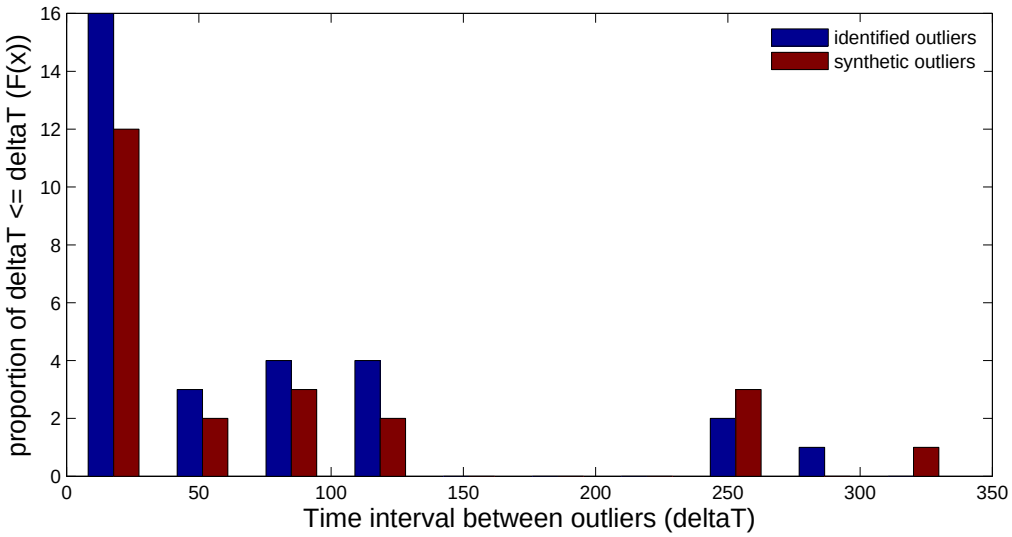

Figure 3.4: Inter-arrival times of identified and synthetic outliers histograms to show their similar distributions.

Let X be a data set with synthetic outliers, Y be a historical data set with identified outliers, and Z the real-time data, daily flow with unknown outliers used by LDCs as inputs to the models. All data sets are for the same operating area. We need to develop a technique to detect outliers from Z. We assume that outliers in Y and Z have the same empirical distribution K because they belong to the same operating area. If we can show that X is similar to Y, then a technique that detects outliers from X can also detect outliers from Y and Z. Therefore, it is important to show that a data set with synthetic outliers is similar to the one with identified outliers.

We show that outliers in X are "similar" to those in Y by presenting the distribution of inter-arrival times between outliers of both X and Y using histograms and cumulative distribution frequencies (CDF). We see that histogram in Figure 3.4

and the CDF plot in Figure 3.5 show the inter-arrival times between identified and synthetic outliers have very similar empirical distributions.

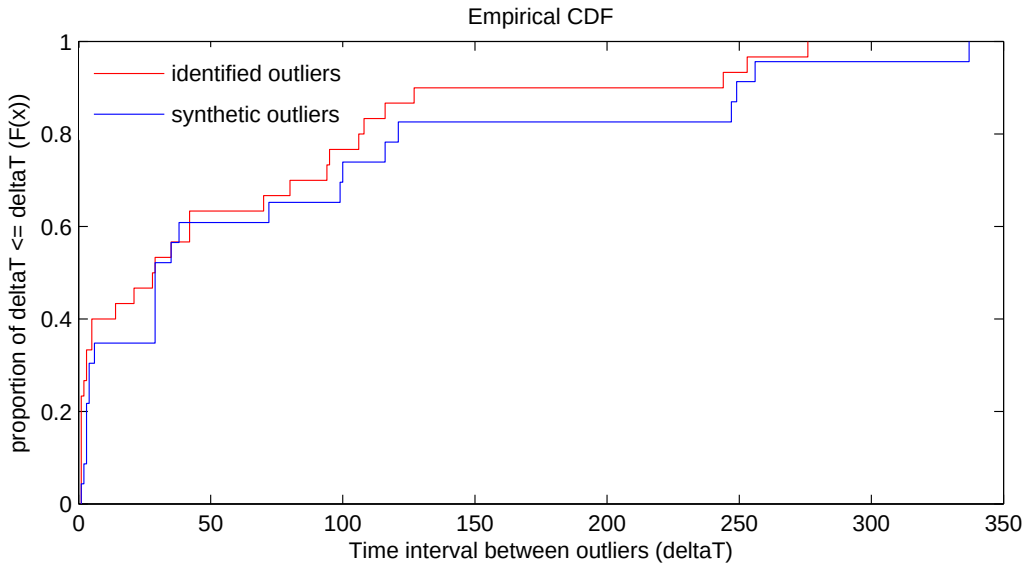

Figure 3.5: Inter-arrival times of identified and synthetic outliers CDFs to help visualize their distributions.

We present flow and residual time series plots (unmarked) for identified and synthetic data sets to experts from GasDay to see if they can tell the difference between the two data sets. Figure 3.6 and Figure 3.7 are presented. The red marks are outliers. It is not easy for the experts to tell the difference. Thus, using the graphs with the approval of the experts, the time interval between synthetic outliers and the outlier magnitudes are similar to intervals between identified outliers and their magnitudes.

Apart from using graphs, we use statistics to show similarity between two data sets. We use the mean, exponential mean, median, standard deviation, and Mean Absolute Deviation (MAD). We have used those statistics because they easy to understand, and they are commonly used. In Table 3.3, we present inter-arrival times for one identified data set used to develop synthetic data sets. We picked 7 random synthetic data sets to show statistics for when to insert the next synthetic outliers as presented in Table 3.3. Table 3.5 shows the same statistics for the distribution of the

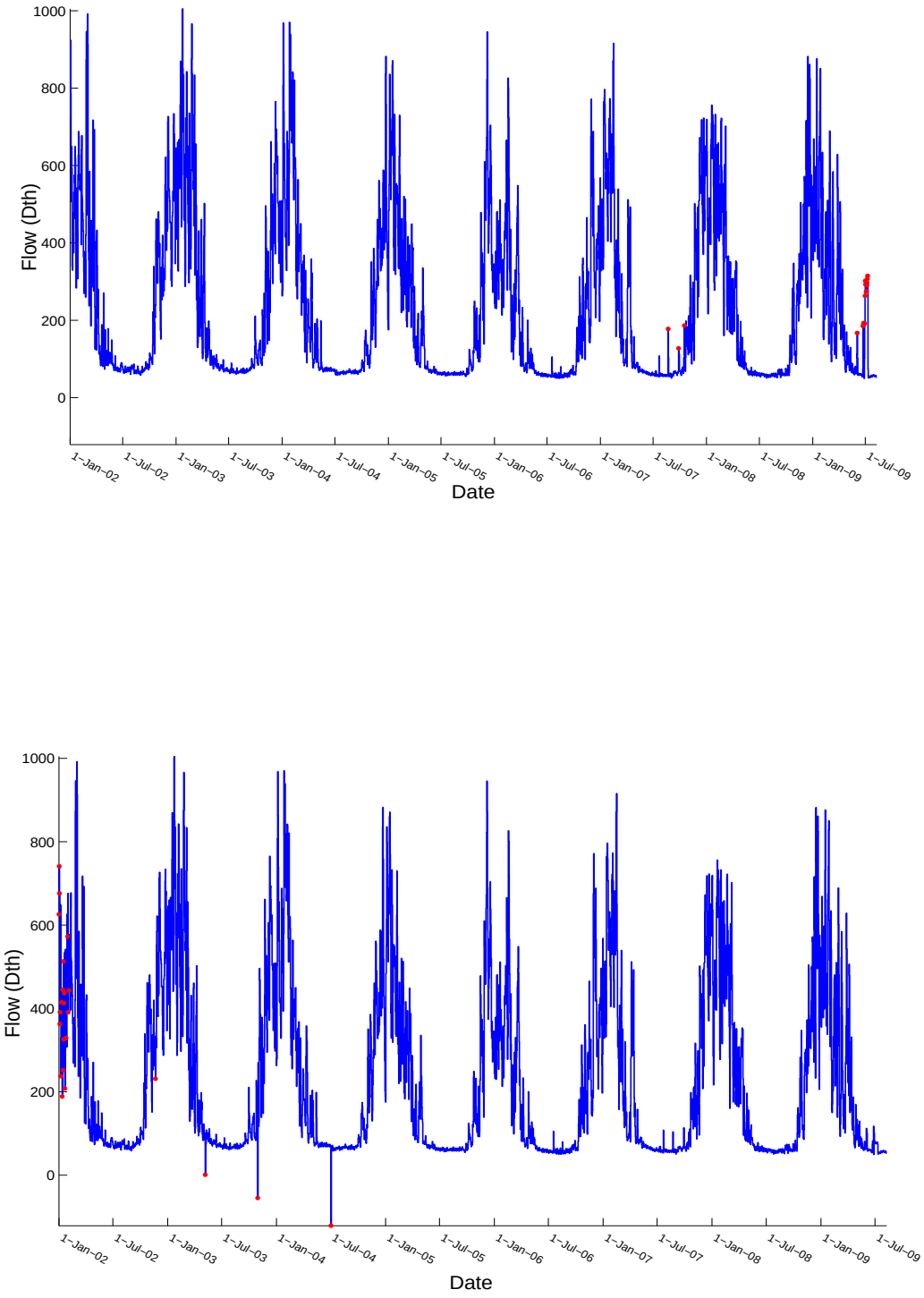

Figure 3.6: Identified and synthetic flow time series to show outlier's time interval similarity

