AN UNSUPERVISED CLUSTER: LEARNING WATER CUSTOMER BEHAVIOR USING VARIATION OF INFORMATION ON A RECONSTRUCTED PHASE SPACE

by

Michele Rae Bizub Malinowski, M.S., P.E.

A Dissertation Submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

May 2018

ABSTRACT
AN UNSUPERVISED CLUSTER: LEARNING WATER CUSTOMER BEHAVIOR USING VARIATION OF INFORMATION ON A RECONSTRUCTED PHASE SPACE

Michele Rae Bizub Malinowski, M.S., P.E.

Marquette University, 2018

The unsupervised clustering algorithm described in this dissertation addresses the need to divide a population of water utility customers into groups based on their similarities and differences, using only the measured flow data collected by water meters. After clustering, the groups represent customers with similar consumption behavior patterns and provide insight into 'normal' and 'unusual' customer behavior patterns. This research focuses upon individually metered water utility customers and includes both residential and commercial customer accounts serviced by utilities within North America.

The contributions of this dissertation not only represent a novel academic work, but also solve a practical problem for the utility industry. This dissertation introduces a method of agglomerative clustering using information theoretic distance measures on Gaussian mixture models within a reconstructed phase space. The clustering method accommodates a utility's limited human, financial, computational, and environmental resources. The proposed weighted variation of information distance measure for comparing Gaussian mixture models places emphasis upon those behaviors whose statistical distributions are more compact over those behaviors with large variation and contributes a novel addition to existing comparison options.

ACKNOWLEDGEMENTS

Michele Rae Bizub Malinowski, M.S., P.E.

The research has been made possible by generous support from my employer, Badger Meter, Inc., in cooperation with the Electrical and Computer Engineering (EECE) department at Marquette University.

I would like to thank the GasDay™ laboratory at Marquette University, for feedback, encouragement and comradery throughout this process.

Also, my deepest gratitude to my advisor, Dr. Richard Povinelli, and to my committee, Drs. Ronald Brown, George Corliss, Mike Johnson, Stephen Merrill, and Elaine Spiller.

Finally, I wish to dedicate this work to my daughters, Sandra and Charlie, who have taught me the value in explaining concepts on an elementary level; and to my husband, Jeff, for his continued support in this adventure.

"It's the questions we can't answer that teach us the most. They teach us how to think."

— Patrick Rothfuss, The Wise Man's Fear

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

AMA – Advanced metering analytics

AMR – Automatic meter reading

AMI – Automated metering infrastructure

ARI – Adjusted rand index

AWWA – American Water Works Association

$\cos\theta_{\mathbf{ab}}$ – Cosine distance between vectors $\mathbf{a}$ and $\mathbf{b}$

DMA – District metering area

EMD – Earth mover's distance

FMI – Fowlkes-Mallows distance

FNN – False nearest neighbors

GMM – Gaussian mixture model

H – Entropy

$\hat{H}$ – Entropy estimated by the volumes of ellipsoid hulls within a GMM

IWA – International Water Association

KL – Kullback-Liebler distance

L_p – Minkowski distance of order p

$\mathrm{Med}_{\mathbf{X}\neq 0}$ – Non-zero median of the time series

MI – Mutual information

$\hat{MI}$ – Mutual information estimated by the volumes of ellipsoid hulls within a GMM

$\mathcal{N}_d\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ – Multivariate normal distribution with d dimensions

$\rho\left(A, B\right)$ – Distance between clusterings A and B

RPS – Reconstructed phase space

t – Time reference

T_m – Time delay of embedding dimension m within a phase space

VI – Variation of information

$\hat{VI}$ – Variation of information estimated by the volumes of ellipsoid hulls within a GMM

V_k – Volume of the k^{th} ellipsoid hull within a GMM

w_{A1} – Component weight of the first component in GMM A

$wH\left(A\right)$ – Weighted entropy of GMM A

wMI – Weighted mutual information

wVI – Weighted variation of information

$\mathbf{X}$ – Time series of data

$\hat{\mathbf{X}}$ – GMM estimation of the time series

$x_{t+\tau}$ – Single measurement in the time series

$\mathbf{Y}$ – Vector of points in $\mathbf{X}$ used for embedding into a phase space

1 RESEARCH MOTIVATION AND SUMMARY OF WORK

1.1 Overview

The unsupervised clustering algorithm described in this dissertation addresses the need to divide a population of water utility customers into groups based on their similarities and differences, using only the measured flow data collected by water meters. Two motivations drive this work - a commercial motivation to provide useful segregation of customer data and an academic motivation to create a new method of comparing two models. The agglomerative clustering method considers the practical limitations of a utility's resources, accommodating limitations in resources. The academic contribution of this work, the weighted variation of information distance measure, presents a novel component-weighting scheme for emphasizing components of Gaussian mixture models with compact distributions.

1.2 Commercial Motivation

Since 2011, more than 25% of the US has coped with drought conditions. In California, one of the most severely affected areas, over 45% of the state has experienced drought conditions over the same period, increasing to over 90% for 2016 [1]. Even though the 2016-2017 winter brought record precipitation to California, filling surface reservoirs, the subterranean aquafers remain low. These depleted groundwater sources supply between 30 and 46 percent of the California water needs [2]. In response to the long-term drought, the state has outlined an aggressive water conservation plan via The Water Conservation Act of 2009, Senate Bill X7-7, targeting 20% reduction in overall water consumption per capita by December 31, 2020. Municipalities have responded with conservation ordinances introducing severe restrictions of water use including irrigation system flow limits, watering date/time restrictions, and punitive monetary fines for violations [3]. These restrictions and conservation projects require timely

water consumption data and processing to enforce the ordinances, as well as educational and targeted communication with the water consumers.

The water utilities need an easy method to identify which customers' behavior is within the accepted normal patterns, and which customers' behavior is wasteful, fraudulent, or in violation of regulations. The unsupervised clustering algorithm presented in this research fills the need for grouping customers by behavior. This assists the utility to determine customers needing additional scrutiny and those that do not. The output of this algorithm is a hierarchical diagram grouping all customers compared with each other using an information-theoretic distance measure based on the temporal behavior patterns observed within the collected flow measurements.

1.2.1 Water Industry Background

This section describes the infrastructure components and the system management of a typical North American water utility and explains the data collection methods and how data is aggregated for this research.

1.2.1.1 North American Water Utilities

While over 15 million American households rely upon private well sources for water [4], the remaining 110 million households are connected to public water supplies. Likewise, the vast majority of commercial and industrial applications use public water supplies. Those public and municipal water utilities must carefully monitor the water they provide for public safety, billing, and resource management.

In North America, water is typically collected from surface reservoirs, freshwater rivers or lakes, or subsurface wells. This untreated source water is transported using gravity or pumps to a treatment plant prior to distribution. The treatment plants remove particulate matter through sedimentation, coagulants, flocculation, and, finally, filtering. After the particulates have been

removed, the water is disinfected and stored for distribution [5], [6]. Water distribution and storage systems in the United States often employ elevated tanks, which serve to store water and to provide a pressure head sufficient to support the gravity-fed distribution system temporarily in the event of a pump outage. From the storage system, the water flows through large diameter transmission lines to local distribution pipes commonly called “water mains.” The mains crisscross the entire distribution zone to supply the service lines as well as fire service connections (hydrants). Service lines are the connection points to individual properties and can vary in diameter from a typical residential ¾” line to an 8” or larger line providing water to an industrial facility [7]. Figure 1.1 shows an example of a utility water system for a small community: Anytown, USA. The source feed from a well or the lake is indicated in blue. This water is treated at the utility and distributed through transmission lines (grey). The transmission lines reduce into distribution mains (green) and finally into supply lines (orange) of various sizes based on end use. Wastewater reclamation and treatment is not shown on this diagram.

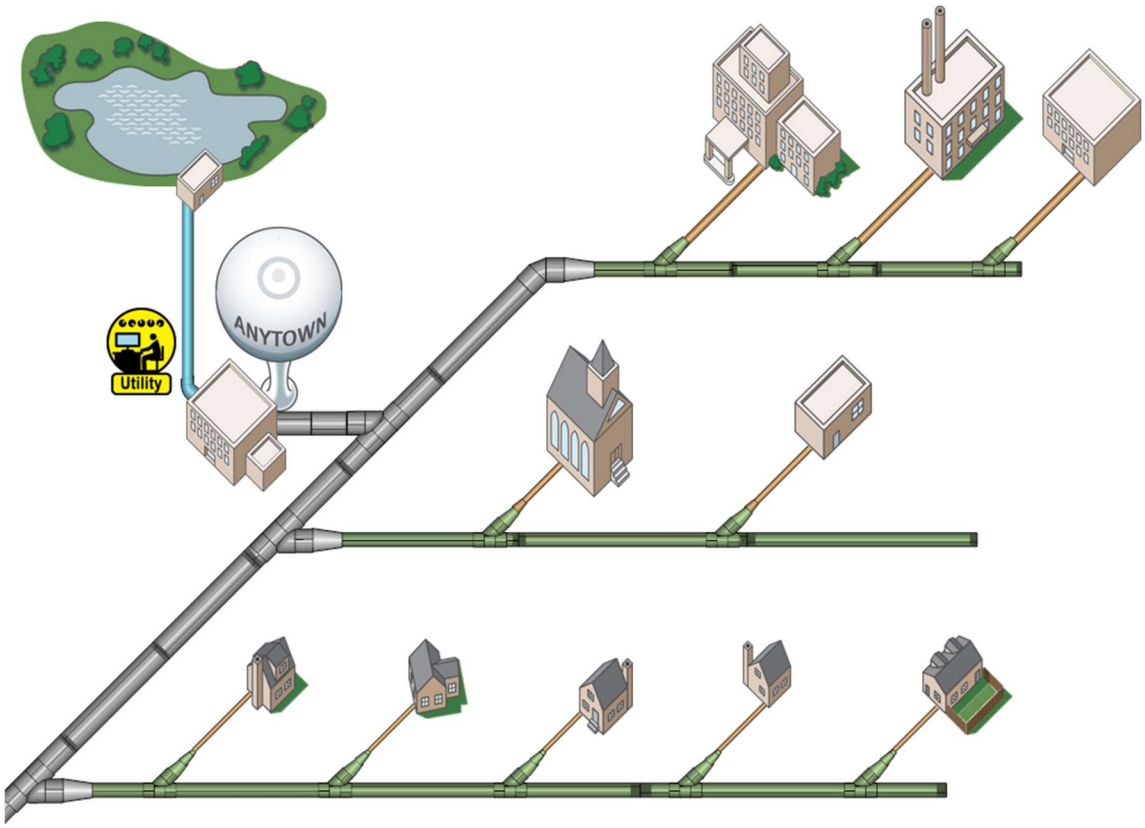

Figure 1.1 Anytown, USA, water distribution network

1.2.1.2 Meter Reading

Since the mid-1800s, public and private utilities have provided water to residential, commercial, and industrial customers. Service fees, such as labor costs, system maintenance, and infrastructure improvements, as well as the actual volume of water delivered impact the operational costs of the utility. In North America, these costs are represented as infrastructure fees in addition to the fee for the actual water delivered to the premises. Since the volume of water delivered relates directly to the cost of supplying that water, the billing systems must also account for the delivered volume. In North America, two billing schemes dominate: Non-metered and metered billing.

Non-metered billing occurs when the water usage is calculated based on the property size, intended use, and amenities. A typical non-metered bill considers number of bathrooms,

total square footage, and whether the location is residential or commercial. Some non-metered bills are computed from a neighborhood master meter, dividing the total volume delivered to the neighborhood by the number of properties serviced.

In contrast, metered billing indicates the service line has a mechanical (developed mid 1800s) or electronic (developed early 1900s) flow meter installed [8]–[11], measuring the flow volume to the property. Some properties may have more than one service line, and thus, more than one meter used to compute the total bill.

Originally, all meters used mechanical register dials, similar to an odometer on a vehicle, to record the total volume passing through the meter. These mechanical dials required a meter reader to physically locate each meter, visually read the numerical total displayed, and manually record the measured flow. Installations where the meter was located within a basement or crawlspace required the meter reader to enter the premises or to leave a postcard for the homeowner to transcribe the meter reading and return to the utility. As with any manual process, this introduced errors within the data.

Later, the mechanical registers included optical coupling mechanisms to read the dial placement, generating a digital signal to report the measured flow to other systems – telephonic, inductive coupling, and radio frequency networks [10]. Over the last few decades, water utility companies have begun installing automated meter reading (AMR) systems to further simplify the process of meter reading, decrease manual labor, and reduce transcription errors within collected data [12]. These systems allow more frequent reporting of measured demand at the individual customers, while simultaneously reducing the manual effort of physically looking at each meter to record the volume measured.

1.2.1.3 Meter Data Management

Figure 1.2 illustrates a basic AMR system for Anytown, USA. The system has radio transmitters connected to each utility meter, sending a radio telegram of the status and recorded volume at predetermined intervals. Handheld- or vehicle-mounted mobile radio receivers are connected to a computer and database system for automatic collection, transmission, and aggregation of the data. All properties in Anytown (residential, commercial, industrial, municipal) have a water meter and a radio installed either in a pit in the yard or as a remote unit in a basement. The image identifies each water meter and radio as a Badger Meter, Inc. ORION® device. The utility may monitor these radio transmissions through walk-by systems with handheld receivers, drive-by systems with vehicle-mounted receivers, or through a fixed-network using a cellular, LAN, Wi-Fi, or proprietary backhaul installed throughout the area. Regardless of the collection method, every meter is monitored by the utility, and all reads are collected and aggregated for evaluation and billing.

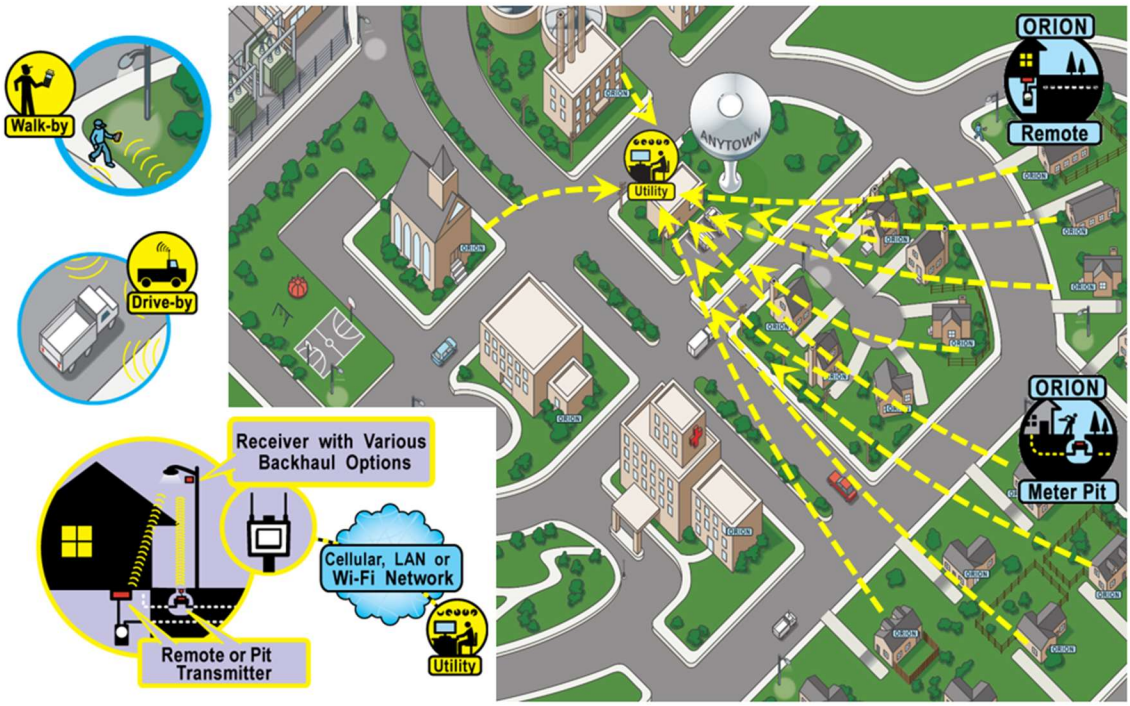

Figure 1.2 Anytown, USA, automatic meter reading (AMR) and automated metering infrastructure (AMI) systems

Recently, utilities have begun installing automated metering infrastructure (AMI) systems to support the desire for increased meter reading frequency and near-real-time monitoring of the water distribution system. AMI systems generally involve a network of fixed-location radio receivers, called collectors, to monitor and aggregate the transmitted details of all meters within a service area. The records are sent through a network to the utility office, where back-end software allows the utility to investigate performance of the system and trends in the customer behavior. The meter and AMR/AMI system manufacturers produce various proprietary back-end software packages, each with its own algorithms and features. These systems are sometimes called "Advanced Metering Analytics" (AMA) systems, and one example is illustrated in Figure 1.3. AMA systems nearly eliminate the manual effort of data collection for billing purposes and enable the utility to monitor daily, hourly, or even sub-hourly flow measurements. The AMA system components include the same brass or composite meters used in any meter reading

application. These meters have an encoder to monitor the total volume and a local display. The encoders also translate the readings via a digital signal to a radio or cellular endpoint. The meter, encoder, and endpoint are all installed at the property, either in an underground meter pit or within the structure. Data collection occurs through walk-by (not pictured), mobile, or fixed-network receivers; and the data is transmitted securely via backhaul to a hosted service. Utilities and customers can view their water usage via an internet portal or smartphone app. This allows users to monitor their consumption, identify leaks and inefficient use, and manage billing.

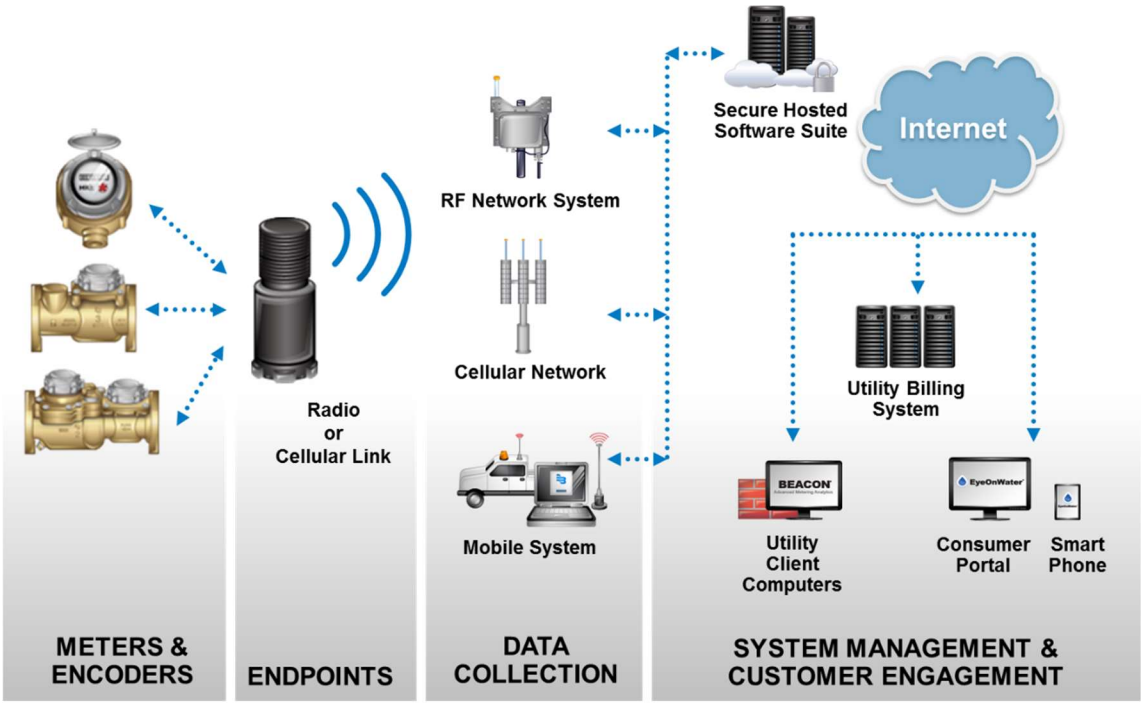

Figure 1.3 Badger Meter, Inc. BEACON® advanced metering analytics system

Whether the readings originate from manual reads, mobile receivers, or fixed networks, all are imported into the utility's water data management system. This system contains the meter readings, billing records, customer information, and other system details. Depending upon the data management system, statistical features, trending, and basic analysis are available for the system managers. Currently, many water data management systems are focused on providing system status, flow measurements, and error reporting information to the utility, but only limited

access to this data for the consumers. The newest reporting systems, as shown in Figure 1.4, supplement a monthly or quarterly bill expressing a cumulative total consumption with consumer portals to the database. Figure 1.4 shows a sample customer record indicating a modest increase in water usage over the previous week, a small leak, and the measured volume over time in multiple charts. These portals present current and historical measurements in hourly or sub-hourly increments, allowing customers to better interpret the impact of their habits on water consumption. Expanding access to interpreted water consumption may improve compliance with ordinances and can empower customers to make better conservation decisions [13], [14]. Profiling reports interpret each household's habits, alert the consumer to a pattern that is wasteful or inefficient, and suggest a list of investigations or actions to take. Algorithms such as those presented in this research will continue to improve the quality of information provided to the customers, empowering them to make informed decisions.

Figure 1.4 Badger Meter, Inc. EyeOnWater® customer data portal

Data for this research has been collected by the Badger Meter, Inc. BEACON® Advanced Metering Analytics system, capable of collecting, recording, and reporting water consumption across utilities with hourly or finer resolution of readings and direct-to-consumer interfaces. This platform collects water records, providing a unique database of time-based consumption data for all types of water utility customers.

1.3 Academic Motivation

Identifying similarities in time series data is not a problem unique to the water industry. The implementation of clustering within time-series data spans technology, utilities, finance, and

art, among other fields [15]. Beyond water, other utilities have similar needs for classifying gas or electric energy consumers based on their behavior [16]–[19]. Financial institutions use behavioral spending patterns to identify credit card fraud [20]. Telephonic and conferencing systems cluster sounds to identify speakers within a meeting for transcription [21], [22]. Music classification systems determine the genre, artist, or component instruments within a work [23], [24].

As sensors and data collection become pervasive in our daily lives, more opportunities for time-series classification will be identified. In many of the applications mentioned above, some type of probabilistic model represents the data. These models are then compared to each other, and a distance between pairs of models is computed and used for assigning clusters. No single approach to clustering data models is a panacea for every domain. The research described here outlines a new combination of processing steps to create a model, comparing multiple models using information-theoretic distances, and weighting the importance of components within the model based on the statistical parameters of the components themselves.

1.4 Summary of Contributions

This dissertation contributes a new method for processing water meter time series data as well as a novel approach to weighting components within a probabilistic model. Existing research on water meter flow measurements focuses primarily upon very high resolution data (sub-minute, minute, or quarter-hour) with a goal of identifying specific behaviors (shower, laundry, irrigation); whereas this research has focused upon the hourly data. The clustering method described here is more suitable and flexible for implementation in a real system where sometimes the optimal number of clusters cannot be used due to limitations of the utility finances, resources, or staffing. This application fits Gaussian mixture models after casting the time series into a reconstructed phase space. The agglomerative hierarchical clustering step combines customer models using information-theoretic distance measures.

The weighting variation of information distance measure presented here offers another option for comparing Gaussian mixture models, allowing more emphasis to be put on the components of the model with compact distributions. This scheme follows the desire to discount behaviors with large variation and weight behaviors with highly repeatable patterns more heavily.

1.5 Dissertation Outline

This introductory chapter has discussed the incentive for a new approach to the unsupervised clustering of time series. It has also introduced the water utility domain, as well as providing a summary of the contributions within this research. The rest of this dissertation is organized in a non-traditional manner. Experimental methods are presented in order of computation rather than as specific chapters identifying the theoretical background, individual methods, experiments, and results. Each section provides references to historical work, relevant mathematical definitions, and examples applying the concept to data collected from the water meter records.

Chapter 2 introduces the data, data cleaning, and initial steps taken prior to clustering. This includes preprocessing for normalization and filtering. The high-dimensional data is reduced into a more manageable set of representative Gaussian mixture models to facilitate the clustering process. The individual existing techniques described in Chapter 2 are described as a foundation for the novel combination of methods presented in this work.

Chapter 3 describes the clustering process. Hierarchical agglomerative clustering, is explained in detail. Several distance measures are described and defined, including the variation of information distance originally applied to the Gaussian mixture models within this research. A thorough explanation is provided of the output linkage and dendrogram. Numerous examples using both cartoon and actual data are presented to aid the explanation, as well as references to supporting research.

The first clustering experiments naturally follow in Chapter 4. Each experiment goal is described, followed by the experiment itself, the results, and discussion on the findings. These results identify the need for a more robust clustering approach with fewer variations in the groupings of customers between subsequent trials. Chapter 5 presents the novel model component-weighting scheme. The theory, mathematical definitions, and systematic examples are thoroughly described. Experiments of Chapter 4 are repeated using the component weighting. A comparison between the results with and without the component weighting illustrates the improvement in consistency of the clustering behavior. Finally, Chapter 6 reviews the findings and contributions, and suggests paths for future research.