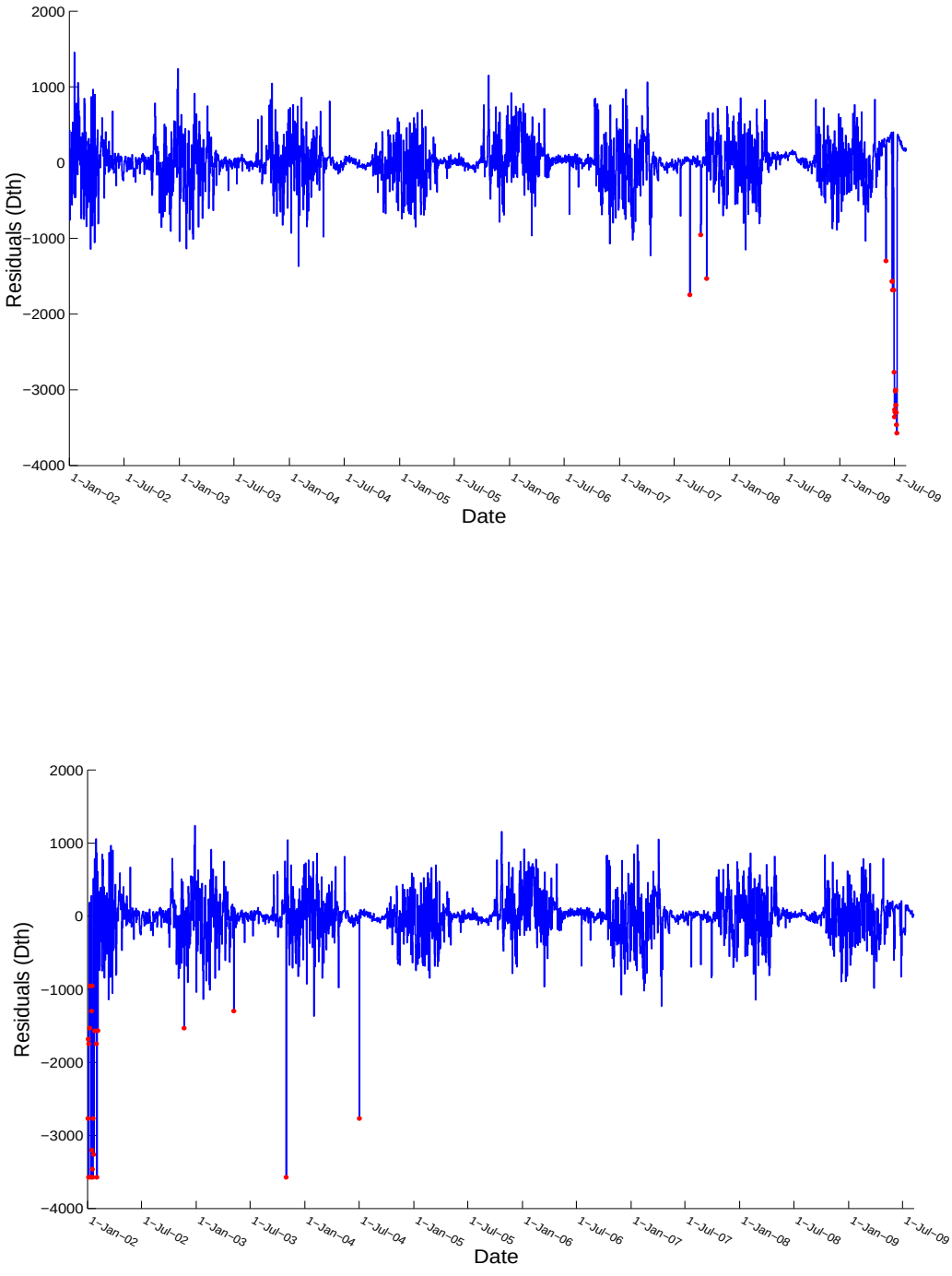

Figure 3.7: Identified and synthetic residual time series to show similarity in magnitudes

Table 3.3: Statistics for inter-arrival times for identified and synthetic outliers.

	Number of Outliers	Mean	Exp Mean	Stdev	Median	MAD
Identified	18	40.41	40.41	143.26	1.00	65.25
Synthetic	19	58.33	58.33	145.06	2.00	90.55
	25	37.75	30.64	95.01	1.50	56.93
	21	49.10	49.10	115.40	1.00	70.87
	18	54.58	54.58	92.01	3.00	73.18
	9	144.75	144.75	248.58	4.00	175.44
	17	66.06	66.06	126.22	3.00	94.96
	19	56.77	56.77	159.75	2.00	84.43

magnitude between identified and synthetic outliers. Table 3.4 and Table 3.6 show statistics for a total of 1000 synthetic data sets with an average of 46 synthetic outliers. By observing all the four tables, we see the values for statistics used are close to each other between identified data set and synthetic data set. The small difference between statistics is also observed in Table 3.4 when considering the average of 1000 synthetic data sets. Still we cannot draw conclusions based only on the small differences. Instead, we use the Kolmogorov-Smirnov statistical test to conclude that both data sets are similar (34). MATLAB has implemented the Kolmogorov-Smirnov test in a function called "kstest2", which compares the distributions of the values in the two data vectors $x1$ and $x2$. In our case $x1$ = identified data set, and $x2$ = synthetic data set. Its null hypothesis is that $x1$ and $x2$ are from the same continuous distribution. The alternative hypothesis is that they are from different continuous distributions. The result is 1 if the test rejects the null hypothesis at the 5% significance level; 0 otherwise (14). For all the data sets in Table 3.3, kstest2 returns a value of 0. This indicates that both data sets are similar. Since we have shown both data sets are similar, the next section presents how DBSCAN is adapted specifically to detect outliers in natural gas flow.

Table 3.4: Statistics for inter-arrival times of one identified data set and an average of 1000 synthetic data sets.

	Number of Outliers	Mean	Exp Mean	Median	Stdev	MAD
Identified	18	40.41	40.41	143.26	1.00	65.25
Synthetic	22	48.20	48.20	120.40	2.00	70.40

Table 3.5: Statistics for the magnitude of residuals for identified and synthetic outliers

	Number of Outliers	Mean	Median	Stdev	MAD
Identified	18	-2459.09	-2884.37	911.52	849.08
Synthetic	19	-2126.66	-1684.41	885.20	812.33
	25	-2542.69	-3009.24	906.63	810.24
	21	-2215.45	-3109.00	916.24	888.39
	18	-2506.55	-3001.53	867.82	819.91
	9	-2536.28	-3201.04	1024.44	901.74
	17	-2514.41	-3001.53	991.41	942.70
	19	-2270.63	-1745.64	892.95	830.55

3.2 Density Based Spatial Clustering of Applications with Noise Adapted to Natural Gas Flow

This section explains how Density Based Spatial Clustering of Applications with Noise (DBSCAN) is applied specifically to natural gas flow. The DBSCAN is implemented in MATLAB (pseudocode 3.1) to be used by the GasDay project to detect outliers in residuals from the models for any operating area. The results of its performance in detecting outliers from data sets with identified and synthetic outliers are presented in Chapter 4.

Table 3.6: Statistics for the magnitude of residual for one identified data set and an average of 1000 synthetic data sets

	Number of Outliers	Mean	Median	Stdev	MAD
Identified	18	-2459.09	-2884.37	911.52	849.08
Synthetic	22	-2115.45	-3009.00	946.24	867.30

In adapting DBSCAN to natural gas flow, we use daily residuals from the GasDay's mathematical models as a set of points instead of using daily flows. We use daily residuals from the models as explained in Chapter 2 because they capture most significant factors that affect the consumption of natural gas flow. In adapting DBSCAN to detect outliers from natural gas flow we use only residuals(r) as set of points to be clustered. The class developed can also work with two-dimensional points. In the event where we need to cluster two-dimensional points with different units, the two points need to be normalized. For example, dates and residuals has different units, days for dates (t) and Dth for residuals (r). We use this example to explain how the two points are normalized.