2 DATA, PREPROCESSING, AND DIMENSIONAL REDUCTION

The next few chapters describe the methods and experiments in detail. Figure 2.1 shows a flow diagram of the techniques applied in this work and calls out the highest-level description of each step. The steps on the left are presented in the same order within this dissertation.

To begin, this chapter defines time-series data and the notation used to describe it. An explanation of the specific data collected for this experiment follows, along with additional background about the data collection and the limitations of the system. Then, we discuss the data preprocessing. Preprocessing includes data cleaning, normalization of values, and basic filtering to prepare for evaluating different customers to each other. Each customer has a large set of individual points, requiring an intermediate dimensional reduction step to keep the computational complexity manageable for clustering. The dimensional reduction includes casting the data into a phase space and representing the customer by a probabilistic model within this space. This chapter concludes with each customer individually represented as a model.

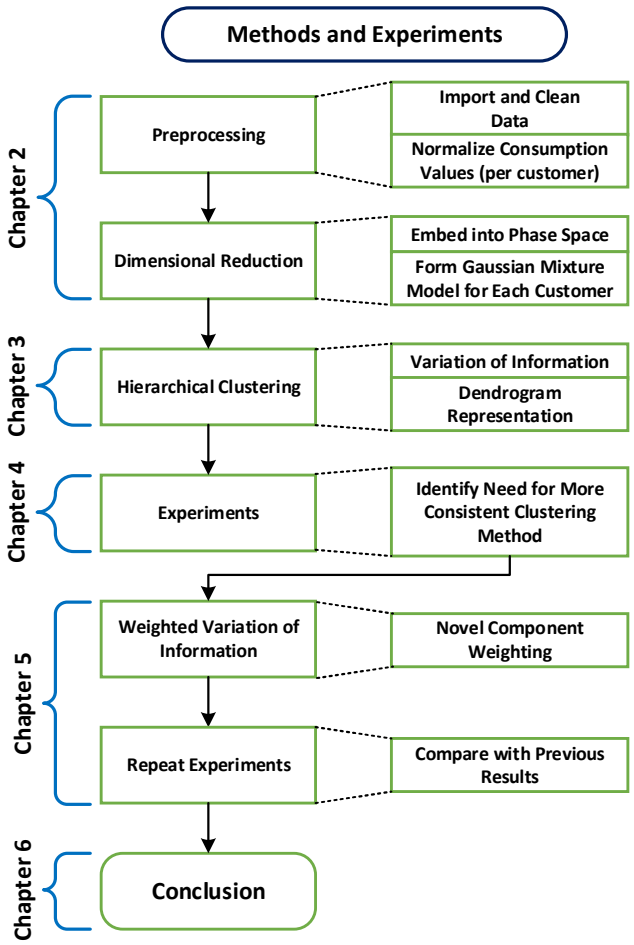

Figure 2.1 Flow diagram of the methods and experiments in this research.

2.1 Time-Series Data

Within any large set of data, groups of similar items exist. Unsupervised clustering is the process of training a program to identify these groups within unlabeled data. Much of the research historically has been applied to clustering of static, unchanging, data; but within the last few decades, clustering of dynamic time-series data has become more common [15], [25]–[27].

A time series is an ordered collection of measurements for a variable, indexed with respect to time. The measurements are collectively defined as a set:

$$\mathbf{X}=\{\ldots,x_{t-\tau},x_t,x_{t+\tau},x_{t+2\tau}\ldots\}. \tag{2.1}$$

Each measurement $x_{t+\tau}$ occurs at a time t, with a measurement interval τ. While a constant value of τ is not *required*, the research presented here uses only data with constant measurement intervals.

The behavior of $\mathbf{X}$ is governed by some probability function of the system. This probability function may or may not be known, and it may or may not be stationary. In the case of this research, the probability function is not known, may be different for every customer, and is not stationary over time. The natural changes over time may involve one or more periodic signals (seasonal, weekly, daily) or an increasing or decreasing trend. Clustering of time series data can be performed on raw input data, extracted features, or data models based on the posterior probability computed from collected samples. Frequently, the input data requires additional preprocessing steps prior to clustering, such as time warping, changes to the sample rate, cleaning, or filtering [15], [26].

When comparing large sets of raw time-series data, the computational and storage space requirements quickly become unmanageable. Many options exist to reduce the dimensionality, sometimes through time-domain transformations such as symbolic aggregate approximation [28], [29], and perceptually important points or landmarks [30]. Other dimensional reduction techniques use an eigenvalue or correlation matrix evaluation such as principal component analysis, curvilinear component analysis, Sammon maps [31], or spectral clustering [32]. The model-based approaches often transform the data from the time domain into another domain such as a frequency representation [33]; a probability-based approach such as the hidden Markov model [34] or Gaussian mixture model, described in detail later; or a phase space, based on time-lag sampling of the data [35]. A model may implement one or more of those transformations. Each technique strives to reduce the large data set into those characteristics that are most useful to classify the set among its peers, reducing the complexity of the clustering problem.

2.2 Description of Data Used in These Experiments

Research data have been drawn from the cloud-based Badger Meter, Inc. BEACON® Advanced Metering Analytics (AMA) system. This database of hundreds of utilities maintains historical records for meters with equipment details, measured flow, time stamps, and status information throughout the life of the meter. Hundreds of thousands of endpoints are tracked daily in the system. Unique identifiers for the meter, radio endpoint, and customer label each record within the system, but have been anonymized for this research, and no personally identifiable information is presented here. All records are used with permission from their respective owners.

The source data used in this study comprise a group of 99 meters from a Midwestern utility with approximately 4 years of historical records archived within the BEACON® AMA system. Experiments and examples using a single customer have been drawn from this collection as well. For this research, all customers are assumed to have resided in their homes for the entirety of the sample period, with no changes in ownership of a property. This is a naïve approach, and future work should investigate methods to identify changes related to ownership or commercial usage of a property.

All collected water records for this study have a 1-hour reading interval. This is in contrast to historical meter reading with standard monthly or quarterly intervals, influenced by billing periods [36]. Studies of electrical utility customer behavior [31], [37]–[42], and the work of Willis et al. [43] focus primarily upon high resolution (sub-minute) data sampling. Cardell-Oliver uses hourly interval data for identifying end uses within a consumption record [44]–[46], but other uses of medium resolution data have not been identified. Data collected from the BEACON® AMA system includes the flow volume as well as status alarms from the meter, radio, and collector. These status alarms may include continuous flow, no reported flow, naïve disaggregation, and communication errors. Data with naïve disaggregation and communication

errors are excluded from the study, while those with continuous flow and no reported flow are included. Only the recorded flow volume and time stamps are preserved as inputs to the clustering process. The other status alarms regarding flow are not used.

The water meters in this study translate the mechanical measurement of a volume into a numerical value on the meter, much like an automotive odometer tracks the miles traveled by a vehicle. Each hour, the total measured gallons is sent to a recording device, and this value is stored in the database. The result is a volume of water measured by the meter during an hour of time, described as gallons per hour. However, the flow rate of the water is not necessarily constant during the measurement period. A three-gallon toilet flush once per hour will appear the same as a small continuous leak. The discrete measurements are also presented on charts that may be interpreted as continuous flow when, in fact, the water consumption occurred sporadically. Extended periods of zero consumption may occur between usage periods, but become undetectable at this resolution. The recorded measurements throughout this dissertation describe flow as "gallons per hour," but the reader is encouraged to remember the limitations of this unit of measure and the resolution of the meters.

2.3 Data Preprocessing

Data preprocessing refers to the steps needed to make the collected data work correctly in an algorithm. This includes removing erroneous data caused by errors not in the scope of the research, often called *data cleaning*; scaling or normalizing the data, if required for comparing series with different bounds; and filtering data to remove high-frequency noise added by the behavioral jitter of imperfect human schedules. The main purpose of preprocessing is to allow the algorithm to function correctly without using erroneous inputs to generate erroneous outputs.

2.3.1 Data Cleaning

Due to the imperfect nature of real world applications, the data collected by any sensor network are riddled with anomalies. Commonly encountered water utility meter anomalies include:

- Missing data caused by equipment, data processing, or communication channel failures
- Broken distribution, supply, or local pipes
- Naïve disaggregation by the data collection system due to interrupted communications
- Human data entry errors, especially of manually read or manually entered flow measurements
- Mismatched meter and encoder/register units
- Outliers and unusual consumption patterns. For this research, outliers are of particular interest as a group to themselves.
- Incorrectly installed meters
- Improperly sized meters
- Encoder or register errors, including mechanical dial jitter and calibration errors
- Damaged or mechanically obstructed meters

Data cleaning is a broad topic left for other researchers to explore in detail [47], and only those methods used to prepare the data for this study are addressed here.

The algorithm in this experiment is not designed to accommodate large gaps in the data. During the data cleaning process, the longest continuous set of hourly measurements without any missing, aggregated, or negative flow data is selected for evaluation, discarding other data. The BEACON® AMA system can disaggregate values naïvely, but it stores an internal flag for those

records, permitting the detection and removal of disaggregated data that would otherwise affect the overall performance of the method. Future research should address this limitation and accommodate periods of missing data.

2.3.2 Normalization

Since the goal of this research is to find behavioral patterns that are similar, the actual volume of consumption is less important than the magnitude of a particular event with respect to the typical consumption for the customer. This approach identifies and groups customers with similar daily or weekly behavior routines. To achieve this, the data is normalized individually per customer record prior to clustering, as explained below. The normalized data provides behavioral clusters, customers with the same schedule, but perhaps have fewer household occupants or a smaller scale business. These clusters allow the utility to identify usage patterns that span segments of the population.

Water meters in residential properties sit idle for many hours of the day while the occupants are at work or asleep, resulting in a majority of recorded data to indicate zero volume. Likewise, many commercial meters sit idle during the evening or early morning hours when the business is closed. Figure 2.2 shows a few days of recorded hourly consumption values for a residential customer. The periods indicating zero consumption coincide with time spent sleeping or at work during the weekdays. A histogram to the right shows the recorded values, as well as the overall median and non-zero median, described below.

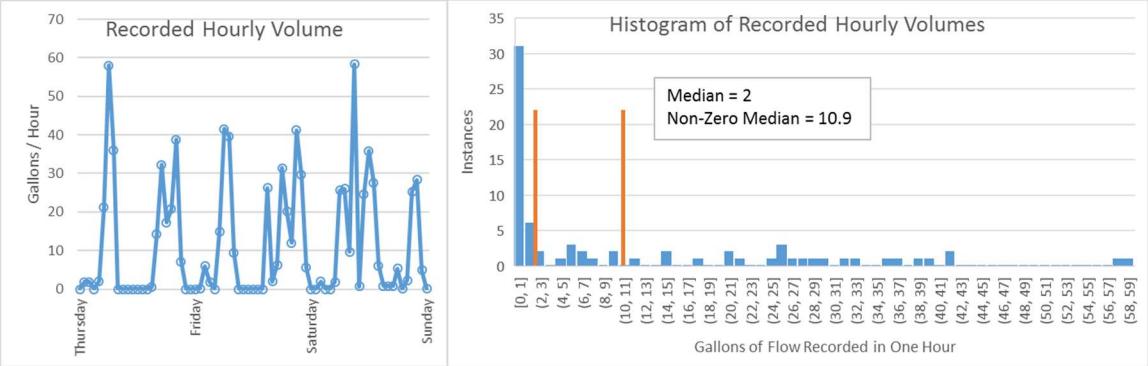

Figure 2.2 Recorded hourly consumption values and histogram of values for a residential customer

The zeros themselves are not unusual, but the number of zero measurements can create computational problems. Traditional normalization by median or mean is deceptively small, due to the large quantity of zeros in this data; or erroneous, due to divide-by-zero errors. Instead, the median of non-zero values is used – this is the number associated with the 50th percentile of consumption for all non-zero consumption records. Using the time series

$$\mathbf{X}=\{\ldots, x_{t-\tau}, x_t, x_{t+\tau}, x_{t+2\tau},\ldots\}, \tag{2.2}$$

take the median of the set of $\mathbf{X}$, excluding values in $\mathbf{X}$ equal to zero

$$\mathrm{Med}_{\mathbf{X}\neq 0} = \mathrm{Median}\left(\mathbf{X}\cap\bar{0}\right). \tag{2.3}$$

$\mathrm{Med}_{\mathbf{X}\neq 0}$ is the non-zero median. Dividing the original data by $\mathrm{Med}_{\mathbf{X}\neq 0}$ produces the normalized data:

$$\mathbf{X}_{\text{normalized}} = \frac{\mathbf{X}}{\mathrm{Med}_{\mathbf{X}\neq 0}}. \tag{2.4}$$

Normalizing the recorded values for each customer in this manner allows comparison between customers of different size (number of household members or size of business). The

comparison then identifies common behavioral patterns, regardless of the volume of the consumption pattern. Figure 2.3 illustrates eight days of normalized consumption for a single customer.

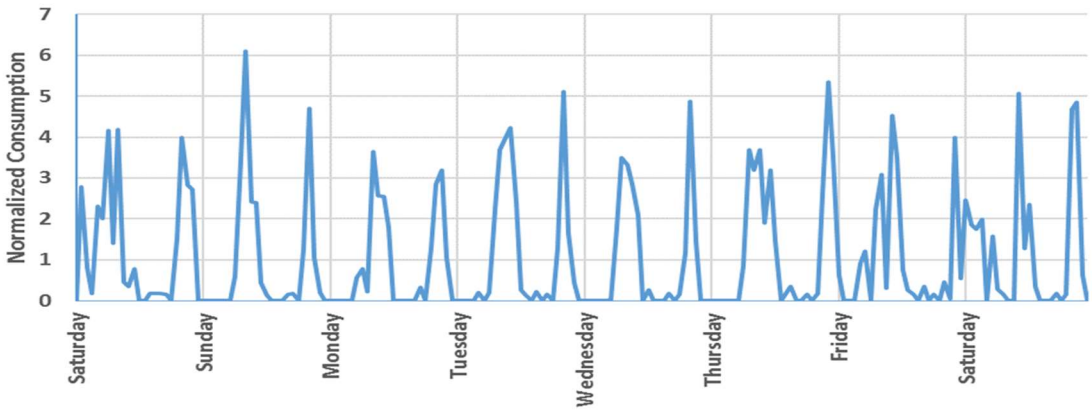

Figure 2.3 Non-zero mean normalized consumption for a residential water customer over eight days

2.3.3 Low-Pass Filter

Human behavior does not always adhere to strict time schedules. One example of this is a family that wakes between 0645 and 0715 to get ready for the day. The shower activity may occur during the 0600 hour, during the 0700 hour, or split between the two hours. Additionally, the water meter endpoint radios individually maintain local clocks, which may drift with age or temperature. To mitigate this behavioral time jitter and systematic drift, a low pass filter is applied to the hourly data.

As water consumption behaviors rarely exceed an hour duration, simple average of three consecutive hours is not as appropriate as a triangular filter of the same three consecutive hours [48]. This 3-point triangular filter accommodates consumption patterns that may not align exactly with the temporal hourly records. The sum of these weights is constrained to unity to avoid misrepresenting the normalized flow volume. The filter assigns weights of ¼, ½, and ¼ to each point,

$$z_{t-1} = \frac{(x_{t-2})}{4} + \frac{(x_{t-1})}{2} + \frac{(x_t)}{4} \quad . \tag{2.5}$$

The result is a smoothed time series representative of the underlying behavioral patterns throughout the day and week. Figure 2.4 illustrates the effects of the low-pass filtering compared to the original normalized data. The low-frequency daily behavior trends are maintained, while reducing the high-frequency noise related to the specific hour on the clock during the recorded behavior.

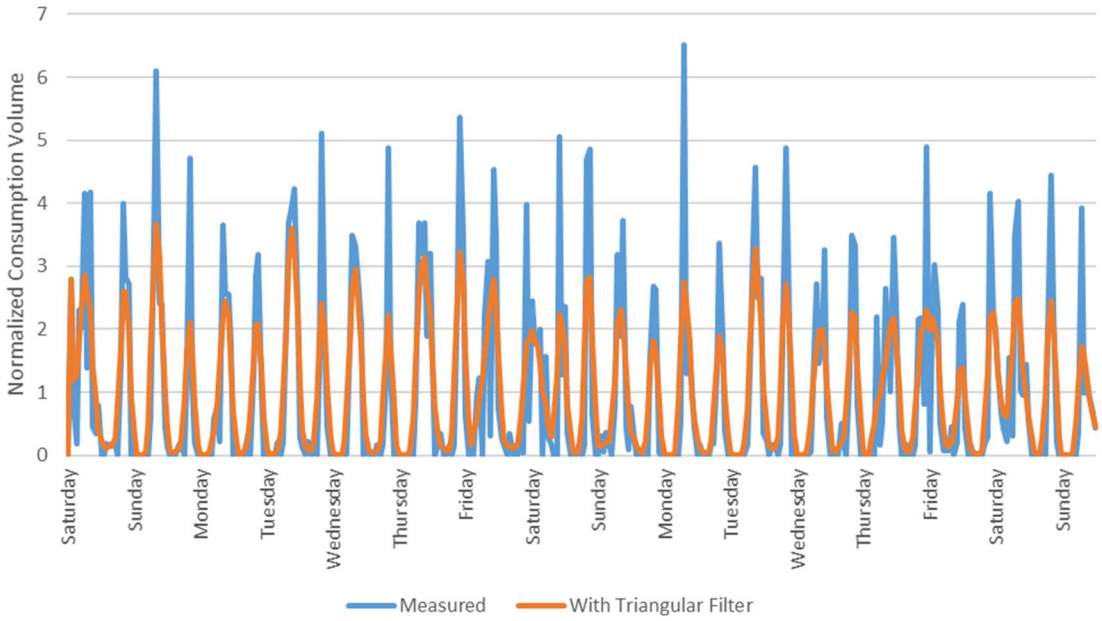

Figure 2.4 Non-zero median normalized consumption as measured and with triangular low-pass filtering

After the preprocessing steps of cleaning, normalizing, and filtering are complete, the individual data records are prepared further for the clustering process through dimensional reduction.

2.4 Dimensional Reduction

The time series in any individual customer record contains thousands of hourly flow measurements. Comparing these sets directly would be cumbersome and computationally intensive. Reducing the large set of individual measurements to a small set of model parameters for each customer makes the comparison more manageable. This section describes our two-step process of dimensional reduction using a reconstructed phase space (RPS) and creating Gaussian mixture models (GMM) of the data within the space. The resulting customer models are directly compared to produce groups of customers, as described in the next chapter.

2.4.1 Reconstructed Phase Space

The evaluation of a time series allows the discovery of an internal structure or trend within the data. Earlier in this chapter, Equation (2.1) defined a time series as a set, $\mathbf{X}$. Each measurement $x_{t+\tau}$ is referenced to time t, with a measurement interval τ. Knowing the internal structure allows prediction of future values, anomaly detection, understanding of the underlying nature of the system, and identification of particular components with the most information. Within this work, residential water consumption habits are the primary influence on the time series measurements.

Consider the weekday water consumption habits of a particular family of four – two adults who work outside the home between the hours of 0900 and 1700 and two children who attend school between September and June. Figure 2.5 shows the weekday consumption records of such a family for 159 weekdays, aligned from 0000 to 2400. Two periods of frequent consumption activity are visible – a morning period before and an afternoon/evening period at the end of the school or workday. The morning behavior is more consistent consumption due to shower activity, which the California Single Family Water Use study found to be an average of 18.2 gallons per shower [49]. The time of departure for work tends to be constant for the majority

of shift work employees, supported by evidence from residential water customer weekday patterns.

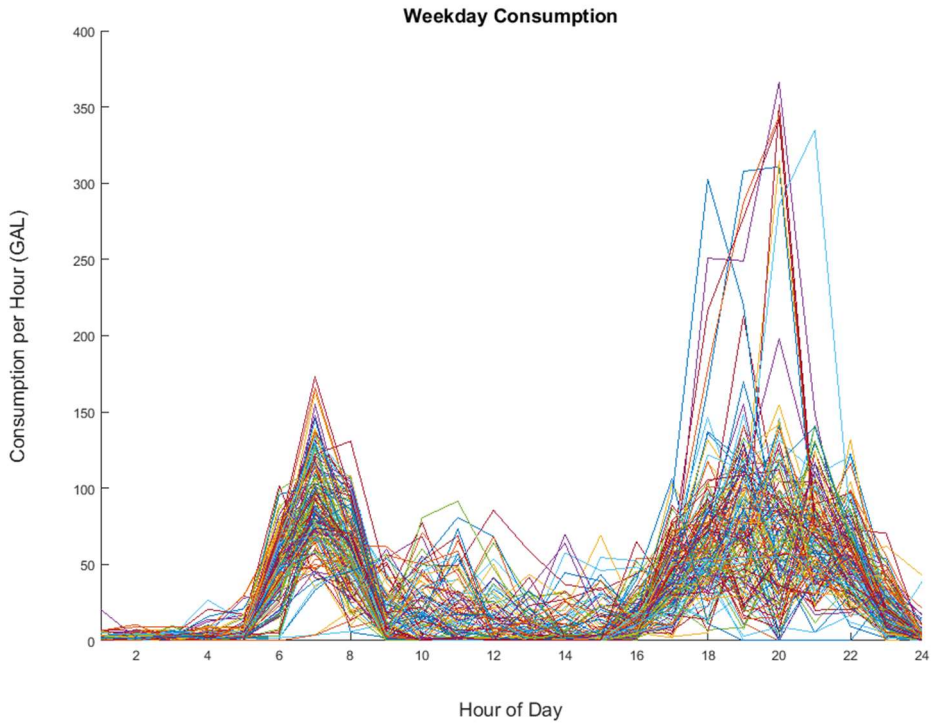

Figure 2.5 Household water consumption for 159 weekdays

If the individual and household behavior were perfectly predictable—never varying in volume or time, one could forecast the exact consumption given only the time of day. In reality, the system varies – hitting snooze on the alarm, waking early to run an errand, returning home during lunch to pick up a package, etc. Instead of creating a specific value associated with a specific time, the consumption pattern can be modeled by a series of states such as "sleep," "morning," "mid-day," "evening," and "irrigation," shown in Figure 2.6. Transitions between these states, the orange arrows in Figure 2.6, happen in a pattern within a probability distribution. While the exact value cannot be guaranteed, the system is governed by the sequence of the underlying pattern known as a *trajectory*. Knowledge of the trajectory enables the estimation of all future states, given the current state. This trajectory is called an *attractor* by Takens [50].

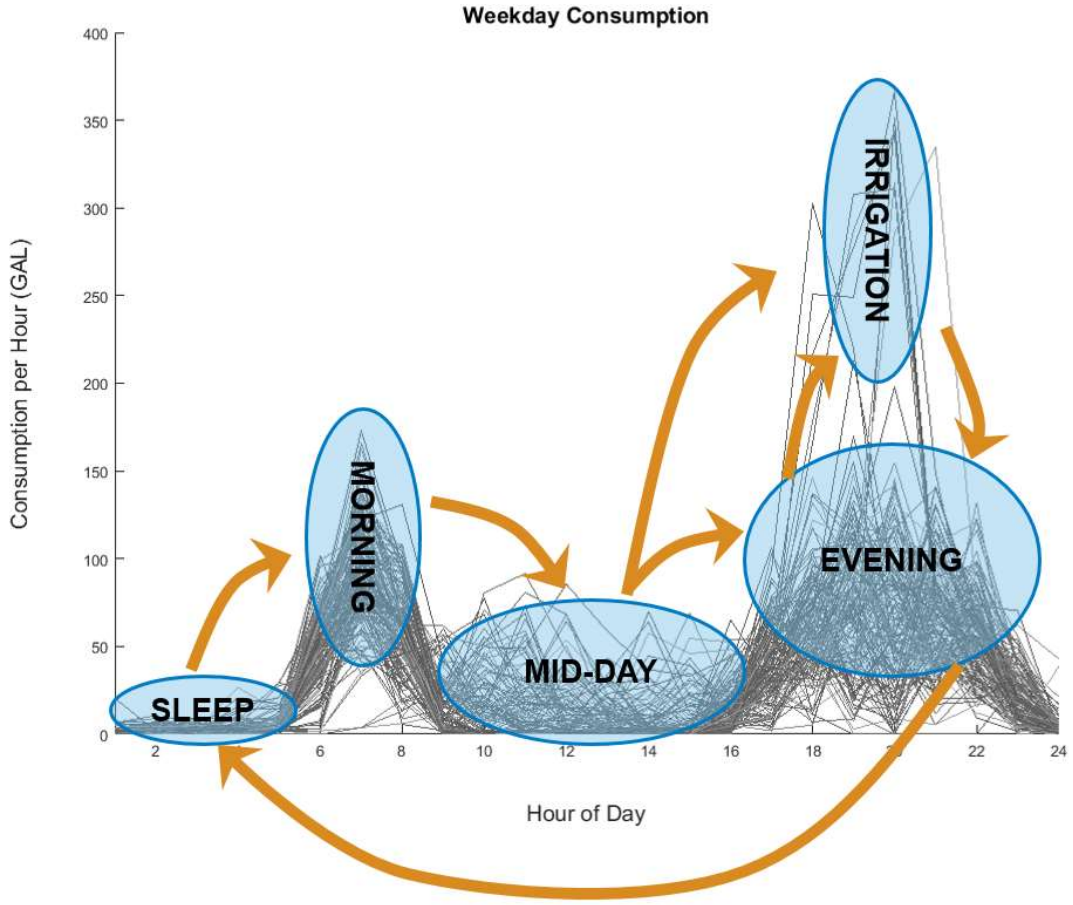

Figure 2.6 States of a weekday consumption pattern for a single-family household

One way to study these systems is to cast the time series into a vector space such that any location within the space identifies the system state at that moment [51]. Phase space embedding [51], [50] is an established method to represent a system in a vector space chosen to illustrate the dynamics of the original system most clearly. Figure 2.7 illustrates the embedding of a few data points as an example. Two groups of repeated behaviors are shown, red dots indicate behaviors occurring on a 24-hour schedule, and blue dots indicate behaviors occurring on a weekly, 168-hour, schedule. The embedding process, indicated by the colored arrows, shows how groups form within the vector space with axes corresponding to 0-, 24-, and 168-hour time lags.