There are various ways to normalize data; we have used the median and median absolute deviation (MAD) approach to normalize dates and residuals (42). We decided to use this approach because it is robust to outliers. Let y be all residuals. Then a normalized residual is

$$r^{'} = \frac{r - median(y)}{MAD(y)}. \tag{3.1}$$

If we let z be all dates, then by replacing r with t and y with z in Equation(3.2), the normalized date is

$$t^{'} = \frac{t - median(z)}{MAD(z)}. \tag{3.2}$$

In our case we do not need to normalize residual so we have set of points P (r) that need to be clustered. DBSCAN also needs the two parameters $MinPts$ and Eps. These parameters are different for each operating area. The MATLAB tool that uses the DBSCAN algorithm can be set up to have default values for each operating area. These default values can be set using the GasDay expert's knowledge on a given operating area or by using the $k - dist$ approach discussed in Chapter 2. An expert using the tool can also change the parameter values when necessary. For example, using the data set with identified outliers from operating area N, different parameters values are used by the DBSCAN to detect those identified outliers. The goal is to find

parameter values that can detect most identified outliers at the same time not detecting non-outliers. The parameter values that enable the technique to detect most of the identified outliers are used as default parameter values. Hence, the next time this technique is used to detect outliers from operating area N with unknown outliers, the same default parameter values are used. We know those parameter values worked satisfactorily in detecting identified outliers, so we have confidence when we use them to detect unknown outliers.

DBSCAN algorithm pseudocode 3.1

```
%Aim:
    Clustering the data with Density-Based Scan Algorithm with Noise (DBSCAN)
%Input:
    SetOfPoints - data set residuals(r));
    Eps         - neighborhood radius
    MinPts      - minimal number of objects required to form a cluster)
%Output:
   A boolean vector with same length as SetOfPoint,
    1 - outlier detected, 0 - no outlier detected
classdef DBCluster
      properties
          SetOfPoints = 0;
          Eps = 0;
          MinPts = 0;
          IsPointVisited   = false;
          IsPointAnOutlier = false;
      methods
      function obj = DBCluster(SetOfPoints, Eps, MinPts)
                %Load files and handles errors
      ENDCONSTRUCTOR
      function [IsPointAnOutlier] = SCAN(thisClass)
            Clusterid = 0
             FOR each unvisited point P(r) in SetOfPoints
                mark P(r)as visited
                PListOfNeigbors = getNeighbors (SetOfPoints, P(r), Eps)
                 IF sizeof(PListOfNeigbors) < MinPts
                    mark P(r) as OUTLIER
                 ELSE
                    Clusterid = next cluster
                    expandCluster(SetOfPoints, P(r), N,...
                    Clusterid, Eps, MinPts)
                 ENDIF
             ENDFOR
      ENDSCAN
      function expandCluster(SetOfPoints, P(r), N,...
      Clusterid, Eps, MinPts)
            add P(r) to cluster Clusterid
            WHILE there is unvisited point P' in ListOfNeighbors
                mark P(r)' as visited
                PListOfNeigbors'  = getNeighbors(P(r)', Eps)
                IF  PListOfNeigbors' <= MinPts
                     PListOfNeigbors  = PListOfNeigbors joined with ...
                      PListOfNeigbors
                    add P(r)' to cluster Clusterid
                 ENDIF
            ENDWHILE
            RETURN
        end EXPANDCLUSTER
        function = getNeighbors (SetOfPoints, P(r), Eps)
            RETURN Eps-Neighborhood of P(r) in SetOfPoints as a list of points
        ENDGETNEIGHBORS
        function [] = Plot()
            Use various time series plots to display outliers characterized
            by GasDay's existing DBSCAN techniques.
        ENDPLOT
    end methods
end classdef
```

3.3 Main Outlier Detector

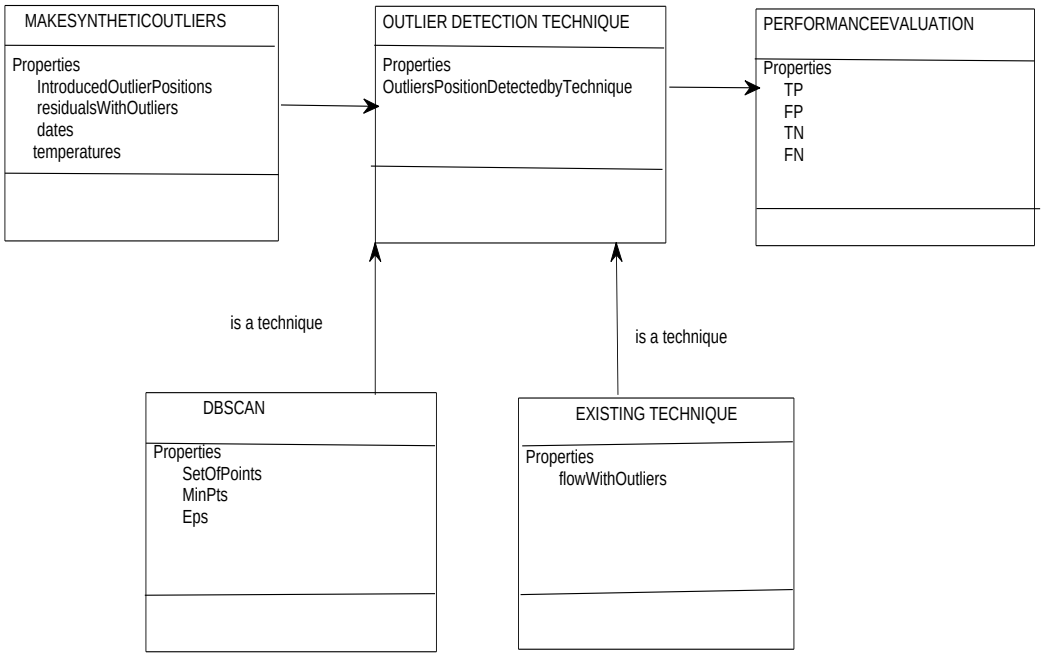

Figure 3.8: Class diagram displaying the classes used in this work

The previous section presented pseudocode for the modified DBSCAN (class) to show how Density Based Spatial Clustering of Applications with Noise is adapted specifically to natural gas flow. To run the DBSCAN to detect outliers from natural gas flow, the user needs to provide daily residuals, Eps, and $MinPts$.

Besides the DBSCAN class presented to show its algorithm, there other classes developed by this work. This section describes all classes written in MATLAB. Figure 3.8 shows a simple class diagram for the entire process, and the data flow diagram in Figure 3.9 describes the entire process.

We outline the classes shown in Figure 3.8:

- **MakeSyntheticOutliers**: The member functions in this class prepare synthetic data sets with the same distribution as identified data sets. They take unclean flow with identified outliers, clean flow with outliers removed, and residuals from

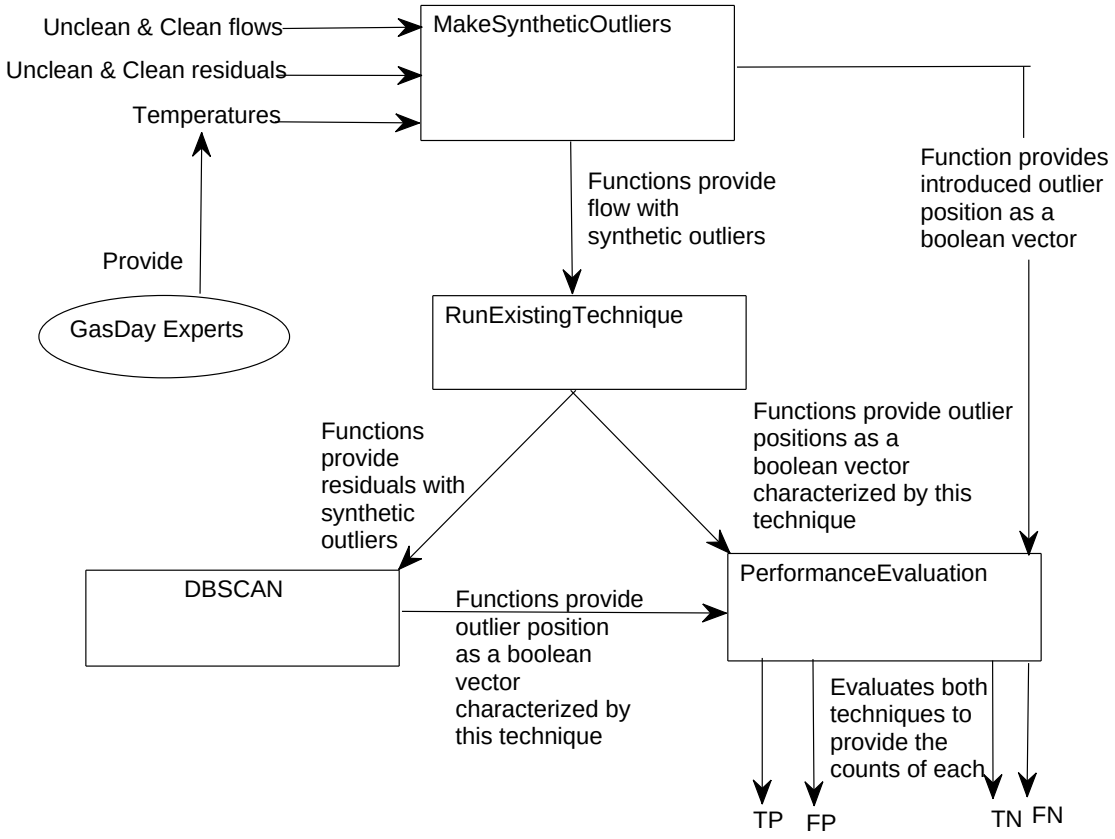

Figure 3.9: A data flow diagram describing the outlier detection process and its evaluation.

clean and unclean flows. They return flow with synthetic outliers and the positions of introduced synthetic outliers as a boolean vector with the same size as cleaned flow. The pseudocode 3.2 presents some member functions of this class without giving all the details.

```
                    MakeSyntheticOutliers pseudocode 3.2
%Aim:
    To make synthetic data sets using identified data set empirical
     distribution
%Input:
    Flows, residuals, temperatures, dates

-------------------------------------------------------------------------
%Output:
    Synthetic Outliers Positions and Flow with synthetic outliers

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

classdef MakeSyntheticOutliers
    properties
        fileSize                    = 0;
        syntheticOutliersPositions = 0;
        flowsWithSyntheticOutliers = 0;

    ENDPROPERTIES

    methods

        function thisMSO = MakeSyntheticOutliers(filePath, listOfFiles)
            IF none of the file is empty
                %Check to make sure files are read in correctly
            ENDIF
            syntheticOutliersPositions = GetSyntheticOutliersPositions();
            flowsWithSyntheticOutliers = GetFlowsWithSyntheticOutliers();
        ENDCONSTRUCTOR

        function syntheticOutliersPositions = GetSyntheticOutliersPositions()
             syntheticOutliersPositions(1:thisMSO.FileSize) = false;
             currentPos = 0;
             WHILE not at the end of the file
                Find when to insert an outlier using smoother CDF function
                IF (currentPos <= file size)
                    Insert an outlier
                ENDIF
                UPDATE currentPos;
             ENDWHILE
        ENDGETSYNTHETICOUTLIERSPOSITIONS

        function flowsWithSyntheticOutliers = GetFlowsWithSyntheticOutliers()
            FOR i = 1 to the length of the file size
                IF (syntheticOutliersPositions(i) == true)
                    update Clean flow with synthetic flow
                ENDIF
            ENDFOR
        ENDGETFLOWSWITHSYNTHETICOUTLIERS

        function fitValues = GetDeltaTfromFit()
            Find distribution for identified data set
            RETURN fitValues from a smoother CDF function
        ENDGETDELTATFROMFIT

    ENDMETHODS
ENDCLASS
```

- **RunExistingTechnique**: This class calls GasDay's existing technique to get residuals from a flow file with synthetic outliers, runs the technique to detect outliers in residuals, and returns outlier positions. It accepts a flow file with synthetic outliers. It returns residuals with synthetic outliers and the positions of synthetic outliers detected by the existing technique as a boolean vector with the same length as residuals.

- **DBSCAN**: This class runs the DBSCAN algorithm to detect outliers in residuals and returns outlier positions. It accepts the residuals from flow with synthetic outliers. It returns positions of synthetic outliers detected by the DBSCAN technique as a boolean vector of size n. Also, this class generates flow and residual time series plot to display the detected outliers. The pseudocode 3.1 presents modified DBSCAN class adapted specifically for natural gas flow.

- **PerfomanceEvaluation**: This class evaluates the performance of DBSCAN and GasDay's existing techniques in detecting outliers. It takes in the boolean vectors with introduced synthetic outlier positions, outlier positions characterized by GasDay's existing technique, and outlier positions characterized by our DBSCAN technique. It returns the count of True Positive, False Positive, True Negative, False Negative, Accuracy, Precision, Recall, and F1 measures for both techniques. The pseudocode 3.3 presents the member functions of this class.

PerfomanceEvaluation pseudocode 3.3

```
%Aim:
    To evaluate outlier detection techniques
% Input
   Introduced outliers file, characterized outliers file
   by GasDay and our DBSCAN
% Output
%       Provides classification metrics for outlier detection technique
--------------------------------------------------------------------------------

classdef PerfomanceEvaluation
    properties
        introducedOutliersPos                    = 0;
        outliersCharacterizedbyGasDayTechnique   = 0;
        outliersCharacterizedbyDBSCANTechnique   = 0;
    ENDPROPERTIES;

    methods

        function obj = PerfomanceEvaluation(IntroducedOutliersPos,...
            OCbyGasDayPos, OCbyDBSCANPos)
            %load the files, handle errors
            obj.IntroducedOutliersPos          = IntroducedOutliersPos;
            obj.OutliersCharacterizedbyGasDay  = OCbyGasDayPos;
            obj.OutliersCharacterizedbyDBSCAN  = OCbyDBSCANPos;
        ENDCONSTRUCTOR

        function [TP, FP, FN, TN] = BinaryValues(IntroducedOutliers,...
         OutliersCharacterized)
            %Outliers are outliers (flag the correct ones) 1 and 1
            RETURN TP
            %Normal data are outliers (flag incorrect ones) 0 and 1
            RETURN FP
            %Outliers are normal didn't flag them - bad) 1 and 0
            RETURN FN
            %Normal data are normal (didn't flag them - good) 0 and 0
            RETURN  TN
        ENDBINARYVALUES;

        function [Accuracy] = GetAccuracy(m)
           RETURN  Accuracy
        ENDGETACCURACY;

        function [Precision] = GetPrecision(m)
            RETURN  Precision
        ENDPRECISION

        function [Recall] = GetRecall(m)
            RETURN  Recall
        ENDGETRECALL

        function [F1Value] = GetF1Value(A)
            RETURN  F1Value
        ENDGETF1VALUE

    ENDMETHOD

ENDCLASS
```

By presenting the entire outlier detection and evaluation process, we have demonstrated the ability to make synthetic data sets from real data sets while making sure both data sets are similar. A technique that can detect synthetic outliers also will detect unknown outliers. More important, we have adapted and shown that DBSCAN can be used to detect time series outliers in natural gas flow.

Chapter 3 has presented the two strategies used to evaluate the performance of DBSCAN and GasDay's existing techniques. The discussion of both data sets with identified and synthetic outliers is presented. This chapter explains how the DBSCAN algorithm is adapted specifically to natural gas flow. The classes for the entire outlier detection process also are presented. In Chapter 4, we present results showing the performance of both techniques using the evaluation data sets.

CHAPTER 4

RESULTS OF THE PERFORMANCE OF DBSCAN AND GASDAY'S EXISTING TECHNIQUES

Chapter 3 explained the data sets used to evaluate the performance of DBSCAN and GasDay's existing techniques. Also, in Chapter 3 we explained how DBSCAN is adapted to detect outliers in natural gas flow. In this chapter, we evaluate the performance of DBSCAN and GasDay's existing techniques. The chapter starts by outlining the evaluation metrics for any classification technique.

4.1 Evaluation Metrics

Recall the following classification evaluation metrics from Section 1.5:

True Positive (TP) - an outlier is classified correctly as an outlier.

False Positive (FP) - correct value is classified as an outlier.

True Negative (TN) - correct value is classified as a correct value.

False Negative (FN) - an outlier is wrongly classified as a correct data.

Using a boolean vector, where true indicates an outlier is present, and false otherwise. Let

IO be a boolean vector indicating outliers introduced into the evaluation data set.

OD be a boolean vector indicating outliers characterized by a technique from the evaluation data set.

When the two boolean vectors are combined using &, true & true is true, true & false is false, and false & false is false. If '-' denotes not,

- TP = sum(IO & OD)
- FP = sum(-IO & OD)
- FN = sum(IO & -OD)
- TN = sum(-IO & -OD)

We know that FN costs more than a FP. It costs the Local Distribution Companies (LDCs) money if an outlier is incorrectly characterized as correct. A false negative might lead to under-charge or over-charge their customers, the end-users of natural gas flow. A false positive only costs LDCs time to investigate. The cost might be something like

$$\text{fcost} = 10\text{FN} + \text{TP}. \tag{4.1}$$

If we consider a technique with less 'cost' (where FP $\leq$ 10) to perform better, we use tables to present cost, the counts of each TP, FN, and FP as metrics to evaluate the performance of DBSCAN and GasDay's existing techniques. 1000 synthetic data sets and five data sets representing five different JOTO are used in this evaluation process. Also, we use time series and scatter plots to display outliers as characterized by both techniques.