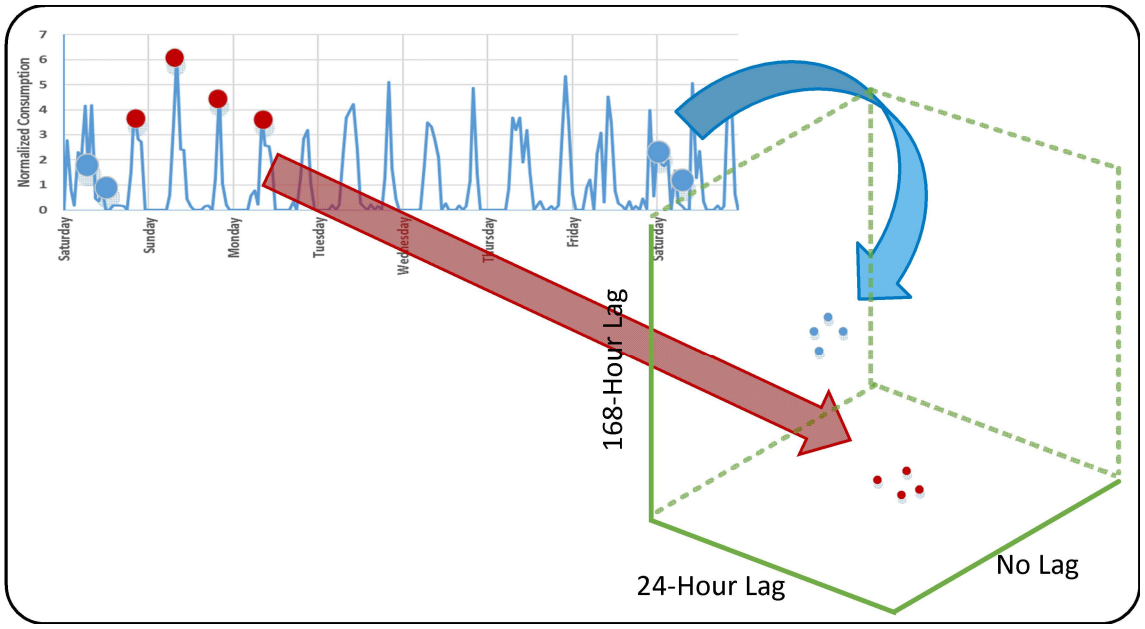

Figure 2.7 Embedding from a time series into a vector space forces groups of points to form based on repetitive behaviors corresponding to the time lag.

When the time series is embedded into the phase space, a single point is defined as a vector $\mathbf{Y}$ of points from the original series $\mathbf{X}$ each separated by time lags T_m to produce the dimensions within the space. The subscript m indicates the particular dimension associated with that lag T_m:

$$\mathbf{Y} = \left[x_{t+\mathrm{T}_1}, x_{t+\mathrm{T}_2}, \ldots, x_{t+\mathrm{T}_m} \right]. \tag{2.6}$$

These vectors are plotted in the newly defined phase space as a *topological embedding* of the original system [50].

Kantz and Schreiber [51] describe methods to guide proper choice of the embedding dimension m, time delay T_m, and a thorough explanation of reconstructed phase space embedding. The embedding dimension m must be chosen such that duplication resulting from

the system does not affect the performance of predictions, and algorithms can easily navigate the phase space.

Kantz and Schreiber [51] discuss the concept of false nearest neighbors (FNN) as one technique to determine a suitable lower bound for m .

$$FNN = \frac{\sum_{\text{all points}} \text{distance between nearest neighbors in dimension } (m_1)}{\sum_{\text{all points}} \text{distance between nearest neighbors in dimension } (m_1+1)} \tag{2.7}$$

When data is projected into a phase space, the sequential points trace a path. In a proper embedding, points that are close to one another within the phase space are also close to one another in trajectory and future states. That is, points representing mid-day activity should always follow two points in the phase space representing morning activity. If the phase space has been folded upon itself, two points may be false neighbors, representing two entirely different trajectories. Computing FNN requires identifying the nearest neighbor to every point within an embedding of dimension m_1 . A proper embedding will have little effect on nearest neighbor pairs if the dimension is increased to $m_1 + 1$, because the added dimension is redundant to the system.

Finding appropriate time lag T_m is dependent upon the internal workings of the system. Kantz and Schreiber [51] offer two suggestions for estimating T_m . One suggestion is to start with T_m equal to one-quarter the period of a periodic component within the signal. The second suggestion involves plotting different values of T_m to watch the data unfold and to choose a lag that has neither collapsed upon itself in a single direction nor is too complex. For the household in question, Figure 2.8 shows different embeddings with dimensions of zero- and various-hour time lags. The comparison shows some combinations have more structure of repeated patterns like the top row center (2-hours lag) and bottom row right (24-hours lag); and others have collapsed into a diagonal, such as the top left (1-hour lag). The top right (4-hours lag) and bottom left (6-hours

lag) show a few paths deviating from the primary trajectories along the two axes. These may be helpful to identify specific anomalous behaviors.

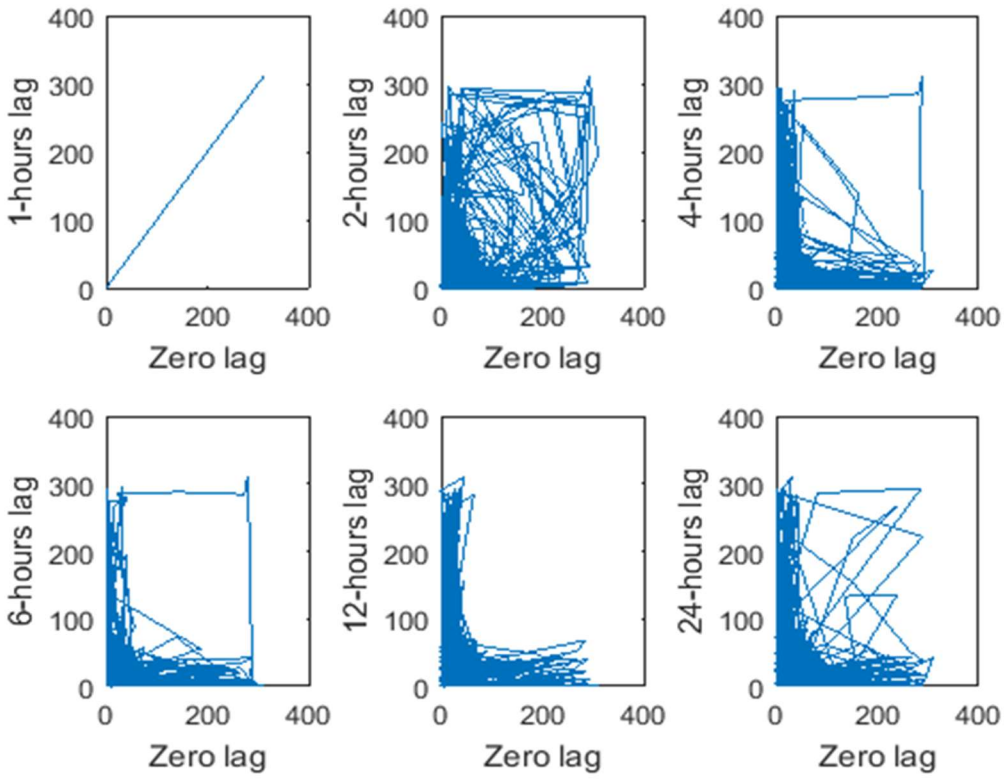

Figure 2.8 Time-delay embeddings of residential customer at different delays

This work focuses upon an embedding with dimensions of 0-, 24-, and 168-hour lags. The result of the embedding is a plot with groups of data forming at coordinates corresponding to repeating daily or weekly (24- or 168-hour) behaviors. Figure 2.9 illustrates the data from one customer as plotted on the phase space with 0-, 24-, and 168-hour lags.

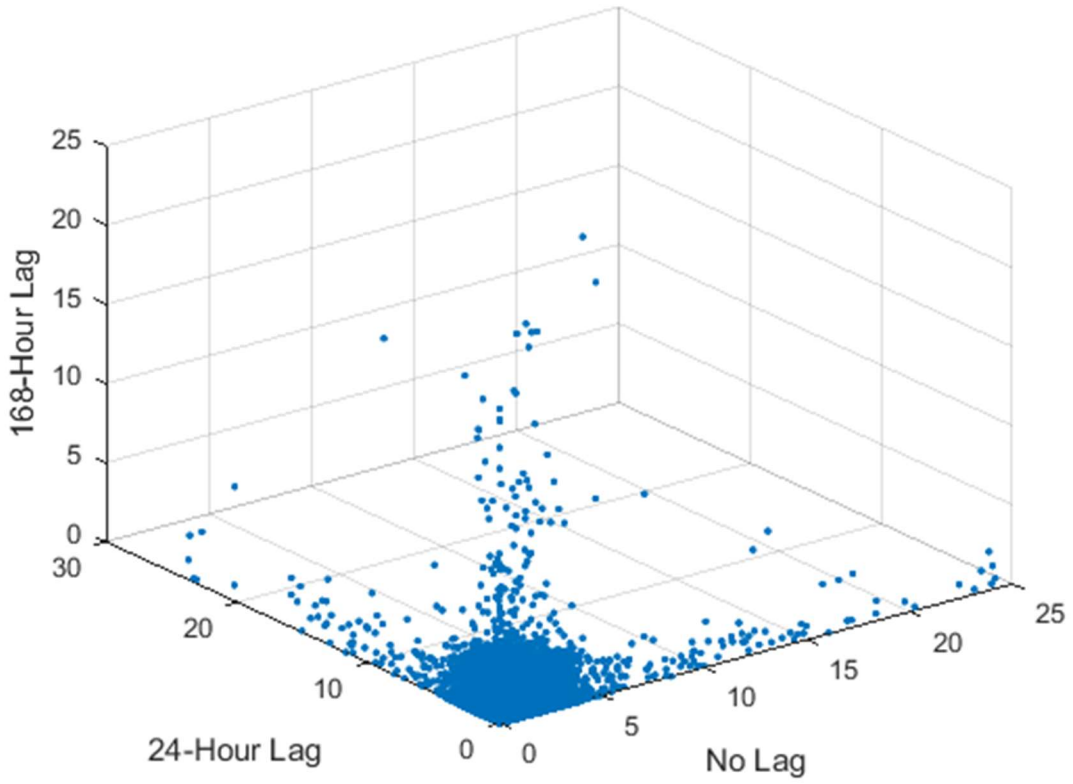

Figure 2.9 Normalized, smoothed, flow measurements for a residential customer embedded in phase space with 0-, 24-, and 168-hour lags

Figure 2.10 illustrates behavioral groupings within the RPS for a single customer. Colors indicate different clusters, with the centroid marked as a black **x**. For simplicity, this plot is generated by clustering with average Euclidean distance, but any clustering method could be implemented. The number of clusters is purposely chosen as a large value to identify as many groups as possible within the RPS, and no effort has been taken to optimize the number of clusters. The corresponding time series in Figure 2.11 has been colored using the zero-lag term of the RPS data point, allowing the comparison of time series points to their RPS location based on matching colors between the plots.

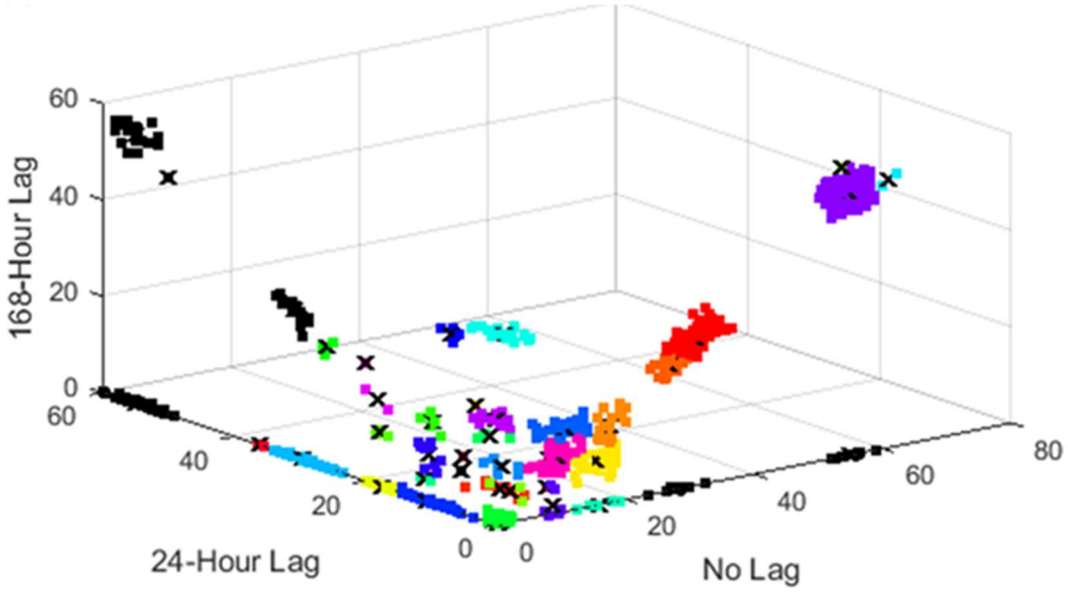

Figure 2.10 Reconstructed phase space with customer data colored by clusters using average Euclidean distance measure

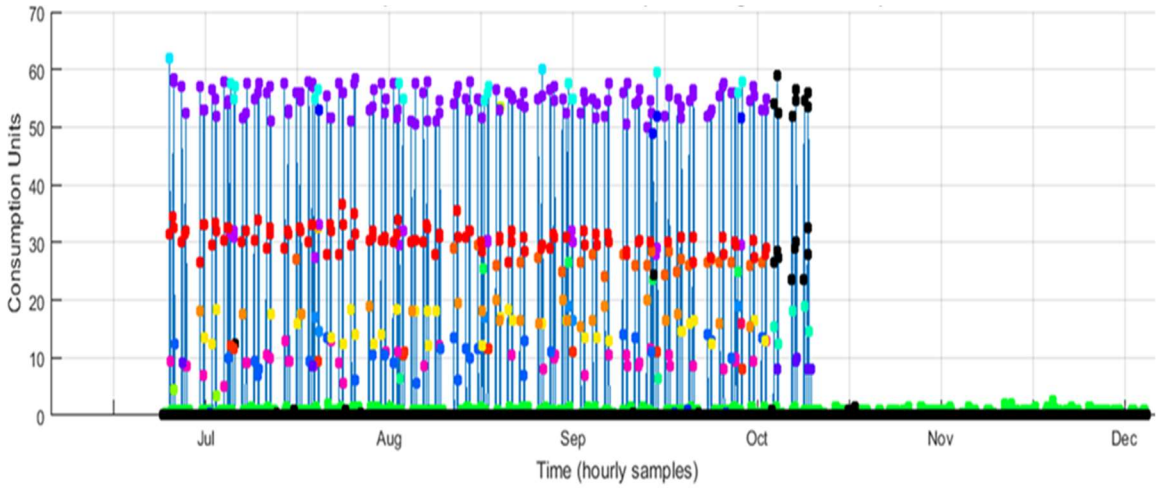

Figure 2.11 Time series data for customer in Figure 2.10, colors correspond to the zero-lag term from the RPS

This particular meter location appears to be largely unused after early October, when the previously routine usage patterns disappear, and only minimal volume is recorded. Easily identified behavior patterns include the high consumption volume purple data showing a strong

weekly component, visible as a cluster in the upper right portion of the RPS in Figure 2.10, and moderate consumption red and orange data also appear as a cluster in the RPS.

These phase space illustrations within this chapter are generated with the aid of the publicly available RPS Toolbox functions for embedding a time series into a reconstructed phase space using MATLAB® [52]. After embedding, dimensional reduction occurs by representing the embedded data by a Gaussian mixture model.

2.4.2 Gaussian Mixture Models

The technique of Gaussian mixture models (GMM) exploits the ability to represent a complex, unknown, system as a combination of known systems; specifically, normal Gaussian distributions [53]. Other work has shown the validity of representing time series data as probability distributions [54]. The GMM also is used to reduce dimensionality of high-dimension data sets into a model representative of the original data. For example, three years of data can be represented as a set of four or five Gaussians in the reconstructed phase space, each with mean and standard deviation in all phase space dimensions [55], [56], and a magnitude to bound the expected value of the range. The GMM representation requires $3dk$ terms, with k-component Gaussians and d dimensions. Thus, describing a five-component GMM in three-dimensional phase space requires 45 terms, regardless of the number of original data points. Equation (2.8) defines a k-component GMM of dimension d as the model $\hat{\mathbf{X}}$, estimating the time series $\mathbf{X}$. Each component $\mathcal{N}_d\left(\boldsymbol{\mu},\boldsymbol{\Sigma}\right)$ is a multivariate normal distribution,

$$\hat{\mathbf{X}} = \sum_{i=1}^{k}\left(\mathcal{N}_d\left(\boldsymbol{\mu},\boldsymbol{\Sigma}\right)\right)_i \ . \tag{2.8}$$

This section explains the creation of a GMM using a simple example in 1-dimensional space and then expands the GMM into the phase space discussed in the previous section. The 24 hourly measured flow recordings for two Mondays from a residential customer are shown as thin

lines of green and orange in Figure 2.12. This example is typical of many households with bimodal early morning and evening flow patterns. The figure also shows this data represented as a two-component GMM, the heavy blue line. In this example, the period between hours 0700 and 0900 has the highest expected value of water usage per hour, indicated by the tall peak in the GMM at that time. A lower expected value exists between hours 1800 and 2100. Early morning, before hour 0500, indicates a very low expected value of water usage.

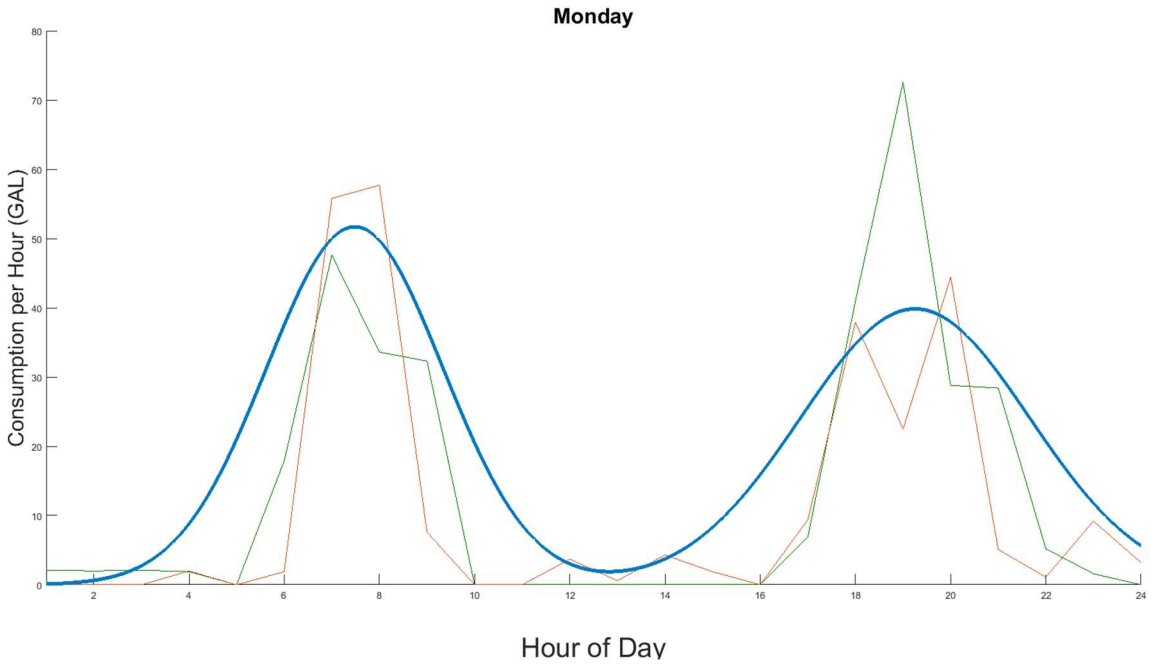

Figure 2.12 Hourly flow recordings for a residential customer with two Monday time series and a Gaussian mixture model representing the time series

Each component of the GMM is represented by a value indicating the central tendency along the time domain; in this example, this central point is the mean time of the component within the GMM. In addition, the component requires a measurement of the width, or dispersion of data along this same axis, in this case it is the standard deviation of the normal distribution. Finally, this one-dimensional example requires a magnitude of the component to represent the maximum expected value if the modeled range has not been normalized to one.

A more accurate probabilistic model is formed by taking a large number of patterns and fitting the combination to a GMM. The improved accuracy arises from the GMM representing enough data to be confident that the resulting model is a probabilistic portrayal of the underlying long-term behavior. Figure 2.13 shows 137 measured flow records for Mondays and the GMM superimposed on the same axis to illustrate the distinct bimodal pattern of behavior often associated with persons who work first shift jobs. This illustrates the very consistent morning consumption patterns between hour 0500 and 0900 in the morning and the much less consistent afternoon/evening consumption pattern and the GMM component width and magnitude reflect the more reliable expected values.

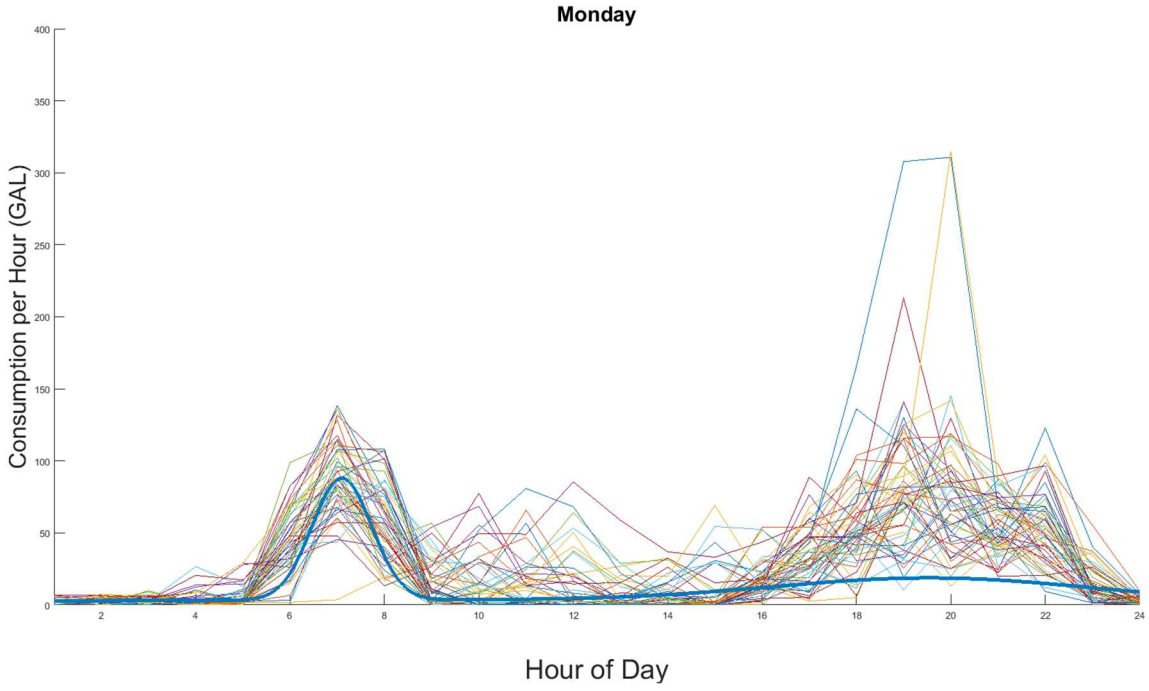

Figure 2.13 Gaussian mixture model and Monday flow records for 137 weeks of data, one residential customer

In contrast to the Monday patterns illustrated in the previous figures, the same plots for Sunday, Monday, and Tuesday can be seen in Figure 2.14, emphasizing the significant change in

water consumption behaviors between weekdays and weekends. The plots in Figure 2.14 also have been created with 137 weeks of data.

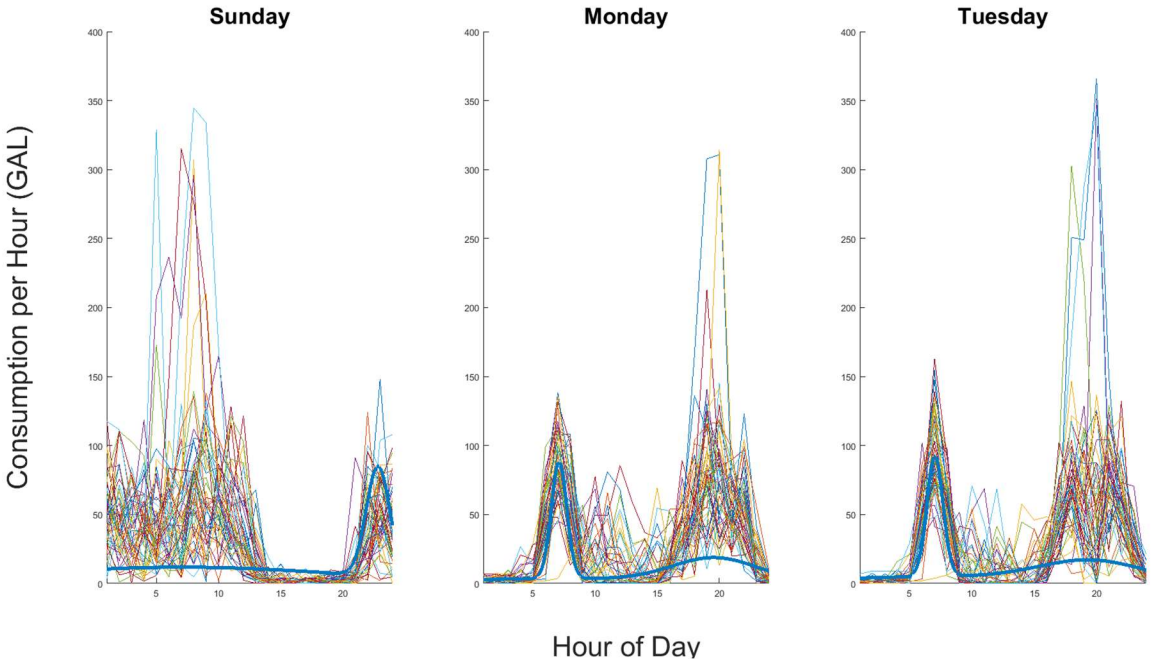

Figure 2.14 Gaussian mixture models with Sunday, Monday, and Tuesday flow records for one residential customer

These examples have all used one dimension for simplicity, but a GMM may be created in any number of dimensions. A three-dimensional GMM can model the data from the previous section as embedded within the phase space with 0-, 24-, and 168-hour time lags as axes. Previous research has been performed using GMM representations with and without embedding in reconstructed phase spaces. Combinations of Gaussian mixture models fit to data in an RPS have been studied by Povinelli et al. [57]. Their work applies the GMM within RPS to electric motor currents, to electrocardiogram recordings, and to the TIMIT speech corpus. Model tuning for the work in [57] is limited to the number of mixtures within the data, and the results indicate acceptable performance for signal classification. McKenna et al. use GMMs without phase space

embedding to represent hourly water consumption data and instead use k-means clustering to classify different groups based on these daily flow patterns [58].

The left portion of Figure 2.15 shows data from a single customer's meter records embedded into the phase space using the techniques presented in Section 2.4.1. The data are colored by the associated component of the GMM as assigned by the MATLAB® function fitgmdist, available within the statistics toolbox of MATLAB® [59]. The right portion of Figure 2.15 shows wireframe ellipsoids representing the components of the GMM. The wireframes are drawn using the MATLAB® function plot_gaussian_ellipsoid, available at MATLAB® Central File Exchange [60] using the vectors of mean and variance generated by the GMM creation of [59]. These ellipsoids are located at the corresponding component centroid and illustrate the component shapes within the space, the colors match the components on the left image. The wireframes indicate regions of expected values for water consumption behaviors within this phase space for this customer. A small region volume indicates a dense population of data falls into this component while a large volume indicates a more dispersed population within the component. The magnitude component of the GMM is related to the location within the phase space of these regions.

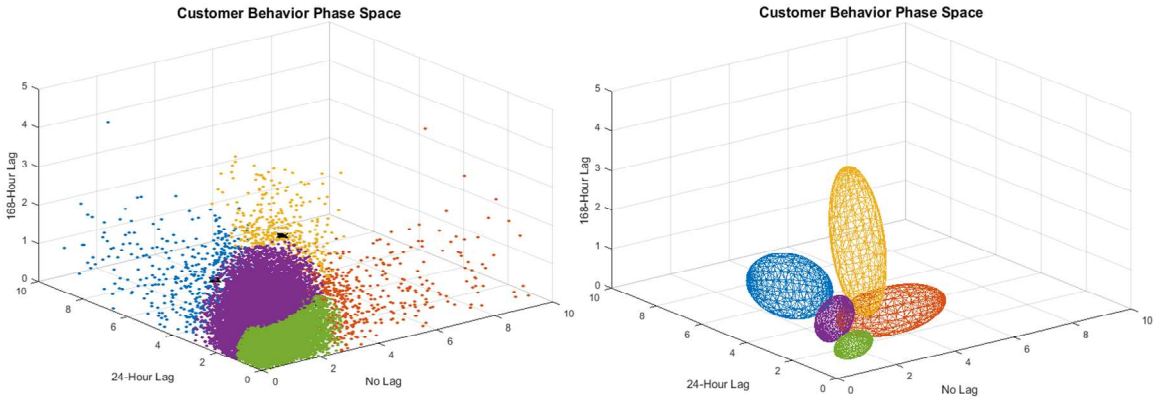

Figure 2.15 Data from a single customer embedded in the phase space (left) and represented by a Gaussian mixture model of five components (right)

These models are an imperfect representation of the original data, but trade the large memory footprint of the complete dataset for a much smaller footprint of model parameters. Clustering within the population of all customer models is possible upon completion of these preprocessing and dimensional reduction steps. The next chapter will address several clustering techniques and distance measures developed by others and offer an in-depth discussion of the specific methods used to compare the models within this work.

3 HIERARCHICIAL CLUSTERING AND DISTANCE MEASURES

The clustering process forms groups of similarly behaved customers after the models are created from clean, normalized, and filtered data. For reference, Figure 3.1 outlines the entire process of methods and experiments used in this research. This chapter focuses upon the clustering of the data and introduces commonly applied methods for clustering time series data. Then, the chosen hierarchical agglomerative clustering technique is explained in detail. After the clustering technique is described, a section is devoted to discussing various distance measures used in clustering research and presenting arguments for choosing the variation of information in this work. Finally, this chapter discusses the use of a dendrogram to display results, and further discusses interpretation of dendrograms found in the experiment results shown in the next chapter.

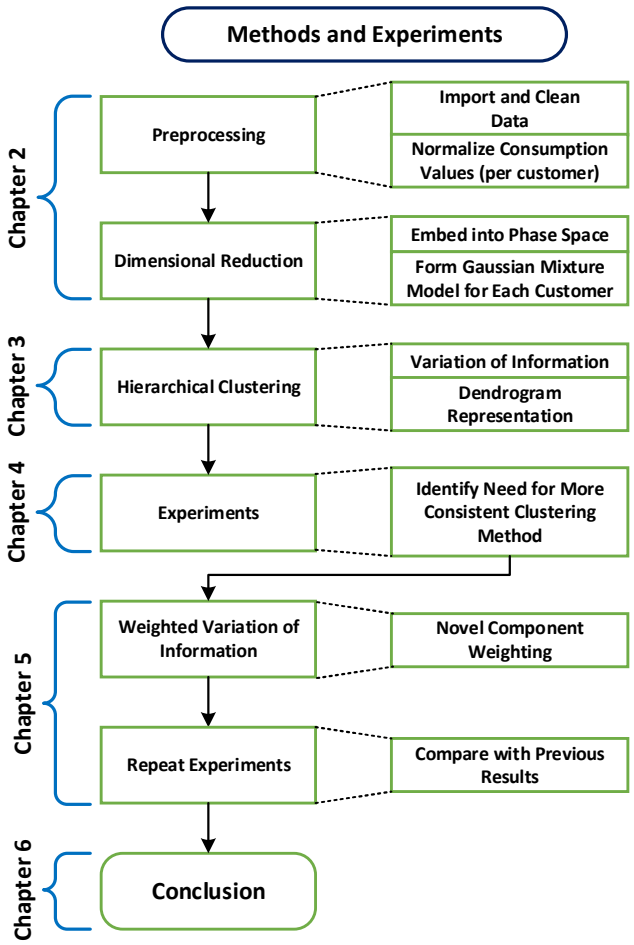

Figure 3.1 Flow diagram of the methods and experiments in this research.

3.1 Techniques for Clustering Time-Series Data

As noted previously, time-series data clustering is not a unique, or even new, concern. Dozens of industries rely upon time-series data and classification thereof. While some datasets for research have labels, a large quantity of applications for studying time-series data involve unsupervised learning. Many real-world sensor networks, including the Badger Meter, Inc. BEACON® Advanced Metering Analytics (AMA) system, do not have expert labeling for every data point. This requires the application of an *unsupervised* clustering algorithm—clustering without the feedback expert labels can provide.

The goal of unsupervised clustering is to assign groups or partitions within a data set based on some measure of similarity. Individual elements are assigned a value with respect to other individuals using a distance measure to determine the magnitude of difference between them. Specific examples of distance measures are discussed later within this chapter. Upon completion, the clustering algorithm results are evaluated as 'good' or 'poor' based on some criteria relative to the original problem or structure of the clustering results, as described in the next chapter. Distance measures, clustering algorithms, and criterion functions are heavily problem- and domain-dependent. A clustering algorithm that performs well for classifying music genre may perform poorly when attempting to find fraudulent credit card transactions.

Other research explores clustering of energy customers using smart meter data. Panapakidis et al. [18], [19] implement clustering of electric smart meter data. As opposed creating models such as our research, their work clusters the daily typical load profiles within a particular customer's data set. Representatives of those clusters are used to complete the second stage clustering across the population of all customers. Their work illustrates the complex problem of identifying the optimal number of clusters in a diverse data set. In contrast to the Panapakidis work, the clustering method presented in this work does not require a definition of an optimal number of clusters.

Bose and Chen track changing cluster populations over time using fuzzy c-means algorithms [61], [62]. Their work focuses upon migratory patterns of cellular phone customers, for the purposes of tracking dynamic market demands and customer retention. Their data exhibit not only customers who migrate from one cluster within the data to another, but also the formation of new clusters and dissolution of others as new behavior patterns emerge within the population.

A related problem arises in clustering music. Genre classification is not an identical problem, as the entirety of the work is available at time of classification. The whole song is

already produced and recorded, but similarity exists in the approach to first model the music, and then compare the model with others during the classification step. Logan and Salomon create models using the audio spectrum of the composition and then cluster multiple works using earth movers distance [23]. Jensen et al., create Gaussian mixture models from the Mel frequency coefficients within a work, then cluster those models based on three different distance measures – Kullback-Leibler distance, earth movers distance, and normalized least squares [63].

Another popular algorithm, spectral clustering, simplifies the problem by reducing the dimensionality in a different manner. First, the similarity matrix is constructed as a representation of the commonalities between every pair of data samples. Then, a graph Laplacian is computed from this similarity matrix. The clustering operates on eigenvectors from this graph Laplacian matrix and some predetermined clustering algorithm such as k-means or c-means. Spectral clustering algorithms vary on the specific details of constructing the graph Laplacian and the clustering step, but the same framework applies [16], [32], [64].

Statistical modeling of biological time-series has been applied to electrocardiogram data for classifying specific heart rhythms [57], [65]. In this work, Povinelli et al. cast the time-series signals into a reconstructed phase space and further apply Gaussian mixture models to represent the attractor within the reconstructed phase space. These models then classify a new time-series as a particular heart rhythm, aiding in medical diagnosis. The clustering method used to group water meter time series in this dissertation is similar to that of [56] and [64], and is discussed in detail in the next section of this chapter. We extend this method by clustering different customer models using the variation of information distance measure.

Some existing research classifies water usage based on metering data. Laspidou et al., use quarterly water billing information and self-organizing maps to cluster customers based on consumption [36]. Willis et al. [43] and Cardell-Oliver [44]–[46] investigate fixture-level consumption patterns to identify specific end uses of water in a location using high-resolution

metering. Other research focuses upon partitioning a utility's entire water distribution network into the optimal district metered areas (DMA) for processing groups of customers sourced by the same supply mains [58], [66]. Related research using smart meter flow data has produced outlier detection and forecasting algorithms [67]–[69] and leakage detection methods [70]. In contrast to this existing water utility research, our work focuses upon clustering similar customers based on temporal behavior patterns using only the hourly flow measurements recorded in the BEACON® AMA system.

3.2 Hierarchical Agglomerative Clustering

Hierarchical agglomerative clustering [26], [71] forms groups within the data by first assuming each sample is a unique group with only one data point. As the algorithm steps through each successive stage of clustering, more points are merged into ever-larger clusters. "Agglomerative" refers to the addition of new points to a cluster at each stage, in contrast to "divisive" clustering, which begins with all points in one group and divides the group into ever-smaller populations until the final stage where each sub-group has only one member. The hierarchy is the relationship between every data point and every sub-cluster of any size in the clustering process.

One of the biggest challenges to unsupervised clustering is choosing the correct number of groups within the population. Many papers present methods to compute the number of clusters as well as validity measures to prove the number of clusters chosen is correct [72]–[74]. Unlike several other clustering algorithms, hierarchical agglomerative clustering maintains the entire sequential process of joining. With this information, fixing the number of clusters before initiating the clustering process becomes less important. Eliminating the need to define the number of clusters in advance comes at a computational price, with hierarchical clustering having

a complexity of $O\left(N^2\right)$, compared to the $O\left(Nkd\right)$ complexity of k-means clustering with k clusters and dimension d [26], [71].

An example of the hierarchical agglomerative clustering process is presented here to illustrate the individual steps within the entire procedure. Suppose a group of data has been cast into the feature space shown in Figure 3.2.

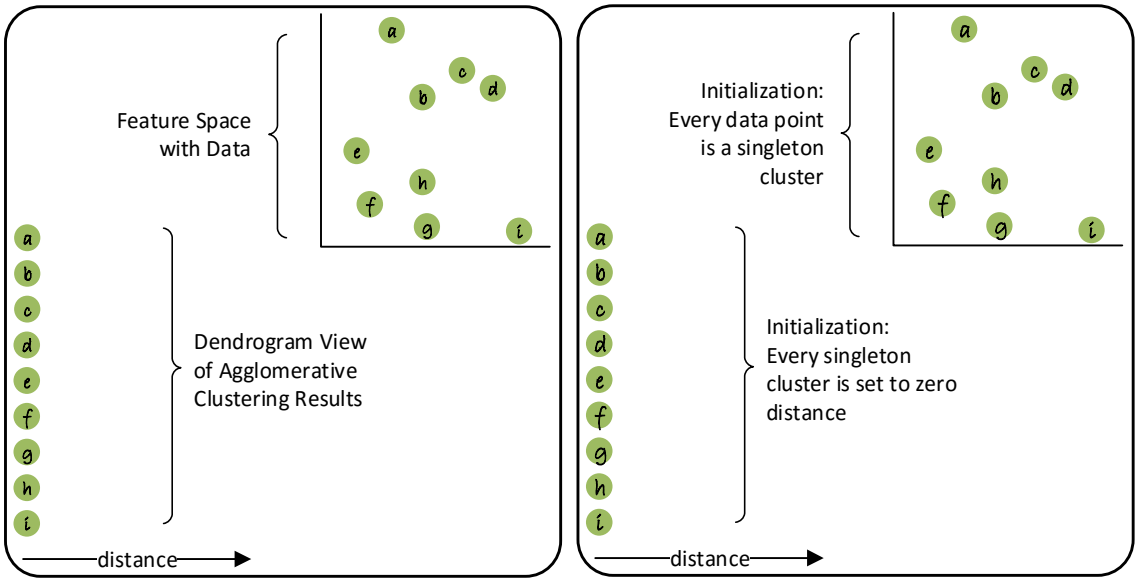

Figure 3.2 Agglomerative clustering example

The axes of the feature space in the upper right of this example can be any dimensions, using any distance measure to determine the "near" and "far." For this example, consider the distance measure to be a geometric measurement between the closest edges of any two clusters. Specific distance measures used in clustering problems are described in detail later, within the next section of this chapter. As the clustering progress is discussed in stages, the feature space in the upper right will indicate new clusters formed at each step. Likewise, the dendrogram view in the lower left reflects the distance between clusters at the time of join based on a consistent distance measure, and new links within the dendrogram are drawn at each step.

Initially, every point in the feature space is a singleton cluster with distance zero, shown in Figure 3.2. During the first merge step in the agglomerative hierarchical clustering process, the two clusters with the smallest distance between the closest edges are identified and combined to form a new cluster. The dendrogram shows a connector joining the two clusters with a horizontal length proportional to the distance between the clusters within the feature space, as defined by the distance measure used for clustering. This process is illustrated in Figure 3.3.

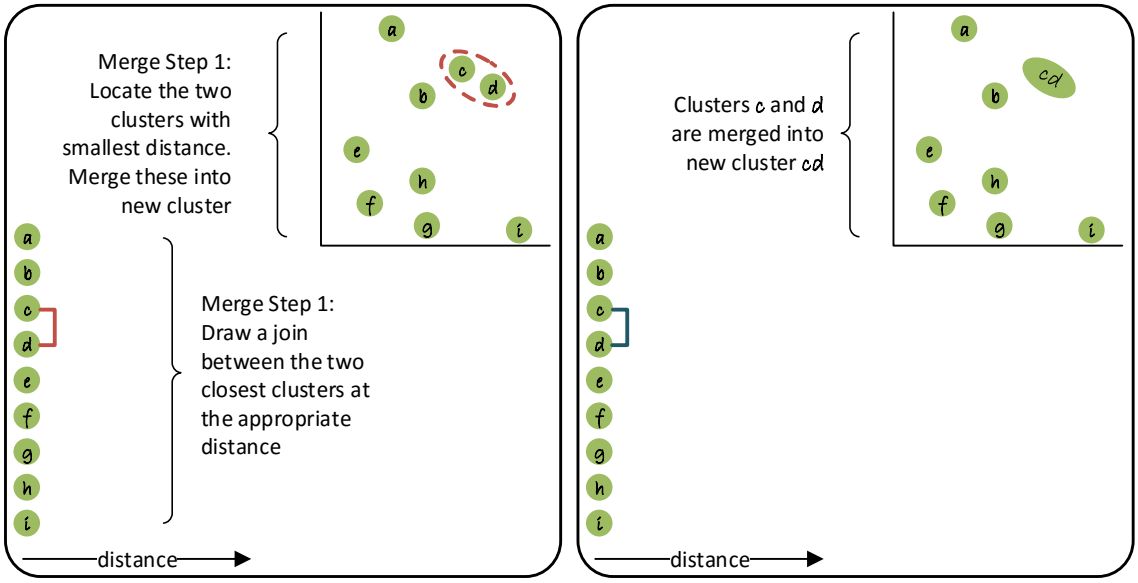

Figure 3.3 Merge step 1 of the hierarchical agglomerative clustering process

The merge process iterates to join the next two nearest clusters at each step, considering both the remaining singleton clusters and the clusters formed by previous joins. This continues for each new step, as in Figure 3.4. The new clusters continue to merge and expand, enveloping the members as they are joined.

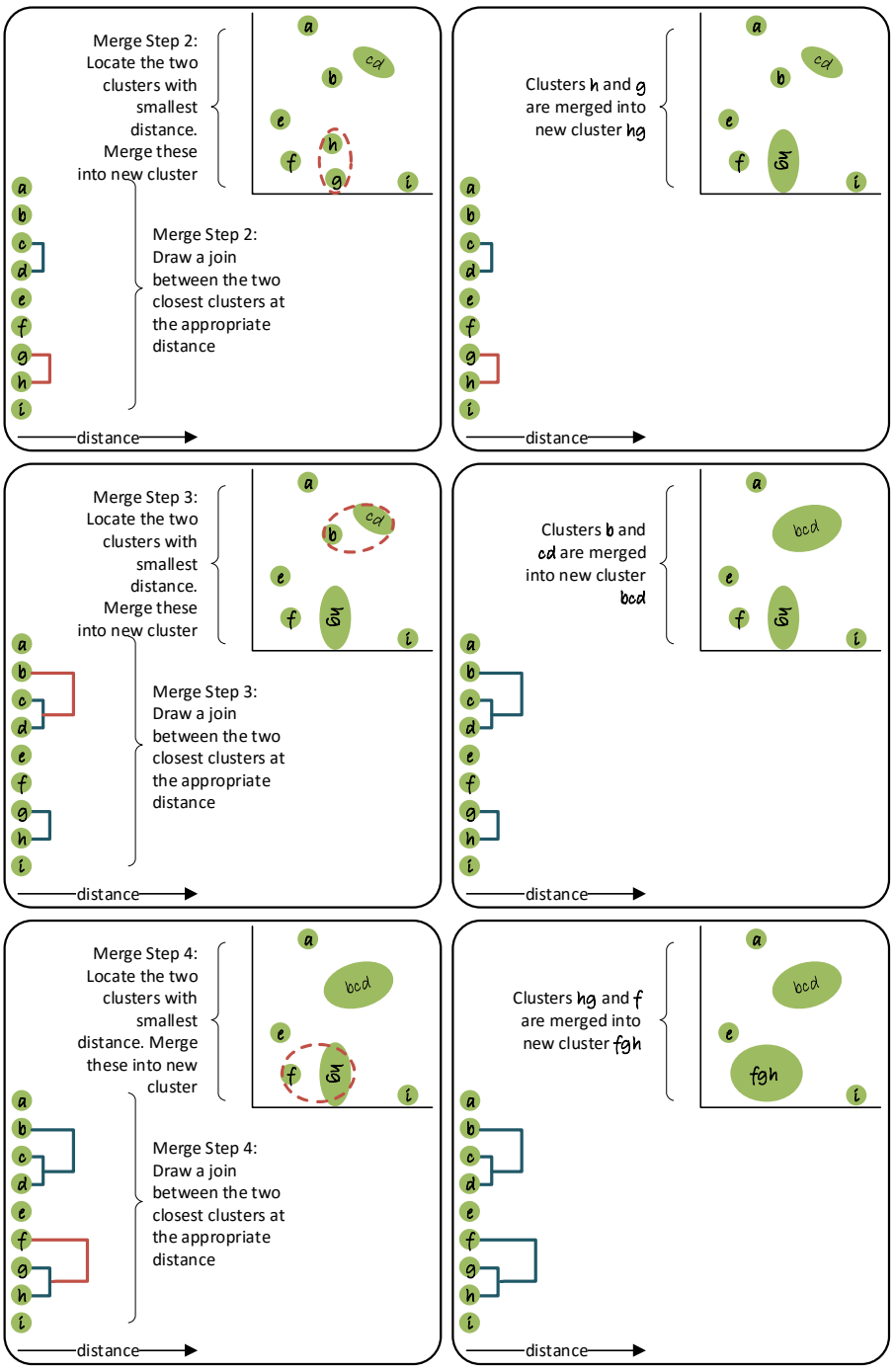

Figure 3.4 Three steps of the hierarchical agglomerative clustering process

At the final step, only one cluster exists, containing all the original data points. The final dendrogram in Figure 3.5 illustrates the entire merge process and maintains all the distance

information of the clustering process. This visual representation is stored as a table of merge steps with linkage distances and cluster populations for simple manipulation.

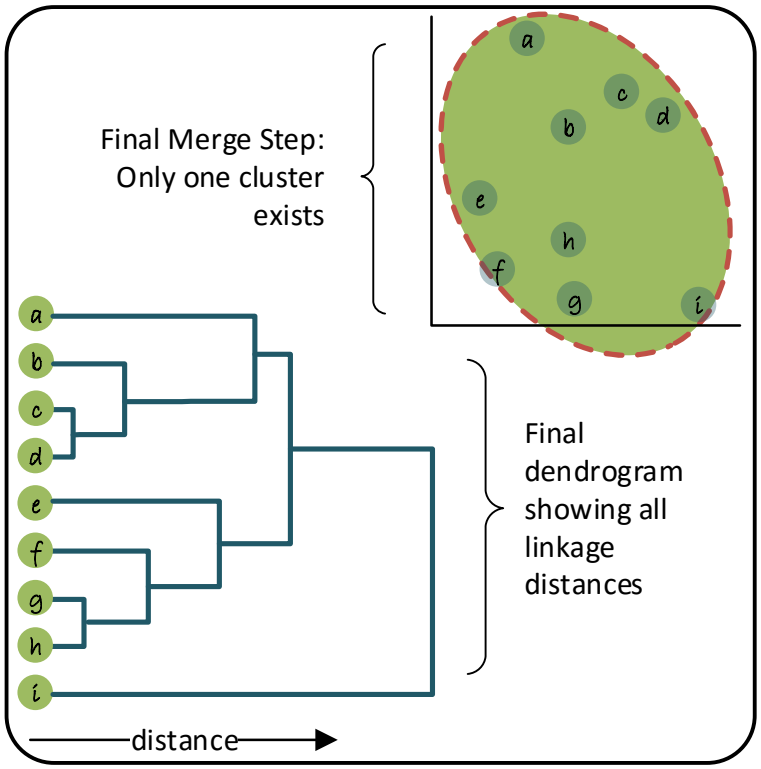

Figure 3.5 Final merge step of hierarchical clustering and the final dendrogram linkage

After the final merge step, all clustering information is stored within the dendrogram and the associated linkage table. This allows the user to decide on the number of clusters desired *post hoc*. In many commercial applications, the number of clusters is affected by external requirements having nothing to do with the mathematically optimal number of clusters within the data set. For water utilities, the number of clusters desired may be influenced by limitations of labor, physical resources, or financial resources available. The hierarchical clustering allows the utility to apply these limited resources efficiently and effectively. Alternatively, a more rigorous statistical method may be used to determine the optimal number of clusters within the data if no external constraints apply [75], [76].