4.2 Results for Synthetic Data sets

We use JOTO E from Section 4.3 as an identified data set that can be used to make synthetic data sets using the approach discussed in Chapter 3. Recall from Chapter 3, in making synthetic outliers, the data set with identified outliers should be similar to the one with synthetic outliers. The similarity is shown by using different

Table 4.1: Summary for 1000 data sets with synthetic outliers

Method	TP	FP	FN	cost
DBSCAN	39172	3159	6898	72149
GasDay	36814	2224	9256	94784

Table 4.2: Classification performance metrics

Technique	Accuracy	Precision	Recall	F1 value
DBSCAN	0.9969	0.9253	0.8502	0.8651
GasDay	0.9959	0.9430	0.7990	0.8651

graphs and statistics. Using this approach, the data set for JOTO E (presented in Section 4.3) is used to make 1000 data with synthetic outliers. A total of 46,319 flow values are inserted as synthetic outliers. Both DBSCAN and GasDay's existing techniques are applied to these data sets, and results are summarized in Table 4.1. $MinPts = 200$ and $Eps = 1.5$ is used by DBSCAN. We conclude that, although GasDay's existing technique has fewer counts for FP but DBSCAN performed better than GasDay'e existing technique because it characterized more outliers and has fewer counts for the FN value.

Using 1000 synthetic data sets, the results for performance metrics introduced in Chapter one are presented in Table 4.2.

4.3 Results from Identified Data Sets

In this section, we evaluate both techniques using five JOTO. In the first JOTO, 17 outliers are present. We see from Table 4.3, DBSCAN using $MinPts = 300$ and $Eps = 3.5$ characterized 14 as TP, 6 as FP, and 3 as FN. GasDay's existing technique characterized 12 as TP, 9 as FP, and 5 as FN. DBSCAN has cost = 44, and GasDay's existing technique has cost = 62. Hence, DBSCAN has performed better than the existing technique for JOTO A. The results for the remaining operating areas

Table 4.3: Classification metrics and cost for identified data sets:

OpArea	Technique	$MinPts$	Eps	TP	FP	FN	cost
JOTO A	GasDay			12	9	5	62
	DBSCAN	300	3.5	14	6	3	44
		150	1.7	17	46	0	17
		500	3.5	12	10	5	62
JOTO B	GasDay			5	12	2	25
	DBSCAN	350	4.7	6	3	1	16
		1000	4.7	12	12	6	82
		350	1.5	15	383	3	45
JOTO C	GasDay			8	8	3	38
	DBSCAN	350	6.1	5	3	6	65
		350	2.5	11	91	0	11
		3000	6.1	10	13	1	20
JOTO D	GasDay			15	1	6	75
	DBSCAN	150	3.2	19	9	2	39
		150	1	21	450	0	21
		450	3.2	11	0	10	111
JOTO E	GasDay		18	0	4	58	
	DBSCAN	250	5	16	1	6	76
		250	1	22	770	0	22
		1350	5	15	18	7	85

can be observed in Table 4.3. Observing costs values, we see that DBSCAN performed better than GasDay's existing technique for JOTO B and JOTO C, while GasDay's existing technique performed better on JOTO C and JOTO E.

Using red circles to represent values characterized as outliers, outliers characterized by DBSCAN and GasDay's existing techniques are presented by Figure 4.1 for JOTO A, Figure 4.2 for JOTO B, Figure 4.3 for JOTO C, Figure 4.4 for JOTO D, and Figure 4.5 for JOTO E.

We use the results presented for the five operating areas and the 1000 synthetic data sets in which the introduced outliers are known to conclude that DBSCAN has shown some improvement in detecting outliers over GasDays existing technique and merits further exploration.

The next chapter presents the conclusion and discusses a few ideas for improving the work presented by this thesis.

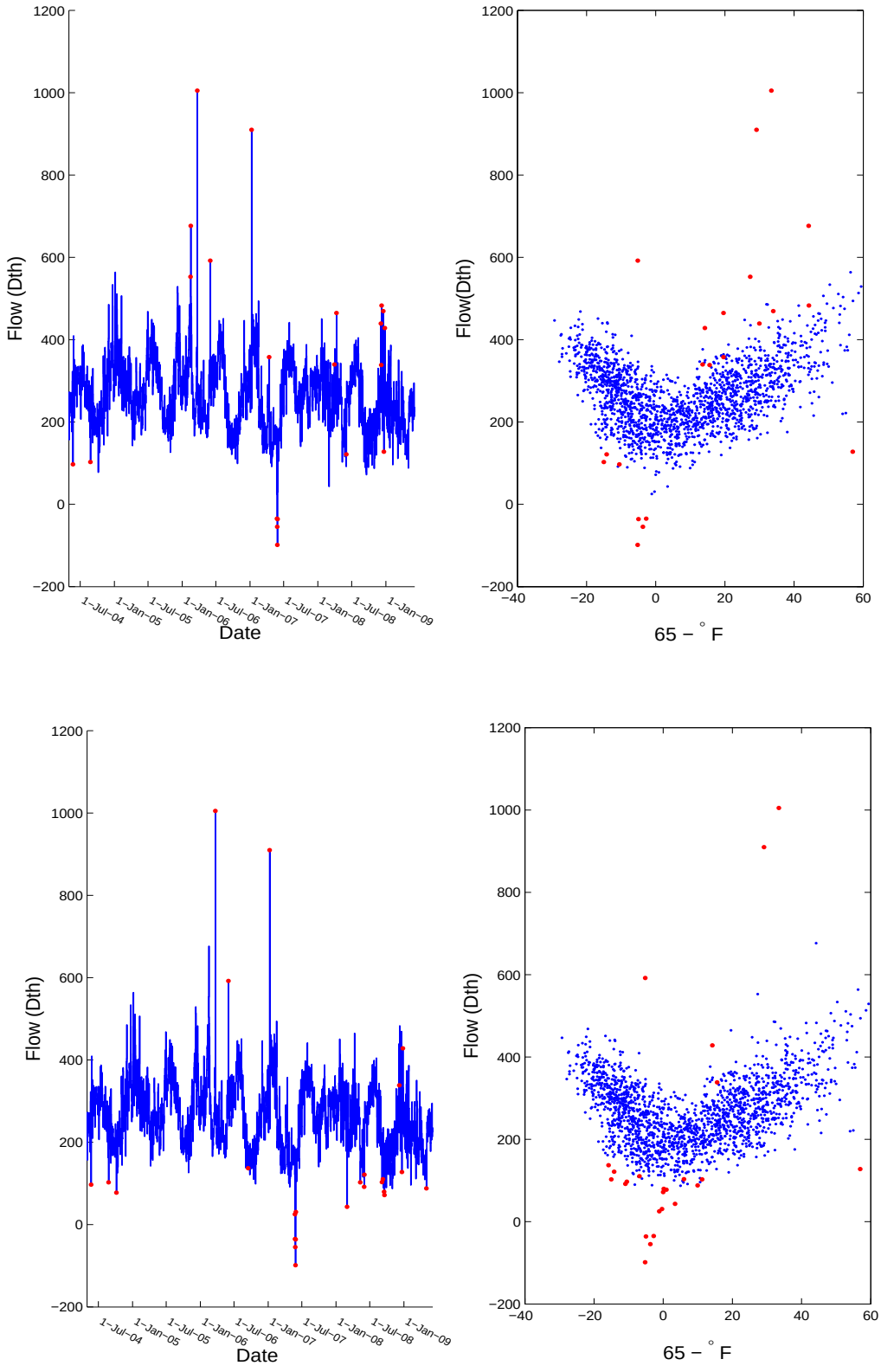

Figure 4.1: Flow time series and scatter plot showing outliers characterized by DBSCAN and GasDay's existing techniques for JOTO A.