Regardless of the method chosen to determine the number of clusters, the dendrogram is cut to form the clusters as shown in Figure 3.6. Visually, a vertical line is drawn on the dendrogram (shown in red dashes), intersecting the horizontal lines representing the different clusters. Since the vertical line intersects three horizontal lines, the data is divided into three clusters. Figure 3.6 illustrates three clusters: {abcd}, {efgh}, and {i}. The upper right corner of the illustration identifies the clusters in the feature space.

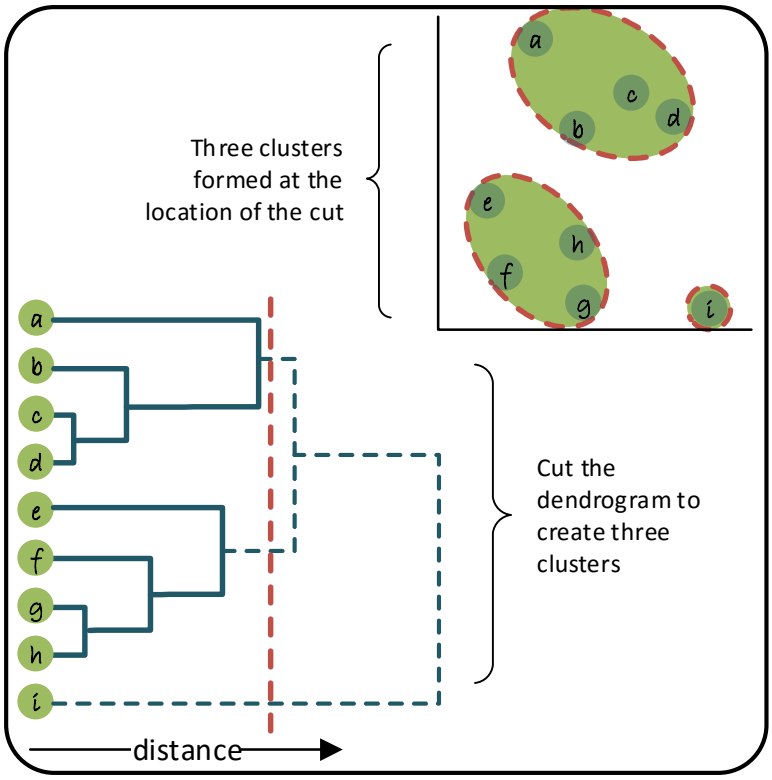

Figure 3.6 Cutting the dendrogram to form clusters

In MATLAB®, the recorded historical clustering sequence is called a linkage. The linkage contains every pair of sub-clusters joined at a particular step along with the separation distance between those two sub-clusters at time of join. After clustering is complete, the population can be divided into any number of clusters from the stored linkage data. The user can choose the desired number clusters at time of use and adapt the number of clusters to the business

case at hand. Such cases may include a high-level "normal vs. abnormal" grouping or breaking the population into several behavioral groups for targeted marketing and conservation campaigns.

The main benefits of the hierarchical agglomerative clustering technique relate to the flexibility of storing results in a linkage. In addition to not needing *a priori* knowledge of the number of clusters within the data, the entire linkage is stored, allowing the data to be clustered into any number of sub-groups, as the requirements or applications dictate. If a group of utility customers has been clustered in this manner, one application can determine four sub-groups for different conservation marketing campaigns with a simultaneous application of two larger groups as "normal" assigned no follow-up, and "outliers" requiring follow-up in-person site visits to investigate abnormalities. Further, as the utility's customer base fluctuates and changes, the clustering algorithm does not assume any fixed number of groups as the one correct answer and can change to reflect the dynamic underlying structure of the customers.

3.3 Distance Measures

As the previous section noted, hierarchical agglomerative clustering requires the definition of a distance measure to identify the two most similar sub-groups to join at each step of the clustering process. Several different distance measures have been studied including, geometric distances, population-based distances, probabilistic distances, and information-theoretic distances.

The terms "measure" and "metric" are not synonymous. From the mathematical sense, a metric must satisfy the so-called *metric axioms* [77], illustrated graphically in Figure 3.7. Consider a space of three clusterings,

$$\begin{aligned} A &= \{a_1, a_2, \ldots, a_m\} \\ B &= \{b_1, b_2, \ldots, b_n\} \\ C &= \{c_1, c_2, \ldots, c_o\}. \end{aligned} \tag{3.1}$$

For a distance measure ρ describing the distance between A, B, and C shown in Figure 3.7, to be considered a metric, it must

1. Be non-negative and equal zero if and only if the clusters are equal

$$\begin{aligned} &\rho(A,B) \geq 0 \\ &\rho(A,B) = 0 \quad \equiv \quad A = B; \end{aligned} \tag{3.2}$$

2. Be symmetric

$$\rho(A,B) = \rho(B,A); \text{ and} \tag{3.3}$$

3. Satisfy the triangle inequality

$$\rho(A,B) + \rho(B,C) \geq \rho(A,C). \tag{3.4}$$

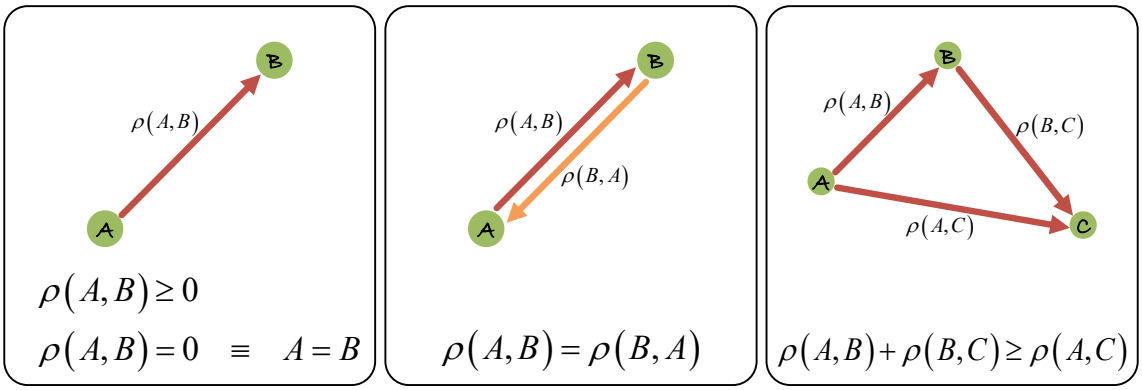

Figure 3.7 Illustration of the three properties required for a metric

3.3.1 Geometric Distance Measures

One commonly used geometric distance is the Minkowski distance, a generalization of the Euclidean distance metric. Hashem and Humaid [78], Chicco [79], and Rao and Cook [80] all present utility consumption classifying solutions using the Euclidean or Minkowski distance. The Minkowski distance L_p of order p is

$$A = \{a_1, a_2, \ldots, a_n\}$$
$$B = \{b_1, b_2, \ldots, b_n\} \tag{3.5}$$
$$L_p(A,B) = \left(\sum_{i=1}^{n} |a_i - b_i|^p \right)^{1/p}.$$

The Minkowski distance satisfies the triangle inequality for values of $p \geq 1$, making the Minkowski distance of orders $p \geq 1$ a true metric. As Figure 3.8 illustrates, Minkowski distance of order one, L_1, corresponds to Manhattan or taxicab distance, and order two, L_2, corresponds to the Euclidean distance.

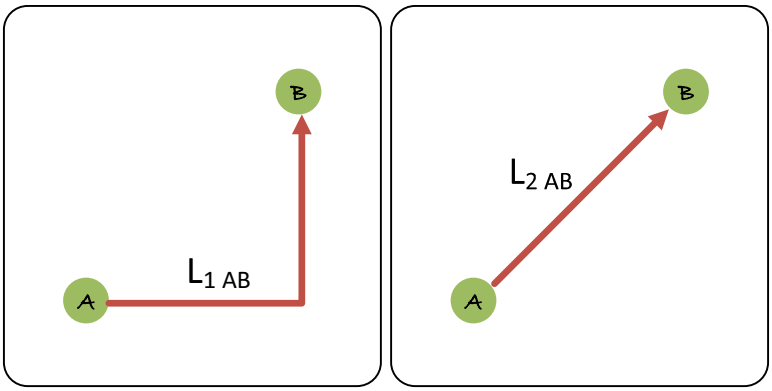

Figure 3.8 Minkowski distances of order 1 and 2 between clusters A and B

However, as [77], and [78] illustrate, within high dimensional data, the elements or subgroups within a set are separated by large distances, and the Euclidean/Minkowski distances become less useful as a clustering measure. Other distance metrics can be used, such as the cosine distance (vector dot product). The cosine distance uses the angular distance between vectors [83], [84] to determine the similarity of two elements, eliminating some of the problems associated with Euclidean distance. If $\mathbf{a} \cdot \mathbf{b}$ is the dot product, and $\|\mathbf{a}\|$ is a vector norm, the cosine distance is

$$\begin{aligned}\mathbf{a} &= (a_1, a_2, \ldots, a_n) \\ \mathbf{b} &= (b_1, b_2, \ldots, b_n) \\ \cos\theta_{\mathbf{ab}} &= \frac{\sum_{i=1}^{n} a_i \bullet b_i}{\|\mathbf{a}\|\|\mathbf{b}\|} \ .\end{aligned} \tag{3.6}$$

3.3.2 Earth-Mover's Distance

The earth-mover's distance (EMD) computes a minimum cost required to transform a given probability distribution into a different distribution [63], [85]. This is compared to the labor cost of moving a pile of earth from one shape and location to a different shape and location as shown in Figure 3.9. Computing the EMD begins by finding the total flow,

$$F = \left[f_{ij} \right] , \tag{3.7}$$

between two clusterings,

$$\begin{aligned}A &= \left\{ \left(a_1, w_{a_1}\right), \left(a_2, w_{a_2}\right), \ldots, \left(a_m, w_{a_m}\right) \right\} \text{ and} \\ B &= \left\{ \left(b_1, w_{b_1}\right), \left(b_2, w_{b_2}\right), \ldots, \left(b_n, w_{b_n}\right) \right\} \ ,\end{aligned} \tag{3.8}$$

composed of individual clusters a_i and b_j with associated weights w_{ai} and w_{bj}, and a distance d_{ij} defined between clusters a_i and b_j. The flow F minimizes the overall cost of work

$$W_{ABF} = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij} \ , \tag{3.9}$$

subject to constraints:

1. allow flow in one direction only (from A to B, not in reverse):

$$f_{ij} \geq 0 \quad 1 \leq i \leq m \quad 1 \leq j \leq n \ ; \tag{3.10}$$

2. limit the quantity exported to the weight of the source cluster:

$$\sum_{j=1}^{n} f_{ij} \le w_{a_i} \quad 1 \le i \le m \ ; \tag{3.11}$$

3. limit the quantity received to the weight of the destination cluster:

$$\sum_{i=1}^{m} f_{ij} \le w_{b_j} \quad 1 \le j \le n \ ; \text{ and} \tag{3.12}$$

4. require the movement of the most weight possible:

$$F_{AB} = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{a_i}, \sum_{j=1}^{n} w_{b_j} \right) . \tag{3.13}$$

Upon solving the optimization under these constraints for F, the earth mover's distance normalizes W by F,

$$EMD\left(A, B\right) = \frac{W_{ABF}}{F_{AB}} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}. \tag{3.14}$$

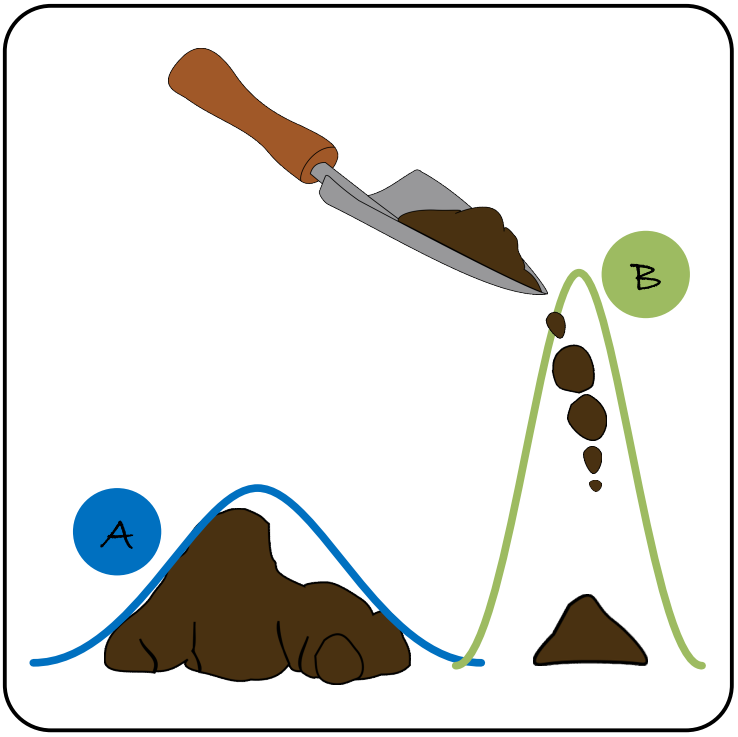

Figure 3.9 Earth mover's distance is based on the cost associated with work effort when transforming one distribution into another, as if the distributions were soil being moved by a shovel.

3.3.3 Population-Based Distance Measures

Population-based distances rely upon the members of each cluster under consideration to determine the similarity. The adjusted Rand index (ARI) compares two possible resulting clusterings and counts the pairs of members agreed or disagreed upon between the two, while correcting for change [86].

Given clusterings A and B, the contingency table is drawn as follows, where every entry n_{ij} corresponds to the number of agreements in both clusterings.

$$A = \{a_1, a_2, \ldots, a_i\}$$
$$B = \{b_1, b_2, \ldots, b_j\} \tag{3.15}$$

class\cluster	b_1	b_2	$\cdots$	b_j	sums
a_1	n_{11}	n_{12}	$\cdots$	n_{1j}	$n_{1\cdot}$
a_2	n_{21}	n_{22}	$\cdots$	n_{2j}	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_i	n_{i1}	n_{i2}	$\cdots$	n_{ij}	$n_{i\cdot}$
sums	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot j}$	$n_{\cdot\cdot} = n$

The ARI is

$$ARI(A,B) = \frac{\sum_{ij}\binom{n_{ij}}{2} - \frac{\left[\sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}\right]}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right] - \frac{\left[\sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}\right]}{\binom{n}{2}}}. \tag{3.16}$$

An ARI value of one indicates exact agreement between the clusterings, and a value of zero results from randomly assigning clusters.

The Fowlkes-Mallows index (FMI) is a population-based measure similar to the Rand index based on the agreements and disagreements of pairs within two clusterings. The FMI compares entire hierarchical clustering trees at each value of k clusters and allows us to compare within a tree to determine if additional clusters add benefit to the overall clustering [87]. Given clusterings A and B, the contingency table entries n_{ij} correspond to the number of agreements. The total population has n elements, and the row and column sums are annotated by $n_{i\cdot}$ and $n_{\cdot j}$, respectively.

$$A = \{a_1, a_2, \ldots, a_i\}$$
$$B = \{b_1, b_2, \ldots, b_j\} \tag{3.17}$$

class\cluster	b_1	b_2	$\cdots$	b_j	sums
a_1	n_{11}	n_{12}	$\cdots$	n_{1j}	$n_{1\cdot}$
a_2	n_{21}	n_{22}	$\cdots$	n_{2j}	$n_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_i	n_{i1}	n_{i2}	$\cdots$	n_{ij}	$n_{i\cdot}$
sums	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot j}$	$n_{\cdot\cdot} = n$

The FMI is

$$FMI(A,B) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij}^2 - n}{\sqrt{\left(\sum_{i=1}^{k} n_{i\cdot}^2 - n\right)\left(\sum_{j=1}^{k} n_{\cdot j}^2 - n\right)}}. \tag{3.18}$$

3.3.4 Information-Theoretic Distance Measures

Information-theoretic measures such as Kullback-Liebler divergence, Mutual information, and variation of information [77] are also used for clustering. The Kullback-Liebler (KL) divergence, or relative entropy, is related to the penalty of mistakenly describing one probabilistic model with an erroneous model. The KL divergence is asymmetric, and a symmetric version can be computed by determining the average of the two directional KL divergences between two models [25], [54], [88]. The KL divergence between two probabilistic models is

$$KL(a,b) = \sum a(x) \ln \frac{a(x)}{b(x)}\ , \tag{3.19}$$

where $a(x)$ and $b(x)$ have been created from the same random variable x. Care must be taken to select sufficient data when creating the models, as an event predicted impossible by one model, $b(x) = 0$, but possible by the other, $a(x) > 0$, results in $KL(a,b) = \infty$.

KL is not a true metric, as it does not satisfy the triangle inequality, nor is it symmetric:

$$KL(a,b) \neq KL(b,a). \tag{3.20}$$

A symmetric version of KL is

$$KL = KL(a,b) + KL(b,a) \ . \tag{3.21}$$

The KL divergence is the cost of assuming an incorrect probabilistic model. In information theory, this may be the cost of additional bits for storage or transmission of data when a non-optimal coding scheme has been applied. In the game of chance illustrated in Figure 3.10, the gambler places bets with the assumption of a pair of six-sided dice. However, the game is actually played with one four-sided die and one eight-sided die. This erroneous assumption changes the probabilities of several roll values, and the gambler loses money by not understanding these probabilities correctly before betting.

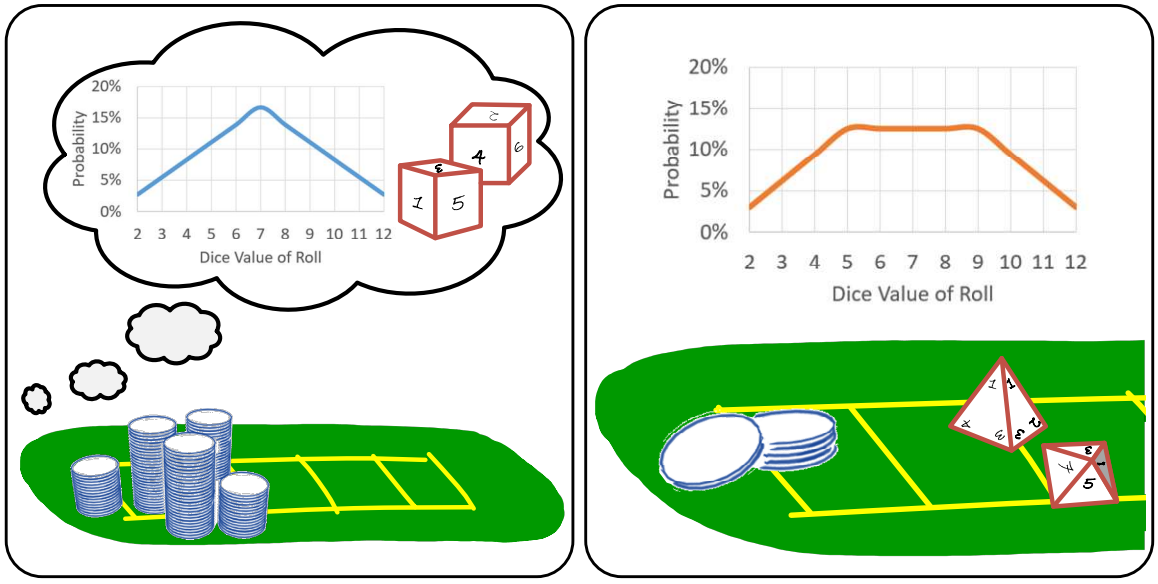

Figure 3.10 KL divergence is the cost of assuming the game is played with the probability distribution on the left, but the reality shows different dice are used. The dice affect the odds, and the gambler loses money.

Mutual information (MI) describes the amount of information known about one probabilistic model through knowledge of a second probabilistic model. MI is the information shared by the two models, defined as

$$\begin{aligned} A &= \{a_1, a_2, \ldots, a_i\} \\ B &= \{b_1, b_2, \ldots, b_j\} \\ MI(A,B) &= \sum_i \sum_j P(a_i, b_j) \log\left(\frac{P(a_i, b_j)}{P(a_i) P(b_j)} \right). \end{aligned} \tag{3.22}$$

The models A and B each contain one or more components, a_i and b_j. MI is illustrated in Figure 3.11 and is used to compute the variation of information [16], [82], [89].

Variation of information (VI) is a measurement of how much information is lost when combing two groups as opposed to keeping the groups separate. Unlike mutual information, VI is a true metric [77], satisfying the triangle inequality and also allowing comparison of clusters with different populations. We chose VI as a distance measure for this research because of these desirable properties.

The VI distance between two sets is the sum of unique information that would be lost if the two sets are combined. With the MI as defined by Equation (3.22) and individual entropy of each set

$$\begin{aligned} H(A) &= \sum_i P(a_i) \log_2 \left[P(a_i) \right], \text{ and} \\ H(B) &= \sum_j P(b_j) \log_2 \left[P(b_j) \right], \end{aligned} \tag{3.23}$$

the non-intersecting parts of all sets are collectively

$$VI(A,B) = H(A) + H(B) - 2\left[MI(A,B) \right]. \tag{3.24}$$

The relationships between the individual entropy of each set, the intersection of the two sets (MI), and the VI are clarified in a Venn diagram such as Figure 3.11.

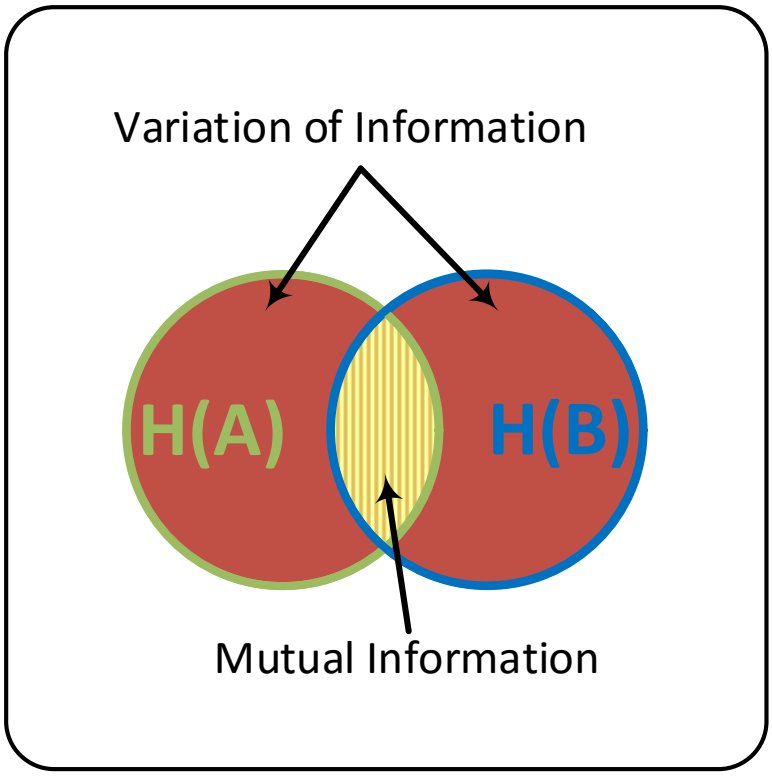

Figure 3.11 Venn diagram describing relationships between entropy, mutual information, and variation of information [77]

3.4 MATLAB® Implementation of Hierarchical Clustering for GMM with VI

The hierarchical clustering algorithm is programmed with MATLAB®. The clustering has three main operations. The first computes the variation of information distance between two models. Next, the join process merges the two models with the smallest distance. Finally, a linkage table stores the joins and a dendrogram is generated to display the results in a graphical manner. These three operations are described in detail here, including references to the significant MATLAB® functions used in each step.

3.4.1 Computing Variation of Information

Computing the distance between two subsets is essential to the hierarchical clustering algorithm. The VI distance metric describes the amount of information lost when two models are combined into a new model. In three dimensions, the VI corresponds to the non-overlapping volume of two convex hulls within the three dimensional space. If the two hulls are coincidental, the VI is small, and combining the two into a new cluster loses very little information. Conversely, if the two hulls are entirely separate, the VI is large and reflects the large loss of unique information if they are combined. Hulls that overlap partially or touch will fall somewhere in between these two extremes.

Several functions in MATLAB® and functions shared through MATLAB® Central File Exchange facilitate the comparison of convex hull volumes for estimating the VI between two models [59], [60], [90]–[92]. Recall a GMM of k component Gaussians in d dimensions is

$$\hat{\mathbf{X}} = \sum_{i=1}^{k} \left(\mathcal{N}_d \left(\boldsymbol{\mu}, \boldsymbol{\Sigma} \right) \right)_i \ . \tag{3.25}$$

A central location vector

$$\boldsymbol{\mu}_k = \left[\mu_1, \mu_2, \ldots, \mu_d \right] \tag{3.26}$$

describes each Gaussian component and a $d \times d$ covariance matrix,

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd} \end{bmatrix}. \tag{3.27}$$

An ellipsoid hull is computed to model each Gaussian mixture component for a particular customer, and a geometric tessellation of the hull is plotted within the phase space using the plot_gaussian_ellipsoid function [60]. The volume of this ellipsoid hull,

$$V_k = \left(\frac{4}{3}\pi\right)\prod_{j=1}^{d} r_j , \tag{3.28}$$

estimates the entropy of this component of the Gaussian Mixture Model, with r_j being the radius of the ellipsoid for any axis. The volume is an output of convhulln MATLAB® function [90], computed based on the Qhull method [91]. A summation of all GMM component hull volumes,

$$\hat{H} = \sum_{i=1}^{k}\left(\left(\frac{4}{3}\pi\right)\prod_{j=1}^{d} r_j\right)_i , \tag{3.29}$$

estimates the entropy of the customer model. If a customer has perfectly consistent water consumption behaviors, the associated GMM component volumes will be small. As variations in the consumption behavior or temporal patterns increase, the GMM component volumes will also increase.

Two models are compared to each other by computing the points of intersection of the GMM component hulls. Figure 3.12 shows a simple model with spheres, illustrating the intersection between the two hulls as a solid volume. When the surface of one hull is located within the enclosed volume of a second hull, an intersection is present. A boundary for the intersecting volume is created by first identifying the set of points on the surface of the large sphere that exist within the volume of the smaller sphere with the aid of the function inhull from MATLAB® Central File Exchange [92]. Then, we identify the reciprocal set of points on the surface of the small sphere that exist within the volume of the larger sphere. A new convex hull is created with the combined set of intersecting points using the convhulln MATLAB® function

[90]; and the volume of the intersecting hull is provided as an output of that function. Additional computations address the special cases when the hulls are coincidental, not intersecting, or if one is fully enveloped within the other. This intersecting volume approximates the mutual information (MI) between the two components.

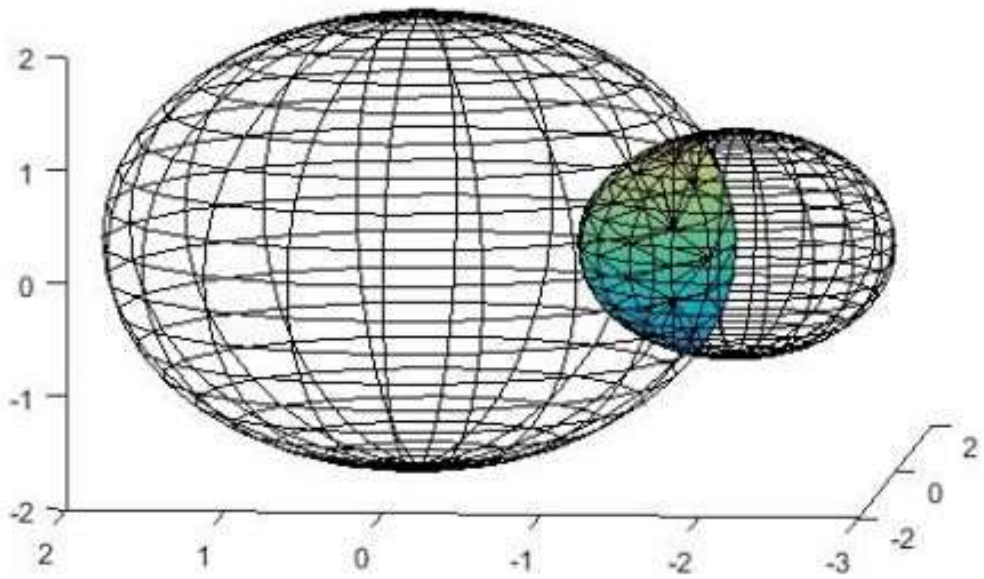

Figure 3.12 Visualizing mutual information (intersecting volume) of two customer models

Since the summation of all model component volumes enclosed within a hull estimates the entropy H of the particular customer model, the summation of all the intersecting (filled) volumes between two customer models is the estimated MI between those two customer models. The variation of information is the sum of all volumes from both models (A and B), subtracting double the volume of the MI

$$\hat{VI}\left(A,B\right)=\hat{H}\left(A\right)+\hat{H}\left(B\right)-2\left[\hat{MI}\left(A,B\right)\right]. \tag{3.30}$$

Gaussian mixture models with multiple components represent customers in this study, creating interactions that are more complex. Figure 3.13 illustrates two customers, each with several Gaussian components in the model. Each component is shown as a wireframe ellipsoid, and all components for one customer have the same color. The combined volume of all the hulls outlined in green estimates the entropy of the green customer. Likewise, the combination of all volumes in blue estimates the entropy of the blue customer. Wireframe ellipsoids illustrate each component of the GMM, with the intersection between the green hulls and blue hulls represented as a solid volume.

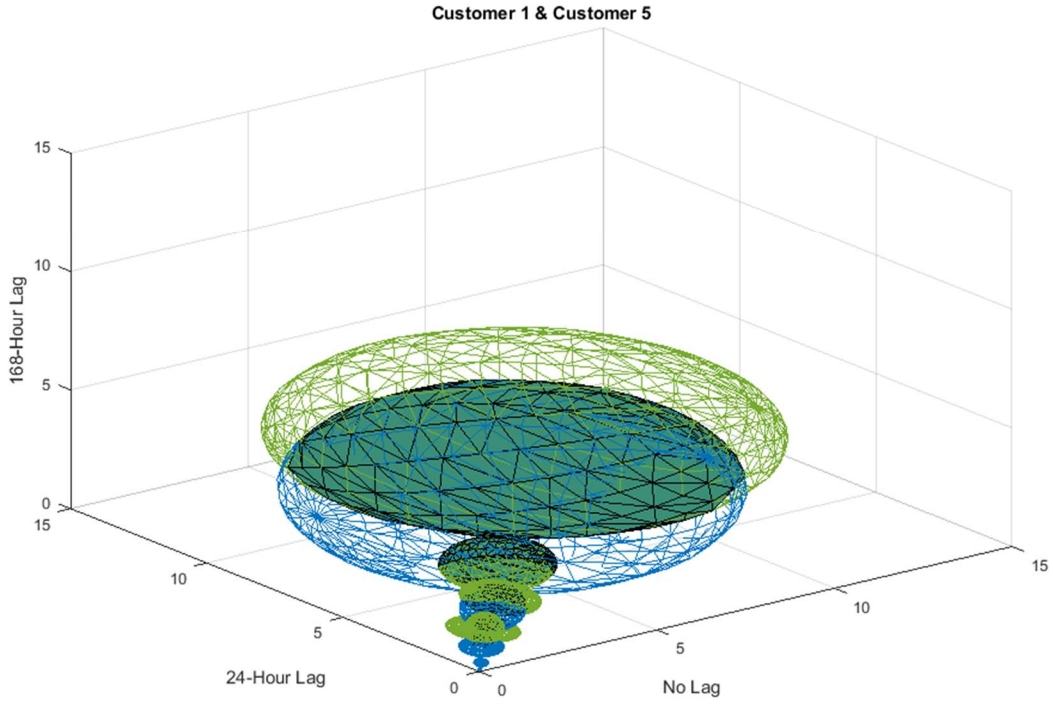

Figure 3.13 Four-component Gaussian mixture models and the mutual information volume for two customers

Intersections are computed between the entire group of green hulls and the entire group of blue hulls. Therefore, a single hull may intersect any number of hulls from the other customer, or none at all. Expanding Equation (3.30) yields estimated entropies

$$\hat{H}(A) = \sum_{i=1}^{n} \hat{H}(A_i), \text{ and} \tag{3.31}$$

$$\hat{H}(B) = \sum_{k=1}^{m} \hat{H}(B_k). \tag{3.32}$$

The estimated mutual information

$$\hat{MI}(A,B) = \sum_{k=1}^{m}\sum_{i=1}^{n} \hat{MI}(A_i, B_k), \tag{3.33}$$

and the estimated variation of information

$$\hat{VI}(A,B) = \sum_{i=1}^{n} \hat{H}(A_i) + \sum_{k=1}^{m} \hat{H}(B_k) - 2\left(\sum_{k=1}^{m}\sum_{i=1}^{n} \hat{MI}(A_i, B_k)\right). \tag{3.34}$$

In the example using four Gaussian components for each customer model, $n = m = 4$. Thus, the estimation of MI for two four-component models requires 16 computations, one for each pair of components.

Due to the random seeding of the models and imperfect alignment of coincidental hulls, a floor of zero is enforced for $\hat{VI}$. This eliminates negative distances between customers that may occur during the computation. The estimated VI distance is computed initially between every pair of customer models within the set, and then new computations are made as models are joined during the hierarchical clustering process.

3.4.2 Computing a New Hull when Two Models are Joined

At each step in the clustering process, the two models with the smallest $\hat{VI}$ are joined. The join process introduces more error into the method, but approaching it this way allows quicker computation of the resulting joined model.

The two models to be joined each already have a set of points defined at the surface – the points used to compute the geometric tessellation of the hull. Joining the models uses the combined set of points from both hulls to form the source for a new hull. Since each hull is already evenly distributed at one standard deviation around a geometric center, forming a new GMM with fitgmdist [59] from the combined set of surface points and then creating new hulls for each component of the GMM using convhulln [90] approximates a convex join between the two models in phase space. After the formation of a new GMM from the join, the $\hat{V}I$ distance must be computed between the newly joined model and the remaining models in the space. Subsequent steps of the clustering process ignores the previous individual components and only compares $\hat{V}I$ of the merged cluster. The recursive process continues, joining the next closest pair of clusters at each step until all customers have been combined into a large cluster.

At each step, the distance and cluster populations at time of merge are recorded in a linkage table to facilitate reconstruction of the hierarchical clusters. Hierarchical agglomerative clustering allows any number of clusters to be chosen by the user after the clustering process has completed. This eliminates the need to predict how many clusters exist within the data or to predetermine the clusters the data is forced into. The hierarchical clustering also provides flexibility for the utility to choose to increase or decrease the clusters desired based on the specific task.

3.4.3 Using a Dendrogram and Linkage Table to Display Results

While the linkage data table containing VI distance at time of the join and the population is sufficient for a database or software, humans require a more visually pleasing presentation. For this study, a dendrogram illustrates the cluster populations, merge operations, and the relative distances between any two clusters at time of merge. Figure 3.14 shows a simplified example of a population, as well as a dendrogram constructed from an agglomerative clustering of the population. Table 3-1 shows the equivalent linkage record stored to create the dendrogram. The

left column of Table 3-1 identifies the join occurring in a particular step of the hierarchical clustering process.

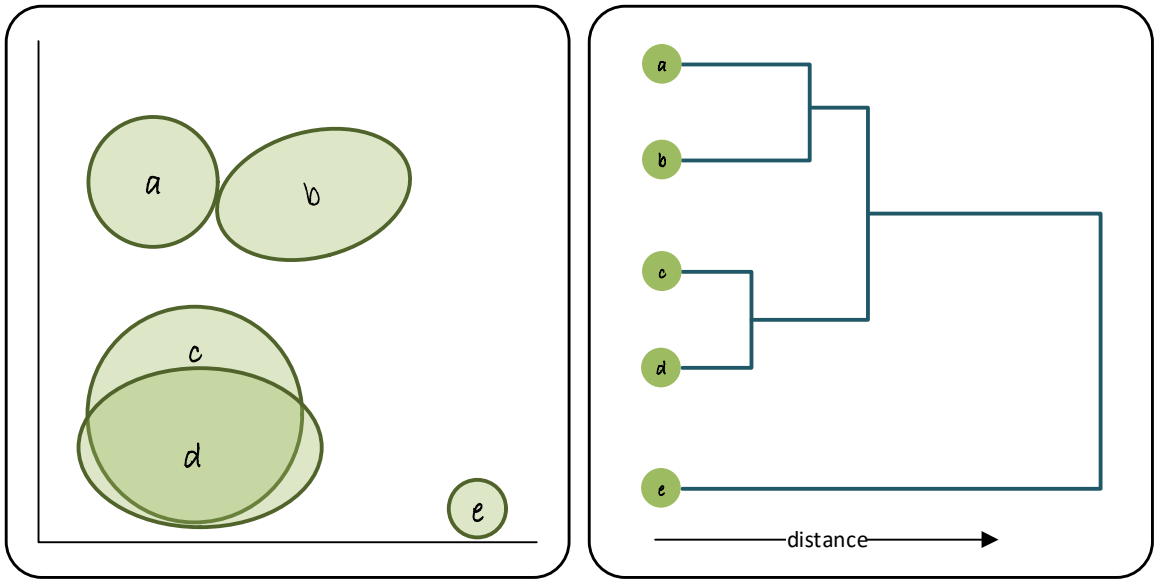

Figure 3.14 Sample population for hierarchical agglomerative clustering algorithm and corresponding dendrogram

Table 3-1 Sample linkage record for agglomerative hierarchical clustering

Clustering Step	First Cluster to be Joined	Second Cluster to be Joined	Distance	New Cluster
1	$\{C\}$	$\{D\}$	1	$\{C,D\}$
2	$\{A\}$	$\{B\}$	2	$\{A,B\}$
3	$\{A,B\}$	$\{C,D\}$	3	$\{A,B,C,D\}$
4	$\{A,B,C,D\}$	$\{E\}$	6	$\{A,B,C,D,E\}$

In Figure 3.14, five original clusters exist. Clusters A and B are close to each other, as are C and D. Cluster E is far from the others. The dendrogram representation of the clustering shows these relationships with the distance between two members of a new clustering on the horizontal

axis. Since C and D are overlapping, the VI distance between them is very small; joining these into a larger cluster would lose very little information. A and B are touching, but not overlapping as much as C and D, indicating a small VI distance, but still longer than the join between C and D. The cluster {A, B} is closer to {C, D}, with a distance further than either of the previous joins on the dendrogram. Finally, the join between {A, B, C, D} and {E} occurs to complete the agglomerative clustering.

The next section of this dissertation will introduce the first set of experiments. The experiments implement the methods presented in Chapter 2 and in this chapter on a set of 99 customers from a Midwestern utility. These experiments are designed to demonstrate the usefulness and the limitations of the hierarchical agglomerative clustering using variation of information as a distance measure between Gaussian mixture models.

4 EXPERIMENT CONFIGURATION AND RESULTS

Having explained the details of the clustering method, the distance measure for comparing two subsets prior to joining, and the format of the output from the clustering process, this chapter now presents the first set of experiments. For reference, Figure 4.1 outlines the entire process of methods and experiments used in this research. This chapter focuses upon experiments to support the chosen process as an appropriate method for grouping water utility customers. The motivation, procedure, results, and analysis for each experiment are presented together.

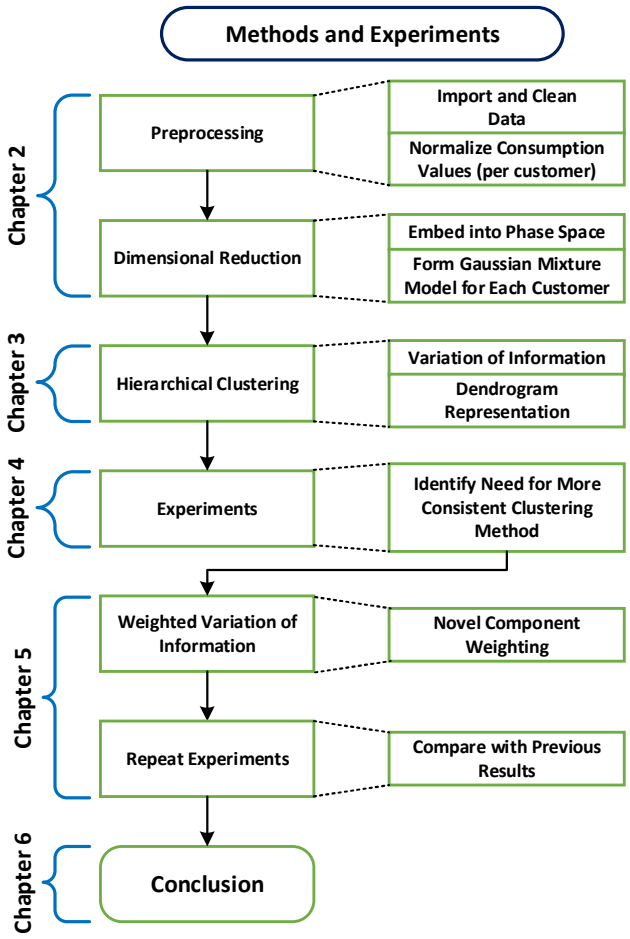

Figure 4.1 Flow diagram of the methods and experiments in this research.

4.1 Evaluation of the Methods

As mentioned in Section 3.1, unsupervised clustering operates without any data labels to confirm the results. Prior to releasing a software package to customers, the system is first tested for correct behavior, appropriate design, and consistent results. The proposed unsupervised clustering method has been evaluated using validation, verification, and consistency testing. While the two terms validation and verification are often confused with one another, they are not interchangeable and represent two distinct sets of testing. Adding to the confusion, the method used to determine if a clustering algorithm is performing well is commonly called "cluster validation."

1. Validation of software determines if the algorithm is designed correctly to meet the needs of the customer, or if a different method would have been a better choice.

2. Verification determines if the algorithm or method works as the programmer intended, or if defects and programming mistakes exist within the software, causing erroneous results.

3. Cluster validation determines if the clustering produces an appropriate separation of the data set for the defined problem.

4.1.1 Validation

Validation testing determines if the algorithm is designed correctly for the problem in question, or if a different method would have been a better choice. This is both an evaluation of the software as well as an assessment of the requirements and understanding of the problem domain.

Validation of unsupervised clustering techniques presents additional challenges. Cluster validation analyzes the results of the clustering to determine if the techniques applied produce clusters that are appropriate for the problem. The two methods of cluster validation measures are internal and external. External validation requires labeled data and compares the actual answers

with the performance of the algorithm. Limited external validation on this clustering method is performed using synthetic customer data generated to mimic customers who should be clustered similar to each other and different from each other, and comparing the output to those synthetic data labels. This technique of test data sets is widely accepted as a means of software validation [93]–[95].

Internal validation attempts to measure performance of the algorithm by evaluating the structure of the clusters at the output. Most internal validation measures, such as Dunn's index or the silhouette index, rely upon quantifying the compactness and separation of the clusters formed by the algorithm [96]. Compactness defines how close the members of a cluster are to one another. Separation defines the distance between members of different clusters. Liu et al., [96] compare eleven well-known internal cluster validation measures and discuss the limitations of these measurements.

Applying cluster validation techniques to agglomerative hierarchical clustering presents computational obstacles. The agglomerative clustering does not define a specific number of clusters within the set, but rather grows each cluster through joining individual leaf nodes or smaller clusters. This method requires computing a validation measure at every join within the process. As an alternative to applying another validation measure, the agglomerative clustering using variation of information (VI) as the joining distance evaluates the clustering algorithm directly. That is, the VI is both the distance metric and the measure of cluster compactness and separation. This technique is built into the agglomerative process, and the dendrogram visually illustrates the cluster compactness and separation.

4.1.2 Verification

Verification determines if the algorithm or software functions as intended, or if programming mistakes introduce errors in the results. One extreme approach to verification is exhaustive testing, where every possible input is tested and the output evaluated [93]. This is the

most thorough tactic, but entirely unreasonable to implement in many systems concerned with testing real-world problems, as the range of inputs is unbounded. Instead, this algorithm is tested using a combination of extremal inputs, special-cases, and cleaned data from a known customer [93], providing a spectrum of test cases intended to represent the span of the input data. For this clustering method, extremal testing includes no-usage (all zero consumption records) and constant usage (indicative of a stuck meter or leak); special input cases include customers with limited historical data; and cleaned test data using a known residential customer.

4.1.3 Consistency Testing

Consistency is used here to describe the stability of a particular outcome when the same data are clustered multiple times by the method. For a clustering method to be valuable to utilities, the cluster populations must remain stable as long as the underlying behavior has no changes. This is determined by stability of individual customer cluster assignments with respect to other individuals and is discussed in the literature as cluster membership or migration of individuals within the data [61], [62], [97], [98].

4.2 Evaluation of Clustering Techniques Using Synthetic Data

Despite the lack of labeled data, unsupervised clustering algorithms must still be tested. One approach is to create synthetic data with known labels as a substitute for identifying specific groups within the dataset. As the customer-grouping problem does not have specific labels without the reference to other customers, various data processing methods create synthetic customers who will be assigned “near” and “far” distances from their source data. Starting with actual customer data, we manipulate the individual hourly meter readings to represent customers that have similar behavior, different behavior, leaks, significant changes, and limited historical data.

4.2.1 Synthetic “Similar” Customers

Shifting the entire time series forward or backward in time creates synthetic similar customers. This is equivalent to taking all the recorded meter data from a household and changing the time – instead of waking at 0745 and showering, the household now wakes at 0545. All behavior maintains the same volumes and temporal patterns. These customers will appear nearly the same when plotted in the reconstructed phase space, as the method extracts behavioral time patterns, not specific times of use. The VI distance of these synthetic similar customers will be very close to the original customer. Figure 4.2 illustrates the synthetic customer (orange) generated from the original customer data (blue) by shifting the time axis by approximately 30 hours without changing any of the hourly flow values.

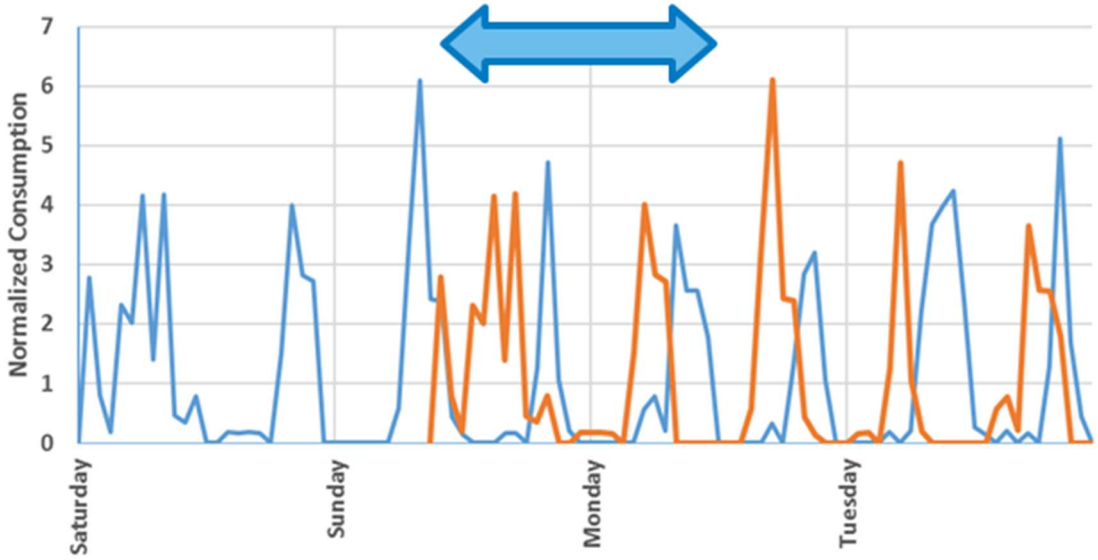

Figure 4.2 Generating a synthetic "similar" customer through temporal shift

The results of clustering synthetic similar customers show a short join distance between the donor customer and the synthetic generated customers. Figure 4.3 illustrates clustering with four synthetic customers, all generated from Customer 1 dataset. The labels indicate the type of

operation used to create the synthetic data. Customers 002 – 005 are actual customers from collected data.