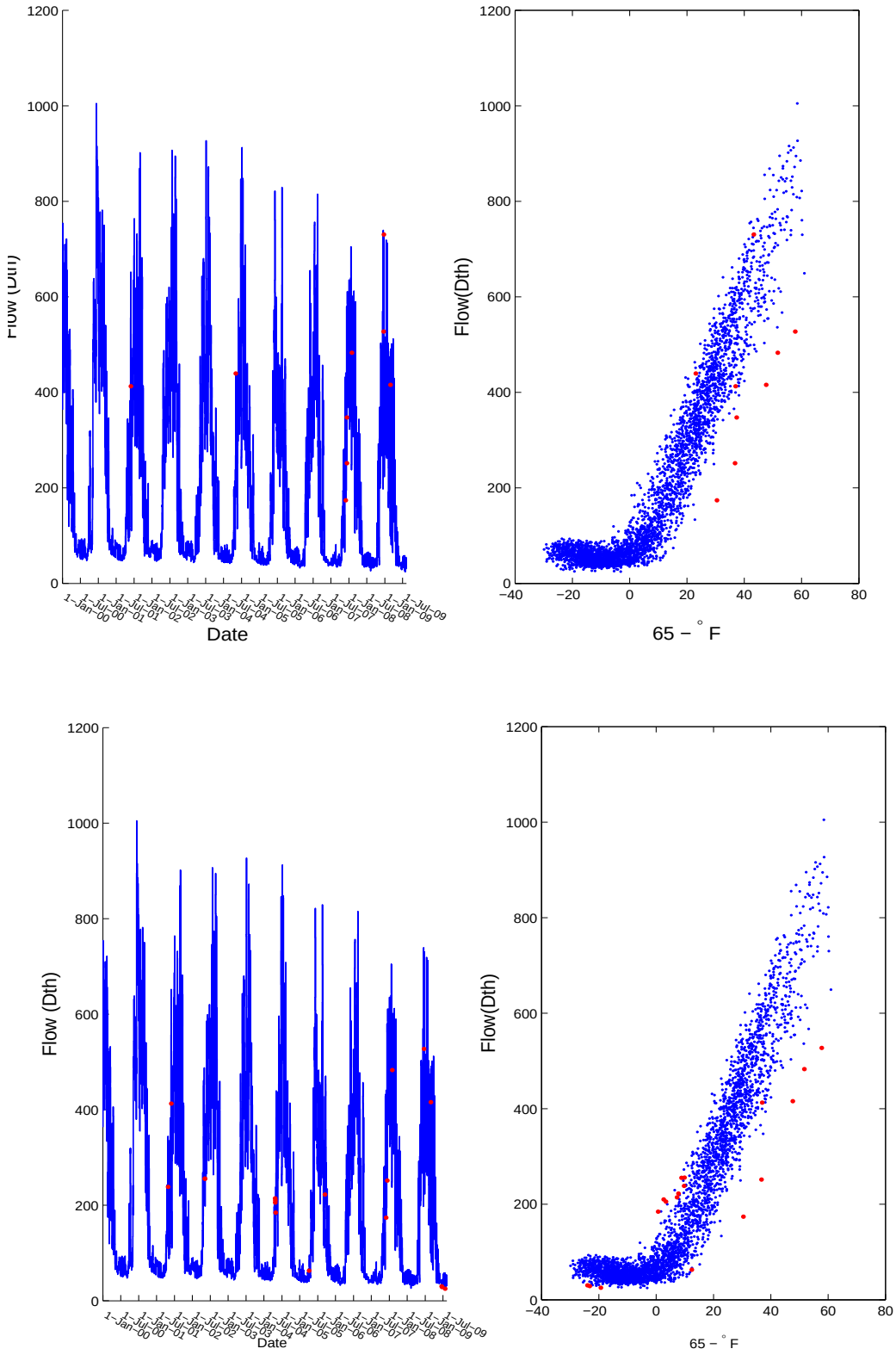

Figure 4.2: Flow time series and scatter plot showing outliers characterized by DBSCAN and GasDay's existing techniques for JOTO B.

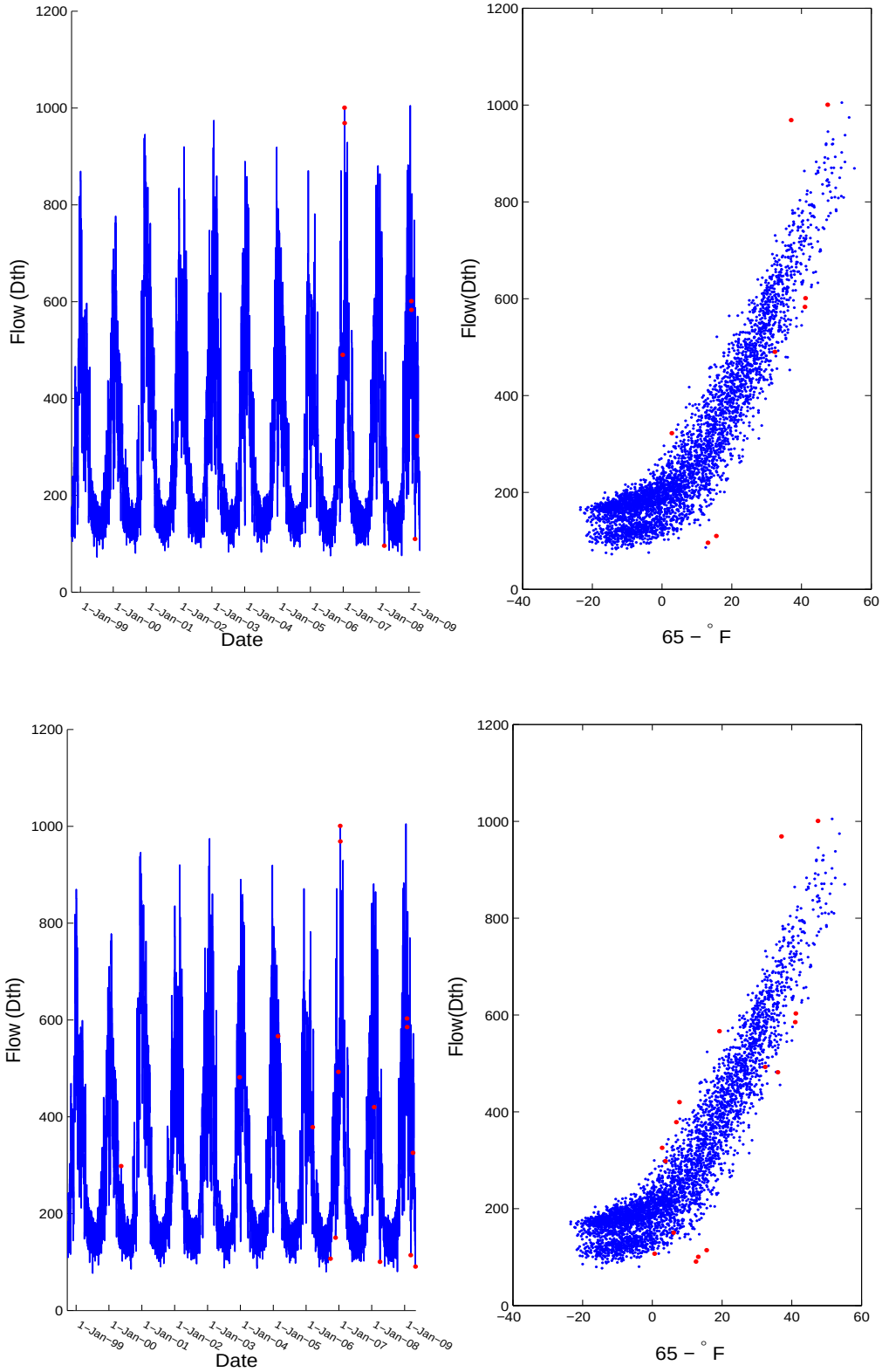

Figure 4.3: Flow time series and scatter plot showing outliers characterized by DBSCAN and GasDay's existing techniques for JOTO C.

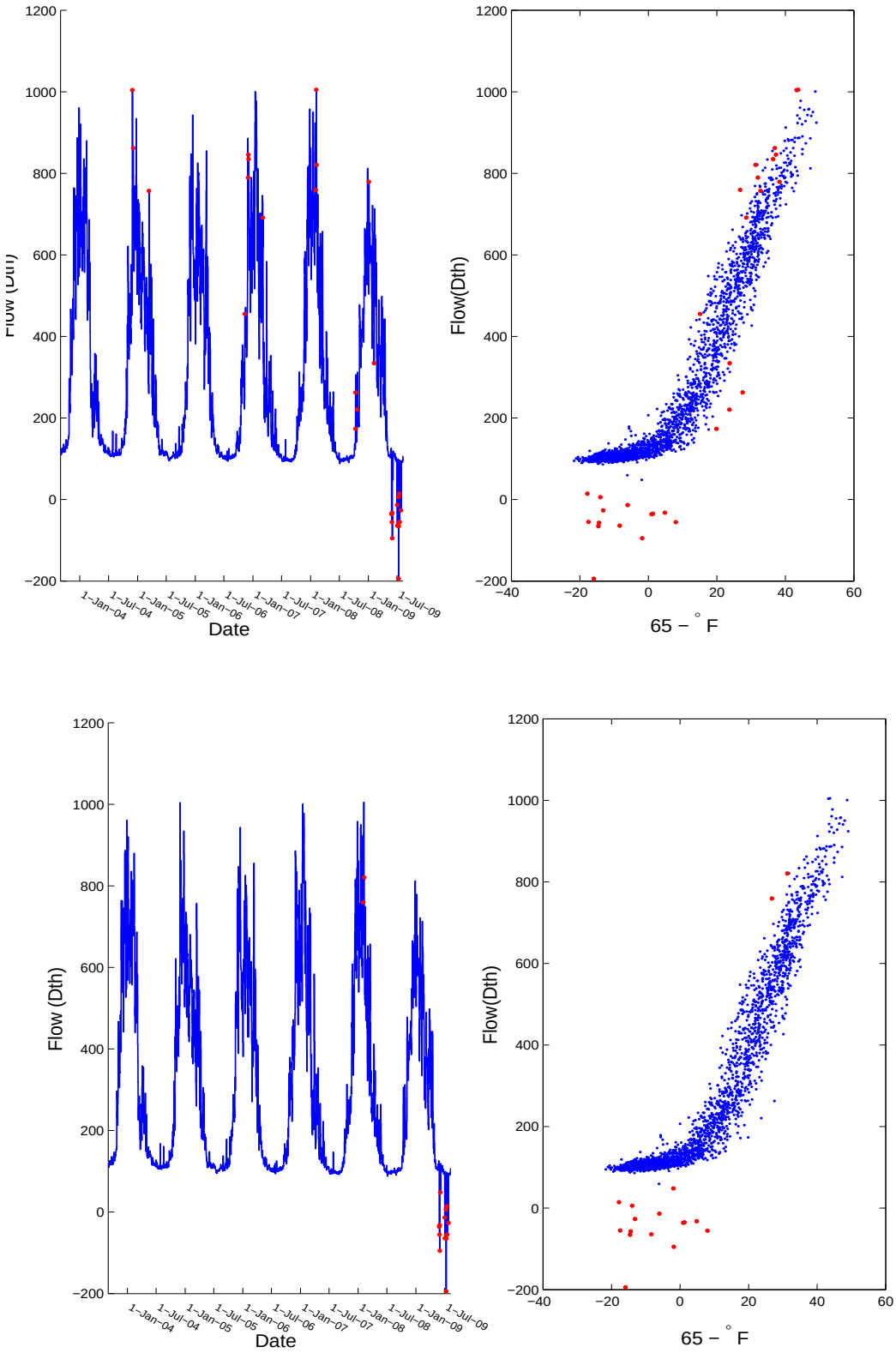

Figure 4.4: Flow time series and scatter plot showing outliers characterized by DBSCAN and GasDay's existing techniques for JOTO D.