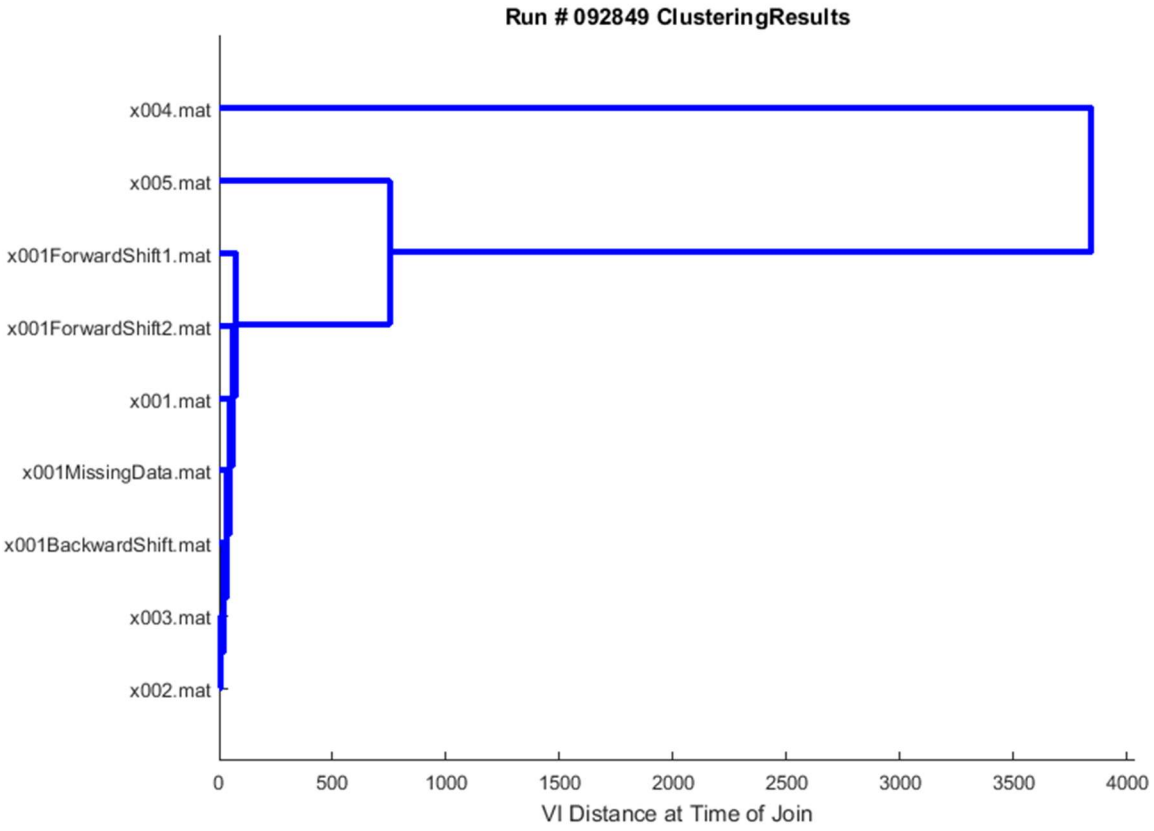

Figure 4.3 Clustering of four synthetic similar customers and five actual customers

Descriptions of the individual customer data used for Figure 4.3 are provided in Table 4-1. The customers used throughout these synthetic clustering experiments are different, with the exception of customer x001 being identical throughout and used as the donor data for all synthetic sets.

Table 4-1 Customer descriptions for synthetic similar customers example

Customer Title	Description
x001	Original data set from one customer, used as donor data to create synthetic data sets
x001 Forward Shift 1	Donor data has been shifted forward in time
x001 Forward Shift 2	Donor data has been shifted forward in time by a different number of hours
x001 Missing Data	A section of the donor data has been eliminated and replaced with all zeros
x001 Backward Shift	Donor data has been shifted backward in time
x002	Original data set from one customer
x003	Original data set from one customer
x004	Original data set from one customer
x005	Original data set from one customer

4.2.2 Synthetic "Different" Customers

Creating dissimilar customers requires changing the temporal behavior patterns within the recorded meter data. The simplest method is to draw a random permutation from the existing hourly data records, as illustrated in Figure 4.4. Repeating this process multiple times creates a group of random customers with exactly the same recorded consumption volumes as the original customer, but no discernable schedules associated with the time of day or day of week. Within the reconstructed phase space, these random permutations have no obvious structure. In the hierarchical clustering, these three random permutations are expected to have small VI distances to each other, but large VI distances to the original customer who has daily or weekly behavioral patterns.

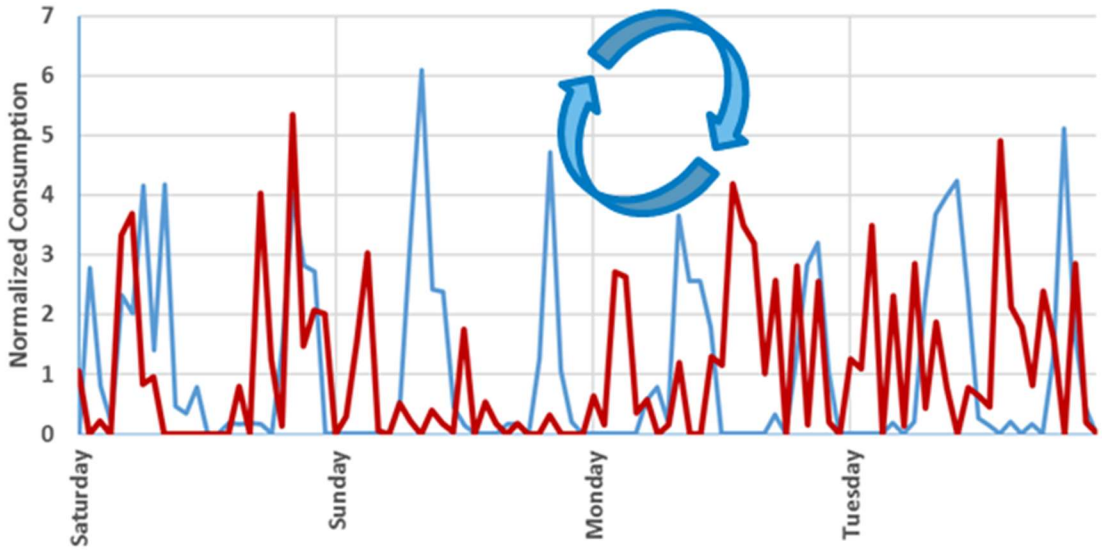

Figure 4.4 Generating a synthetic "different" customer through random permutation of hourly flow measurements

As the name implies, synthetic different customers tend to be grouped randomly far from the donor data set. Figure 4.5 shows these results. One of the random permutation synthetic meters is grouped near to the donor meter, while the other two are grouped further away. These results are not surprising, as random permutations occasionally form similarities that resemble the source.

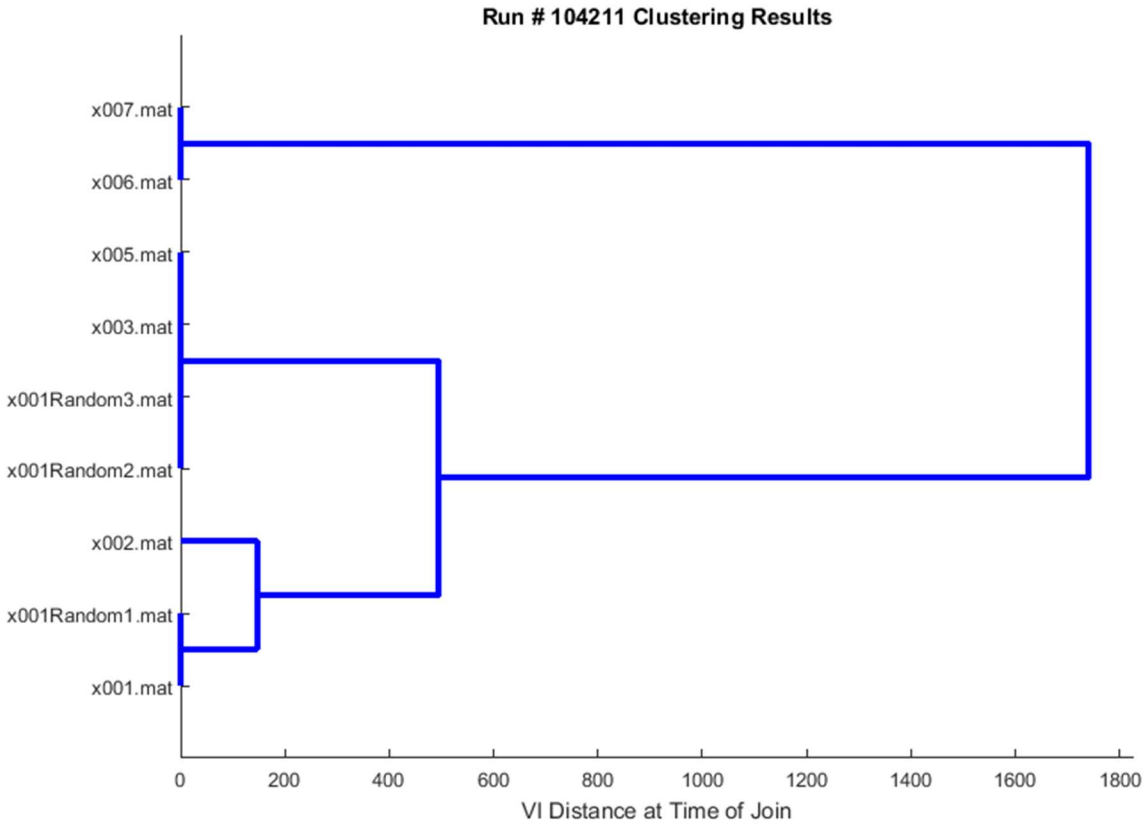

Figure 4.5 Clustering of three synthetic different customers generated through random permutations of the time series, with six actual customers

Descriptions of the individual customer data used for Figure 4.5 are provided in Table 4-2. The customers used throughout these synthetic clustering experiments are different, with the exception of customer x001 being identical throughout and used as the donor data for all synthetic sets.

Table 4-2 Customer descriptions for synthetic different customers example

Customer Title	Description
x001	Original data set from one customer, used as donor data to create synthetic data sets
x001 Random 1, 2, 3	Three different random permutations of the donor data
x002	Original data set from one customer
x003	Original data set from one customer
x005	Original data set from one customer
x006	Original data set from one customer
x007	Original data set from one customer

4.2.3 Synthetic "Leak" Customers

In the water industry, a leak is any unintended loss of water from the pressurized distribution system [99]. While much of the focus in the water industry has been on distribution network leakage [70], [100]–[103], consumer-side (after the meter) leakage is important to the individual residents and commercial accounts, as they must pay for the lost water and the maintenance caused by water damage [100], [104]. Leaks may occur when a mechanical failure has occurred in a fixture or pipe, such as a leaky valve on a commode, a failed weld on a pipe joint, or a worn seal on a faucet. Human error may also appear as a leak from the perspective of measured flow – forgetting to turn off an irrigation system. Due to the low probability of detection, small volume leaks may run for weeks or months before repair, contributing to the total volume lost.

To test the ability of the clustering algorithm to separate leaks from typical behaviors, synthetic leaks are created by choosing a duration the leak is present and a volume per hour of the recorded leak flow. At a random time, the leak begins, and the leak volume is added to every hourly data point for the duration, as illustrated in Figure 4.6 for a very short duration leak (blue)

compared to original customer data (gold). This assumes a fixed-volume leak, which is not entirely accurate. Future improvements to this algorithm should represent more realistic leaks: a small initial flow rate, increasing over time, sometimes progressing to a rupture with high flow rate [99].

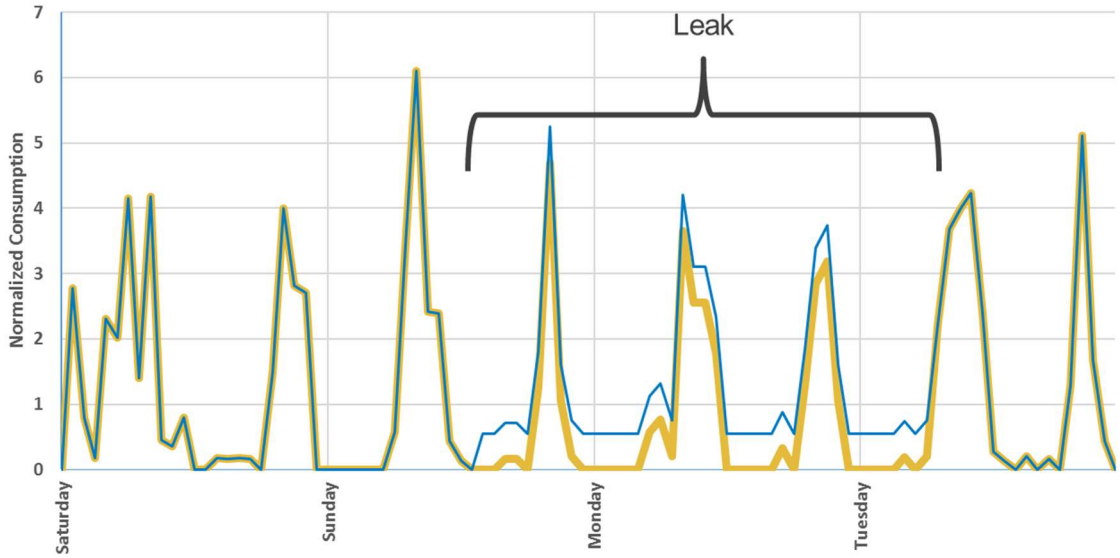

Figure 4.6 Generating a synthetic "leak" customer by adding a fixed flow volume for a random duration

The synthetic leak customers have been generated from Customer 001 by creating either a small volume of 0.75 gallons per hour for a duration of 500 consecutive hours or a medium volume of 2.3 gallons per hour for 200 consecutive hours. This does not imply leaks follow these volumes and durations, but provided a case for supporting future work to investigate these results. Figure 4.7 illustrates the output of the clustering algorithm for the leak customers as compared to six actual customers, including the donor data. The medium leak of 2.3 gallons has been grouped much further from the original customer than the small leak.

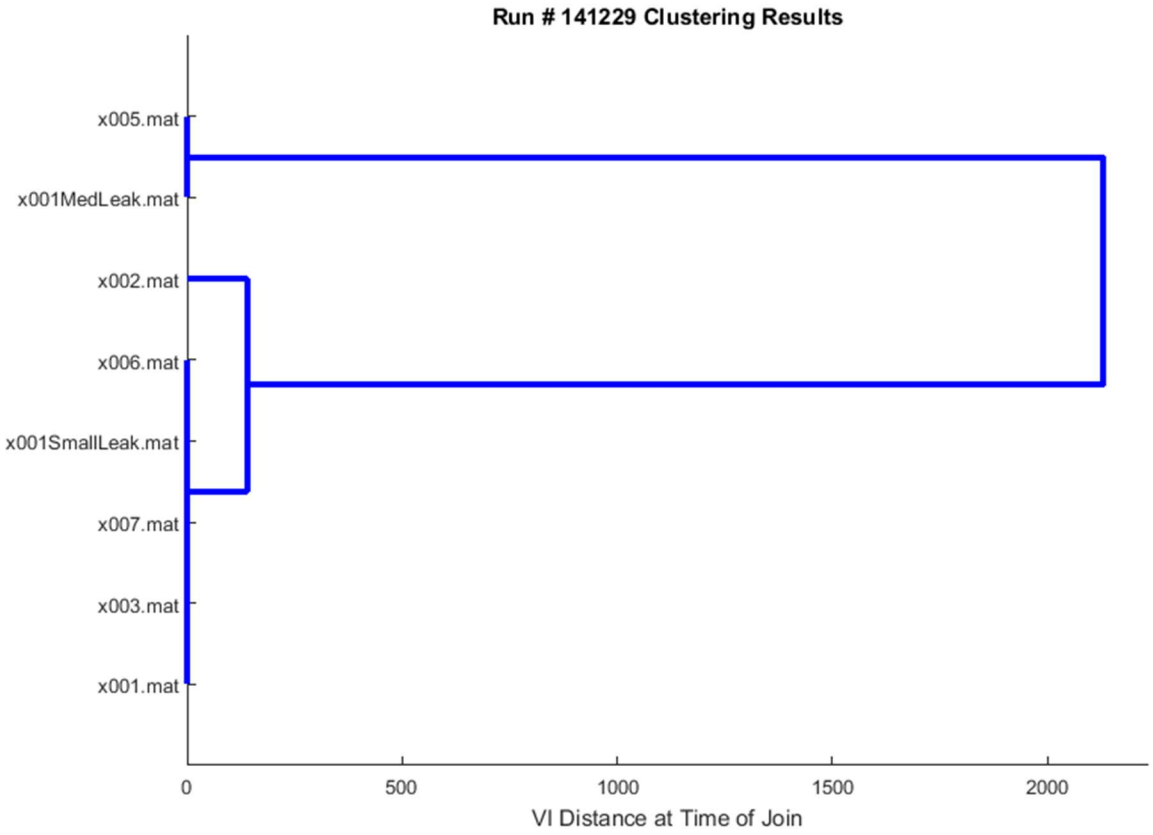

Figure 4.7 Clustering of two customers with synthetic leak events and six actual customers

Descriptions of the individual customer data used for Figure 4.7 are provided in Table 4-3. The customers used throughout these synthetic clustering experiments are different, with the exception of customer x001 being identical throughout and used as the donor data for all synthetic sets.

Table 4-3 Customer descriptions for synthetic leak events example

Customer Title	Description
x001	Original data set from one customer, used as donor data to create synthetic data sets
x001 Medium Leak	A leak of 2.3 gallons per hour has been applied to 200 consecutive hours with a random start time
x001 Small Leak	A leak of 0.75 gallons per hour has been applied to 500 consecutive hours with a random start time
x002	Original data set from one customer
x003	Original data set from one customer
x005	Original data set from one customer
x006	Original data set from one customer
x007	Original data set from one customer

4.2.4 Extremal Testing

Extremal testing is performed when the test data are selected to explore the boundaries of the input data space, anticipating the test to exercise boundaries of the output space [93]. In the case of this research, extremal testing occurs when tests are performed on synthetic data with no consumption, continuous consumption, and random consumption values. These extreme cases are designed to be outliers within the data and should have appropriate clustering distances from the other data. Experiments using fixed consumption values indicate the method performs poorly as it currently is designed.

During the first step of modelling for every customer, the Gaussian mixture models fail to converge within the phase space if all recorded volumes are identical. After manually altering the GMM to have a small nonzero volume, the next step in clustering also failed to converge, and this failure propagated throughout the experiment. Automatic handling of extremal cases – customers with no consumption or constant volume consumption – requires implementation of a new

preprocessing step or a modification to this method to accommodate the near-zero-volume model components.

4.3 Consistency Testing

The method of using Gaussian mixture models to represent real-world data sets introduces an element of randomness during the generation of the models. This randomness may result in no two GMMs created on the same data set being identical. Since the input to the clustering mechanism is not identical for each trial, it is important to test the procedure multiple times to determine if the results are widely variant or inconsistent. Common accepted practice demands testing software for consistency [93], [95].

If the same raw time series data is provided, the output GMMs ought to be comparable. Further, the clustering of similar GMM inputs ought to produce similar clustering results. The underlying relationships in the GMMs and clustering results will then generate dendrograms with structure and distances that are consistent from trial to trial. Figure 4.8 clarifies the process of testing clustering consistency. The illustration has only three customers for simplicity, but the experiments include clustering with every customer.

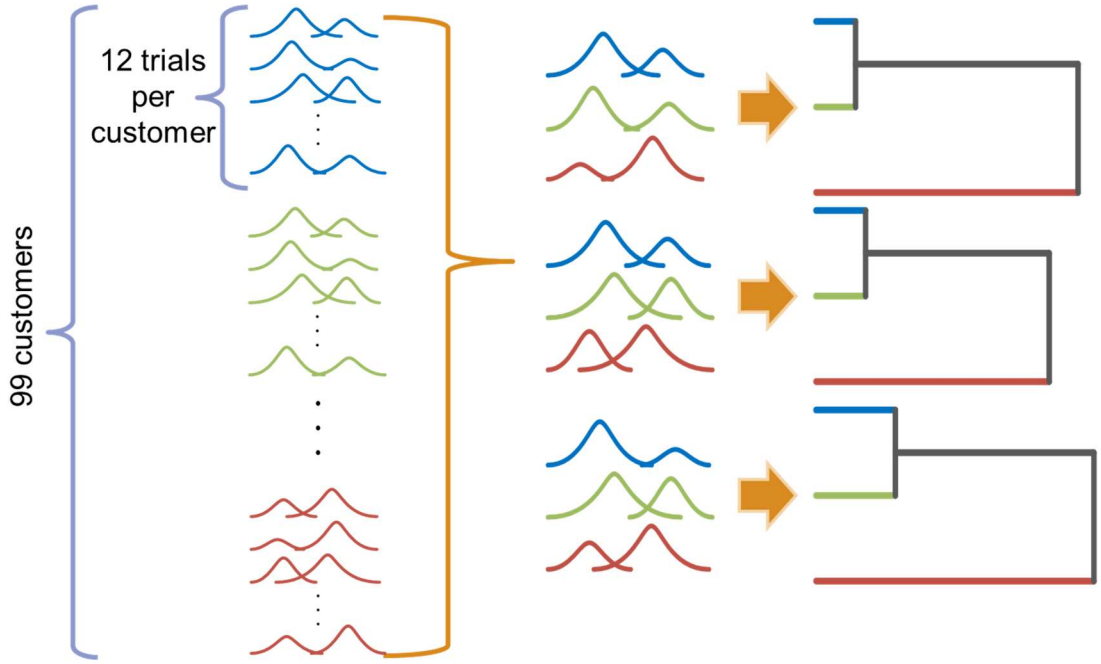

Figure 4.8 Experiment configuration for testing consistency of the clustering method

First, twelve different GMM trials are performed for each customer (blue customer, green customer, red customer, etc.). Then, a random GMM is chosen from each customer's set of trials, and those are clustered to form a dendrogram. The dendrograms are then compared to each other. If the experiment process is consistent, the dendrograms should be constructed in a similar, but not exact, manner. Customers with short variation of information distances at the time of join should maintain a short distance through all trials of this experiment, while those with large VI distances when joined to the other customers should maintain the large distance as well.

4.3.1 Interpretation of Consistency Dendrogram Figures

To display the results of consistency testing in a meaningful and clear manner, several dendrograms are grouped horizontally, as shown in Figure 4.9. Each dendrogram is one result of a clustering experiment trial. The horizontal grouping allows comparison of results between several trials. The Gaussian mixture models representing each customer are different for each

trial, contributing to the variation in results. Individual trials are labeled with the title “Run #” and a number related to the timestamp of the experiment for recordkeeping purposes. Two dendrograms with the same Run # title are results from the same experiment.

Customer labels are consistent through all trials of an experiment and through all dendrograms on the figure. However, other experiments may label a different meter Customer 001, based on the order the customer models were loaded into the program for that experiment. The title of the figure indicates the experiment displayed, and a different title implies the customer labels may be different meters, unless noted otherwise.

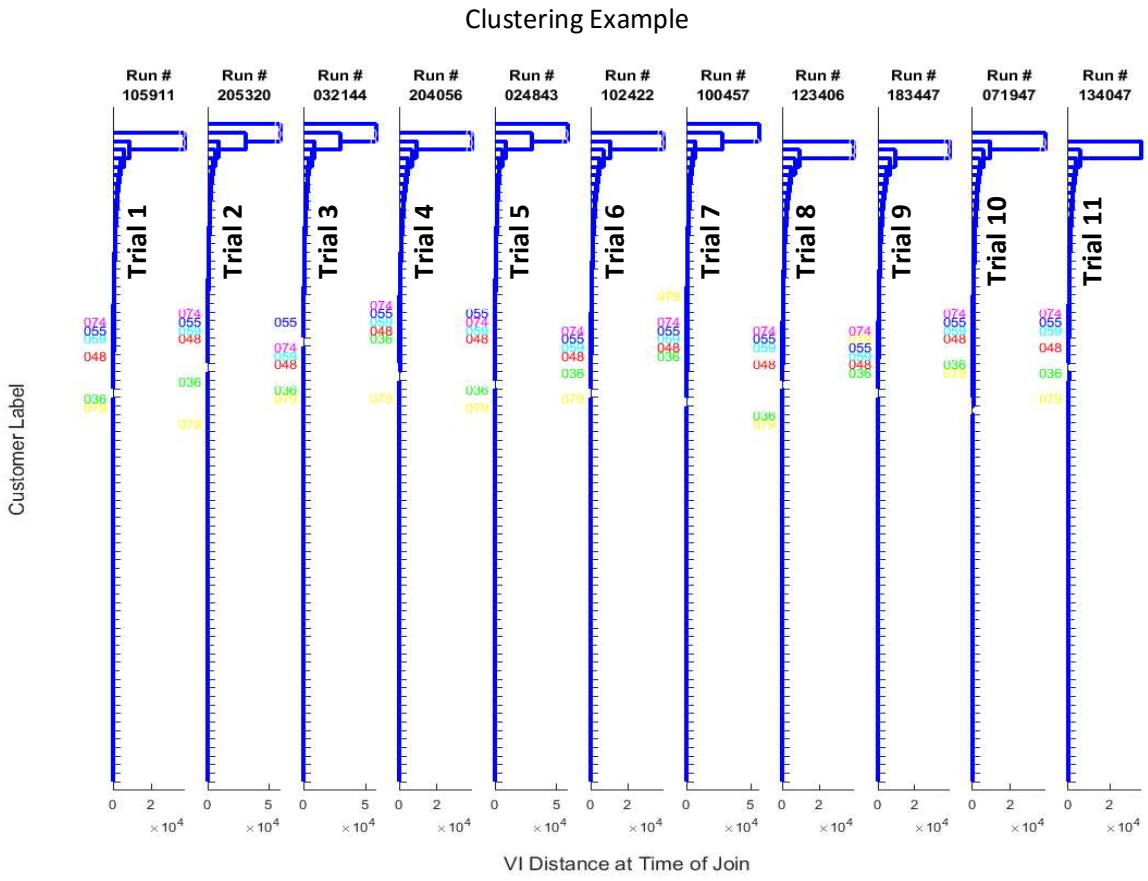

Figure 4.9 Sample results diagram from consistency experiments

As these graphics present a great deal of data, they can be confusing to interpret. Most of the labels on the vertical axis for each dendrogram are hidden to reduce visual clutter, and

specific labels are colored for emphasis in the comparison. Some experiments show multiple figures with identical collections of trial dendrograms, except for the highlighted customers. This approach shows all the results without being visually overwhelming.