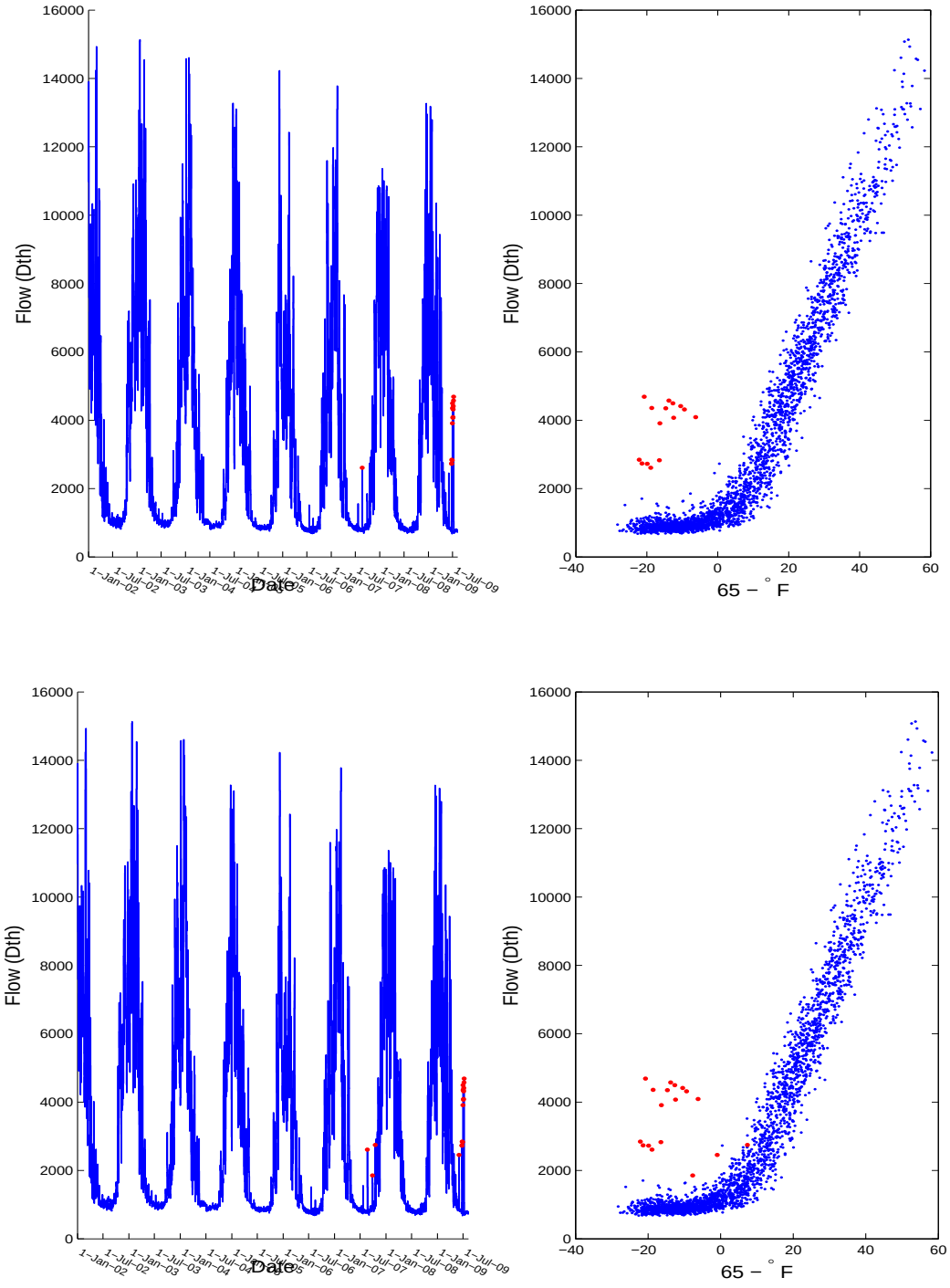

Figure 4.5: Flow time series and scatter plot showing outliers characterized by DBSCAN and GasDay's existing techniques for JOTO E.

CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH

5.1 Conclusions

Our goal was to develop a technique that detects outliers in time series data. We considered a technique that is not motivated by normally distributed data sets. The focus was to develop a technique to detect outliers more accurately than the existing GasDay outliers detection technique for temperature-sensitive operating areas. We have used a density-based clustering technique known as Density Based Spatial Clustering of Applications with Noise (DBSCAN), an idea from (3; 16; 25). This technique can be used to detect outliers in any time series data set. In this thesis, we have adapted the DBSCAN algorithm to natural gas flow to detect outliers in residuals from the mathematical models used by the GasDay project.

We have presented two strategies to develop evaluation data sets that can be used to evaluate the performance of an outlier detection technique. The first class of data sets contains outliers identified by experts who have domain knowledge of the data. The second data set contains synthetic outliers with the same empirical

distribution as identified outliers. Both data sets are shown to be similar using statistics and graphs.

We evaluated the performance of DBSCAN and GasDay's existing techniques by using four metrics; True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) and cost values from the cost function. Accuracy, precision, recall, and F1 measures were used as classification metrics to assess the performance of both techniques using 1000 synthetic data sets. As presented in Chapter 4, we found that three out of five identified data sets, DBSCAN showed increased performance over GasDay's existing technique. Although GasDay's existing technique performed better in some of the 1000 synthetic data sets with cost value = 56.78%, overall DBSCAN with cost value = 42.22% has shown improvement in performance. We conclude that DBSCAN has shown some improvement in detecting outliers over GasDay's existing technique and merits further exploration.

5.2 Future Research

To further this work, an approach of using Bayesian probability can be considered in developing synthetic data set(s) (37). In detection of time series outliers, a new clustering algorithm based on distance and density, which is an enhancement of the DBSCAN technique can be explored (25). Also, a gas flow software package like Flow-Cal can be investigated to see if it can be used by the GasDay project to detect

outliers in natural gas flow (28). These ideas are discussed by the following subsections.

5.2.1 Developing Synthetic Data Sets Using Bayesian Probability

Although this work has shown both synthetic and identified data sets are similar, there are still improvements which can be made in developing synthetic data sets.

We suggest using a Bayesian probability (a conditional-based probability) approach. The probability of a hypothesis given the data (the posterior) is proportional to the product of the likelihood times the prior probability (37). In making a synthetic data set, this work assumed the time of arrival between outliers was independent of each other. There is no need to make the same assumption when using the Bayesian probability approach. In flow measurements, if a meter is stuck, we expect the same flow value is reported for several days until the meter is fixed. We expect abnormal flow values as a result of events such as hurricane or storms. Also, we know that we cannot have a negative flow value, so inserted outliers should not have negative flows. All these can be used as conditions when using Bayesian probability in developing synthetic data set(s).

In Section 1.2, outliers observed at the GasDay project are presented. Researchers can choose one type and develop synthetic outliers for that one type using approach used by this work or Bayesian probability approach proposed.

5.2.2 A New Clustering Algorithm Based on Distance and Density

This work has studied and adapted the DBSCAN algorithm in detecting outliers in natural gas flow. Although this technique has shown improvement over the existing GasDay's technique, it is difficult to set its two density thresholds (Eps and $MinPts$) properly. Yu in (25) explains a new DBSCAN based on k-nearest neighbors (KNN) as an algorithm which merges KNN and DBSCAN to enhance DBSCAN. Using the entropy theory, local parameters (Eps, $MinPts$) of each fuzzy cluster (FC) are determined using unsupervised learning techniques. Each local Eps is mapped to the global Eps, and each FC is clustered separately (20; 25).

5.2.3 Using Gas Flow Measurement Software to Detect Outliers

The DBSCAN algorithm proposed in this work to detect outliers in natural gas flow can work with any time series data sets. In detecting outliers specifically in natural gas flow, the use of gas flow measurement software (e.g., Flow-Cal) can be investigated. In (28), Flow-Cal, Inc., reports it is the industry leader in Electronic Flow Measurement (EFM) data management software and says Local Distribution Companies are using Flow-Cal software to manage the flow of gas measurement data from field operations throughout their organizations. Perhaps the same software can be used by the GasDay project to detect outliers in natural gas flow.

In this thesis, we outlined various DBSCAN applications, and we were able to

introduce a new application by adapting DBSCAN to detect time series outliers from natural gas flow. We have shown that DBSCAN can be used to detect outliers in a time series data set. We recommend other researchers to try using DBSCAN to detect outliers from a time series data set. Also, the future research ideas we proposed can be studied and applied to different data sets to compare their performance in detecting outliers over the DBSCAN. The idea of developing synthetic data set similar to real data sets is also discussed. Researchers with limited real data sets but in need of more data can explore and take advantage of the approach discussed.

REFERENCES

[1] WISN-TV, "Marquette students' research helps you save money on heat", 2008, `http://www.youtube.com/watch?v=rVLzhjjRSKA`.

[2] Ronald H. Brown, Brian Marx, and George F. Corliss, "Mathematical Models for Gas Forecasting", Tech. Rep. GasDay 129, Marquette University. Department of Electrical and Computer Engineering., Milwaukee, WI., July 2005.

[3] Pang Ning Tan, Michael Steinbach, and Vipin Kumar, *Introduction to Data Mining*, Addison Wesley, Boston, MA., 2006.

[4] Stefatos George and Hamza A. Ben, "Cluster PCA for Outliers Detection in High Dimensional Data", *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference*, pp. 3961–3966, 2007.

[5] Ronald K. Pearson, "Outliers in Process Modeling and Identification", *IEEE Transactions on Control Systems Technology*, vol. 10, no. 1, pp. 55–63, January 2002.

[6] Rohan Kennedy, "Detecting Outliers and Meter Anomalies in Natural Gas Customer Flow Data", December 2006, MS Thesis, Marquette University, Department of Electrical and Computer Engineering, Milwaukee, WI.

[7] Zhengxin Chen, *Data Mining and Uncertain Reasoning: An Integrated Approach*, Wiley-Interscience, New York, NY., 2001.

[8] Peter J. Rousseeuw and Annick M. Leroy, *Robust Regression and Outlier Detection*, Wiley, New York, NY., 2003.

[9] M. Abramowitz and A. I. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, NYC, NY, 1964.

[10] D. N. Joanes and A. C. Gill, "Comparing Measures of Sample Skewness and Kurtosis", *The Royal Statistical Society (Series D)*, vol. 47, pp. 183–189, 1998.

[11] F. J. Kenney and S. E. Keeping, *Mathematics of Statistics, Pt. 2, 2nd ed*, Van Nostrand, Princeton, NJ., 1951.

[12] Deb Partha and Martin Sefton, "The Distribution of a Lagrange Multiplier Test of Normality", *Economics Letters*, vol. 51, pp. 123–130, 1996.

[13] Carlos M. Jarque and Anil K. Bera, "A Test for Normality of Observations and

Regression Residuals", *International Statistical Review*, vol. 55, no. 2, pp. 163–172, 1987.

[14] MATLAB, "Matlab help", 2008.

[15] John Robert Taylor, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, University Science Books, Sausalito, CA, 1999.

[16] Martin Ester, Hans Peter Kriegel, Jorg Sander, and Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.

[17] Domenica Arlia and Massimo Coppola, "Experiments in Parallel Clustering with DBSCAN", 2001.

[18] Caihong Yang, Fei Wang, and Benxiong Huang, "Internet Traffic Classification Using DBSCAN", *Information Engineering, International Conference on*, vol. 2, pp. 163–166, 2009.

[19] Stefan Brecheisen, Hans-Peter Kriegel, and Martin Pfeifle, "Efficient Density-Based Clustering of Complex Objects", *Data Mining, IEEE International Conference on*, pp. 43–50, 2004.

[20] Yasser El-Sonbaty, M. A. Ismail, and Mohamed Farouk, "An Efficient Density Based Clustering Algorithm for Large Databases", *Tools with Artificial Intelligence, IEEE International Conference on*, pp. 673–677, 2004.

[21] Tao Yang and Hongli Xu, "An Image Index Algorithm Based on Hierarchical Clustering", *Intelligent Information Hiding and Multimedia Signal Processing, International Conference*, pp. 459–462, 2009.

[22] Kai Li, Lan Wang, and Lifeng Hao, "Comparison of Cluster Ensembles Methods Based on Hierarchical Clustering", *Computational Intelligence and Natural Computing, International Conference on*, vol. 1, pp. 499–502, 2009.

[23] Wenjun Wang, Junying Zhang, Jin Xu, and Yue Wang, "A Graph-Based Approach for Clustering Analysis of Gene Expression Data by Using Topological Features", *Computer Science and Information Engineering, World Congress on*, vol. 1, pp. 559–563, 2009.

[24] Zhan Xue, Feng Cen, and Li Wei, "A Weighting Fuzzy Clustering Algorithm Based on Euclidean Distance", *Fuzzy Systems and Knowledge Discovery, Fourth International Conference on*, vol. 1, pp. 172–175, 2008.

[25] Xiaopeng Yu, Deyi Zhou, and Yan Zhou, "A New Clustering Algorithm Based on Distance and Density", *Services Systems and Services Management, 2005. Proceedings of ICSSSM '05. 2005 International Conference*, vol. 2, pp. 1016–1021, 2005.

[26] Ronald H. Brown, T. M. Richardson, and J. E. Buchanan, "Forecasting daily sendout demand with artificial neural networks", in *American Gas Association Operating Section Pre-Print Proceedings*, Cleveland, OH, May 1999, pp. 131–139, American Gas Association.

[27] F. K. Lyness, "Gas Demand Forecasting", *The Statistician*, vol. 33, no. 1, 1984.

[28] FlowCal-Inc, "What is New Gas Measurement Software", *Pipeline and Gas Journal*, 2006.

[29] Hancong Liu, Sirish Shah, and Wei Jiang, "On-line Outlier Detection and Data Cleaning", *Computers and Chemical Engineering*, vol. 28, pp. 1635–1647, August 2004.

[30] K. A. Sumithradevi, M. N. Vijayalakshmi, Annamma Abraham, and Dr. Vasanta, "Evaluation of Fuzzy ARTMAP with DBSCAN in VLSI Application", *World Academy of Science, Engineering and Technology*, vol. 36, 2007.

[31] B. E. Fowlkes and L. C. Mallows, "A Method for Comparing Two Hierarchical Clusterings", *IEEE Journal of the American Statistical Association*, vol. 78, no. 383, pp. 553–569, 1983.

[32] Somsak Phattarsukol and Pornsiri Muenchaisri, "Identifying Candidate Objects Using Hierarchical Cluster Analysis", *Eighth Asia-Pacific Software Engineering Conference, 2001. APSEC 2001*, pp. 381–389, 1998.

[33] E. Stefanakisa, "Net-dbscan: Clustering the Nodes of a Dynamic Linear Network", *International Journal of Geographical Information Science*, vol. 21, pp. 427–442, 2007.

[34] G. Marsaglia, W. Tsang, and J. Wang, "Evaluating Kolmogorov's Distribution", *Journal of Statistical Software*, vol. 8, 2003.

[35] P. S. Bradley, Usama Fayyad, and Cory Reinna, "Scaling Clustering Algorithms to Large Databases", 1998, American Association for Artificial Intelligence.

[36] Yansheng Lu, Yufen Sun, Guiping Xu, and Gang Liu, *A Grid-Based Clustering Algorithm for High-Dimensional Data Streams*, Springer-Verlag, Berlin, Germany, 2005.

[37] Carlin Bradley and Louis Thomas, *Bayesian Methods for Data Analysis, Third Edition*, Chapman and Hall, Boca Raton, FL., 2008.

[38] Michael H. Kutner, Christopher J. Nachtsheim, and John Neter, *Applied Linear Regression Models*, McGraw-Hill, Boston, MA., 2005.

[39] S. M. Aldenderfer and R. K. Blashfield, *Cluster Analysis*, Sage Publications, Newbury Park, CA., 1984.

[40] Ronald K. Pearson, *Mining Imperfect Data*, Society for Industrial and Applied

Mathematics, Philadelphia, PA., 2005.

[41] Abonyi Jonas and Feil Balazs, *Cluster Analysis for Data Mining and System Identification*, Birkhauser, 2007.

[42] H. Charles Romesburg, *Cluster Analysis for Researchers*, Lulu Press, North Carolina, 2004.

[43] Leonard Kaufman and Peter J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley-Interscience, New York, NY., 1990.

[44] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan Kuafmann Publishers, New York, NY., 2001.

[45] Damodar N. Gujarati, *Basic Econometrics*, McGraw Hill, Boston, MA., 2003.