Figure 4.10 shows one way to interpret these figures. The red path identifies the same customer across all dendrograms and illustrates the volatility of the position for Customer 048 in the clusterings. The two black bars indicate the uppermost and lowermost positions among the clusterings, with the span between these two bars a representation of the volatility. If this experiment were perfectly repeatable, the volatility would be very small, the black bars would be nearly touching, and the highlighted customer would have exactly the same location in every clustering.

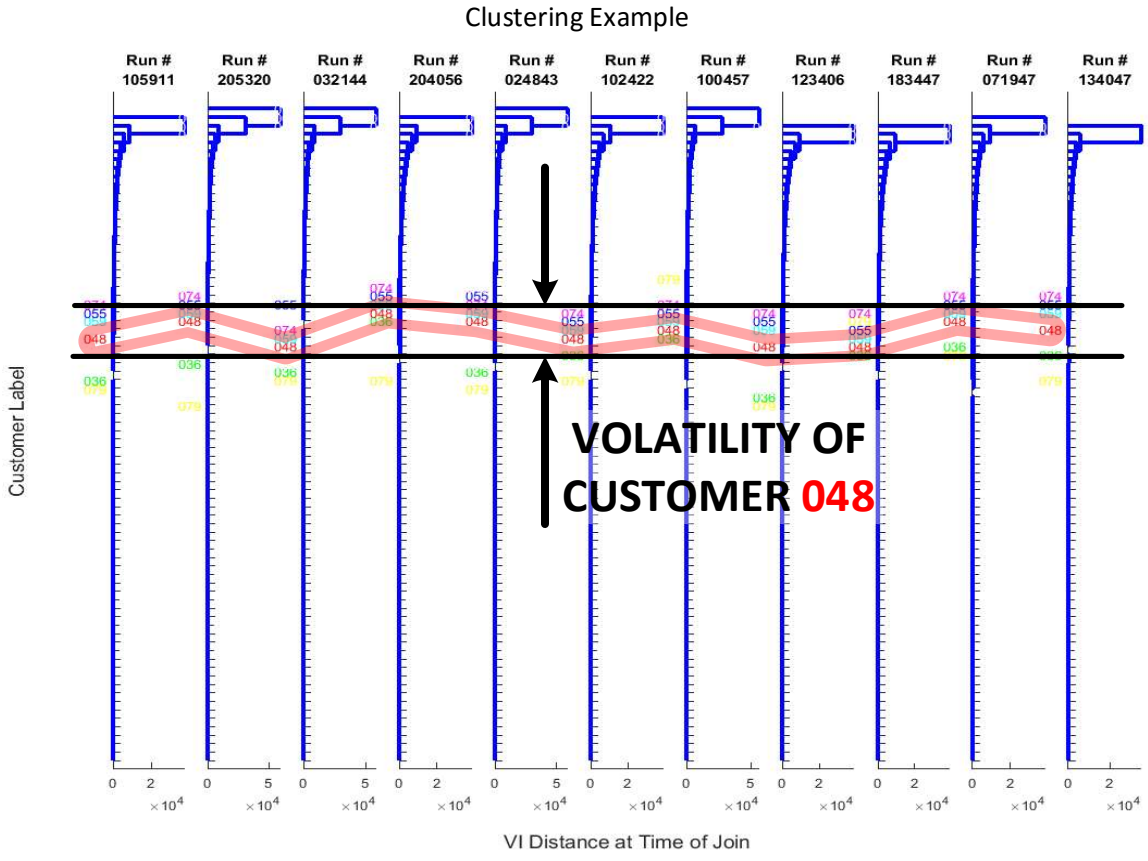

Figure 4.10 Illustration of the volatility of one customer throughout many experimental trials

In general, more volatility is less useful to a utility implementing this system. The utility wishes to have customers grouped consistently each time the clustering is run. A highly volatile result indicates the models being generated for each trial are not similar. The definition of a robust measurement for volatility is out of scope for this dissertation, but this remains a useful representation to evaluate the consistency of test results and to compare the experimental methods presented in this work.

4.3.2 Results of Hierarchical Clustering with VI Distance

This experiment compares multiple GMMs created from the same 99 customers. Each trial of the experiment uses a new set of GMMs, and creates unique cluster models at each stage of the clustering. The purpose of the experiment is to demonstrate the clustering method can be repeated and the output will be similar after each clustering. In a real-world application, the utility will need to run the clustering at intervals to group the customers. If the groups are very volatile, the method is not useful. In this sense, volatility describes the cluster group membership changing drastically each time the clustering runs, distances between one customer and his/her peers fluctuating wildly, or a customer moving from the large group of "typical" customers into the smaller group of "unusual" customers.

Over the next several pages, results of the consistency testing using traditional variation of information distance are presented. In each figure, several dendrograms are placed horizontally, each with a heading label Run # xxxxxx. These numbers indicate the time stamp that particular clustering experiment occurred, for documentation and tracking purposes. The dendrograms all include the same original customers, but not the same Gaussian mixture models for each customer, resulting in the variations seen. Generic labels (Customer 098, etc.) have replaced meter serial numbers, to preserve the anonymity of the end users and the utility. Customer labels are consistent throughout every experiment in the figure. Therefore, "Customer 098" always describes the same meter, regardless of the individual subplot dendrogram in

question, and this entire set of figures maintains consistent labels. Figure 4.11 shows the full set of customers and all the labels.

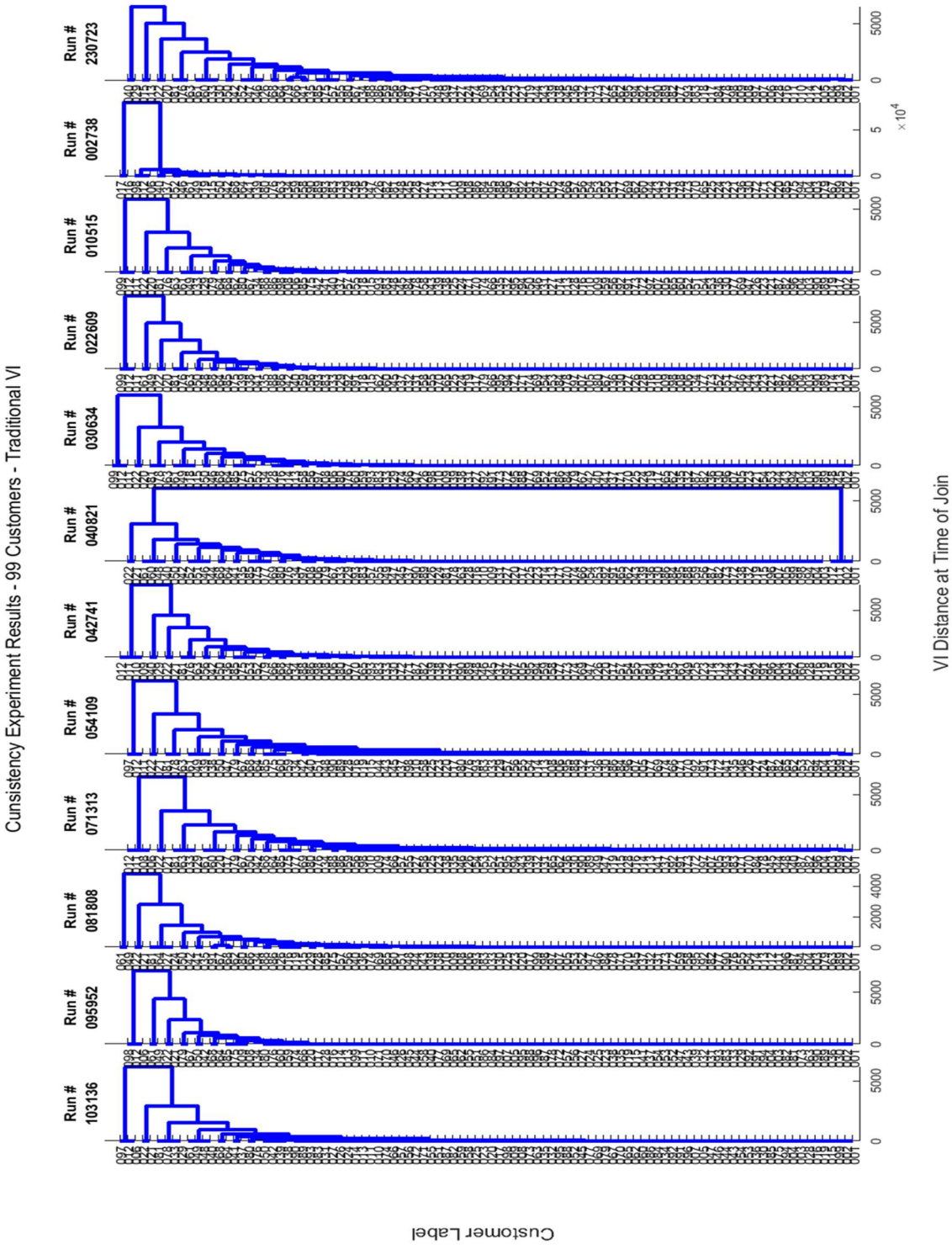

Figure 4.11 Consistency experiment results with all labels visible

Clearly, the diagrams become difficult to read with every label visible. Separate figures have been created from Figure 4.11 to improve legibility, with colorized labels focusing upon a smaller group of customers and hiding the remaining labels. Care has been taken to present all the customers in this manner, with no exclusions. This produces a set of 20 diagrams, shown over the next several pages (Figure 4.12 through Figure 4.31). Presenting the data in this manner allows visual identification of those customers with generally consistent results and those customers with clustering placements that vary wildly between the different experiments. The figure title identifies the customer labels highlighted in each illustration. Some experimental trials found the model of a particular customer never to intersect with the others, and in these cases, the customer label is not visible on the dendrogram for that trial as the distance to the rest of the meters is infinite.

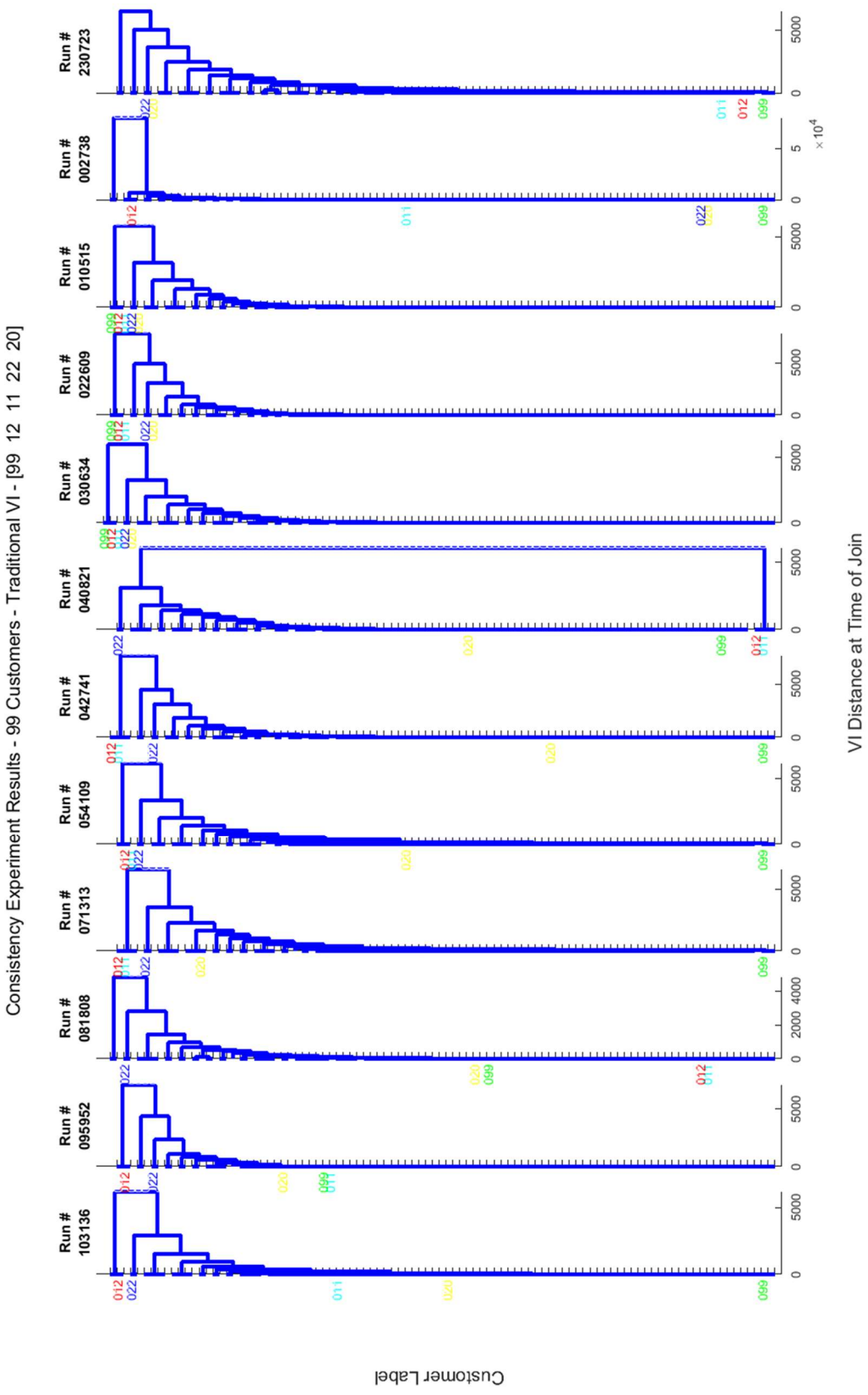

Figure 4.12 Consistency experiment results 1 of 20

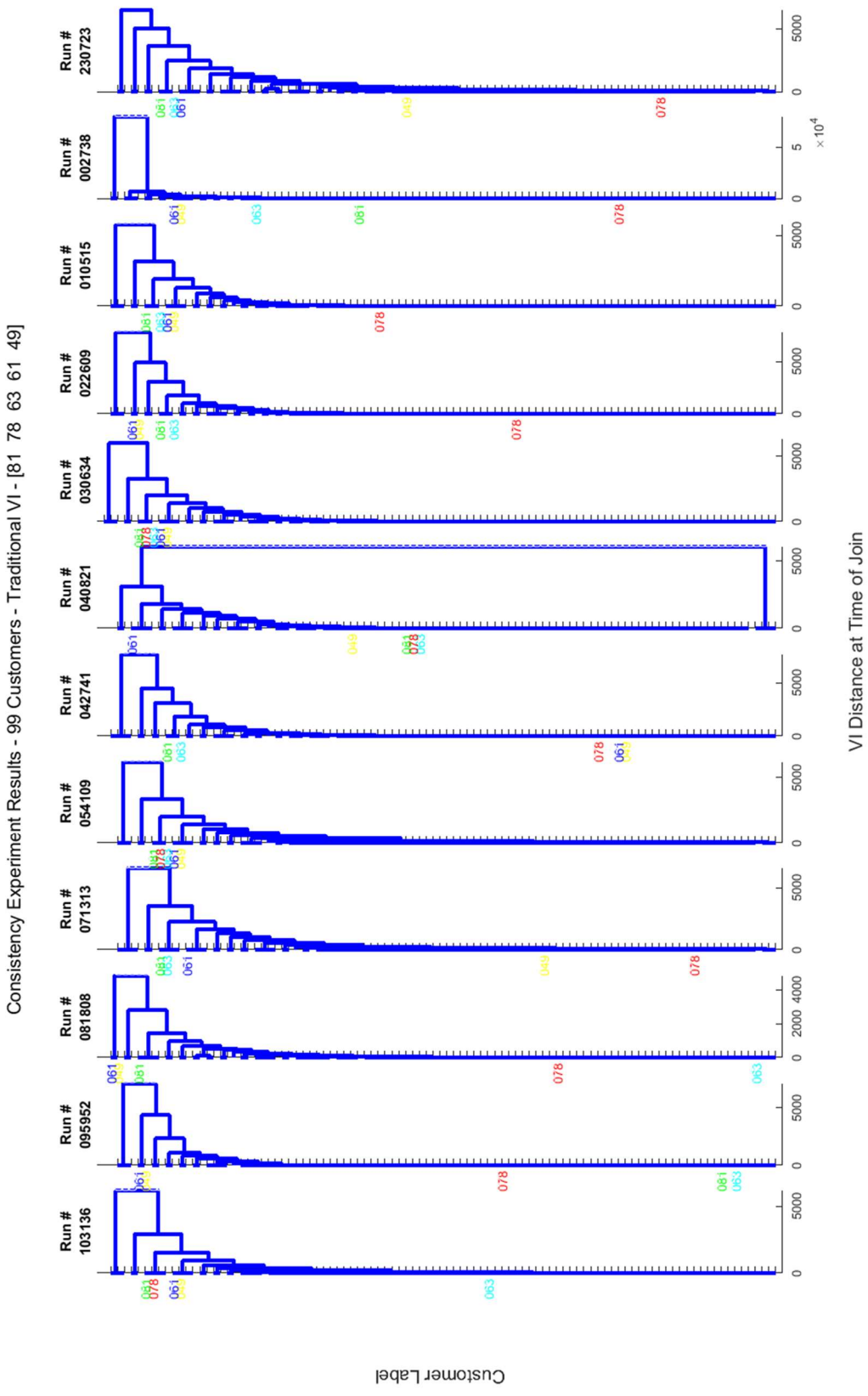

Figure 4.13 Consistency experiment results 2 of 20

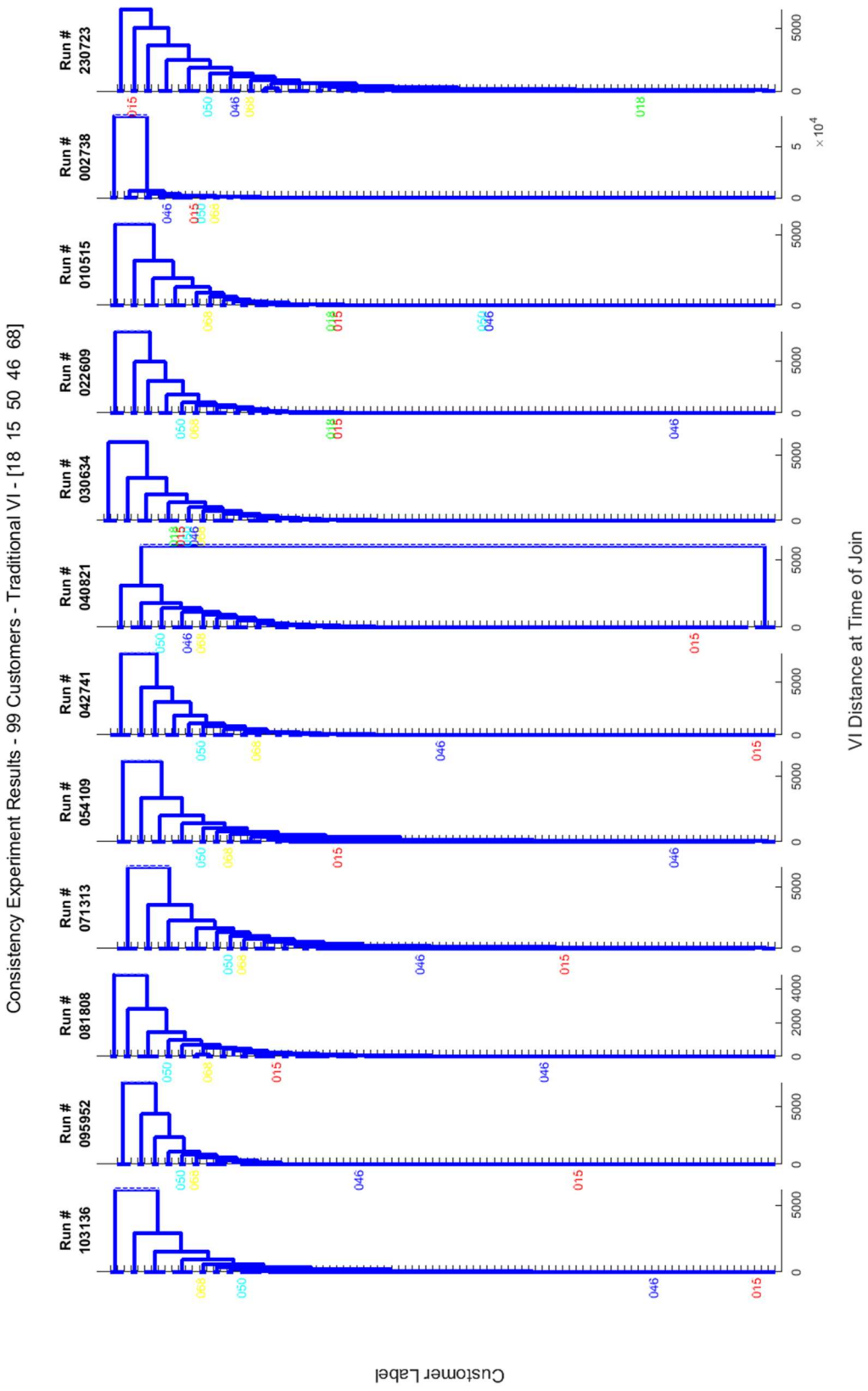

Figure 4.14 Consistency experiment results 3 of 20

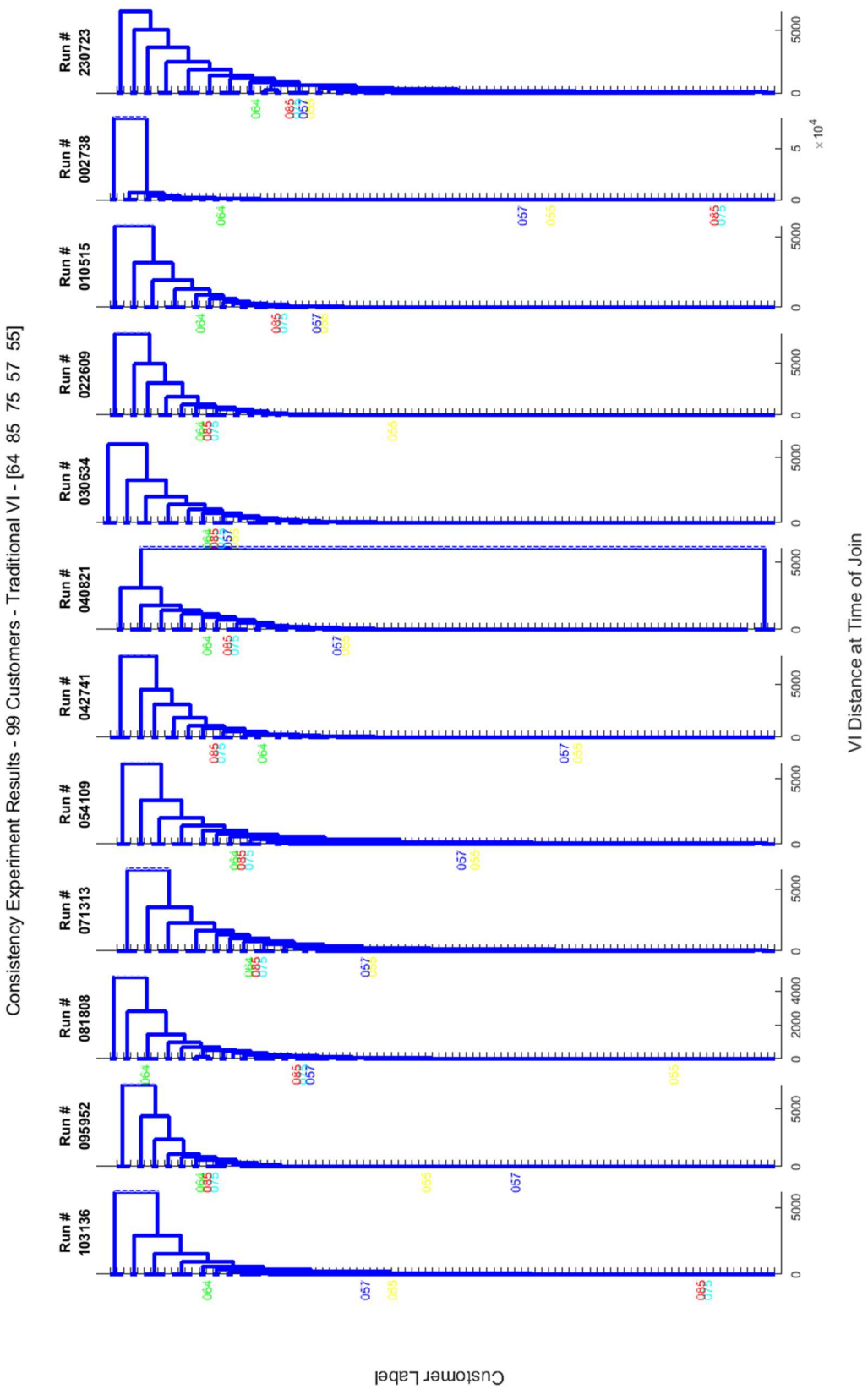

Figure 4.15 Consistency experiment results 4 of 20

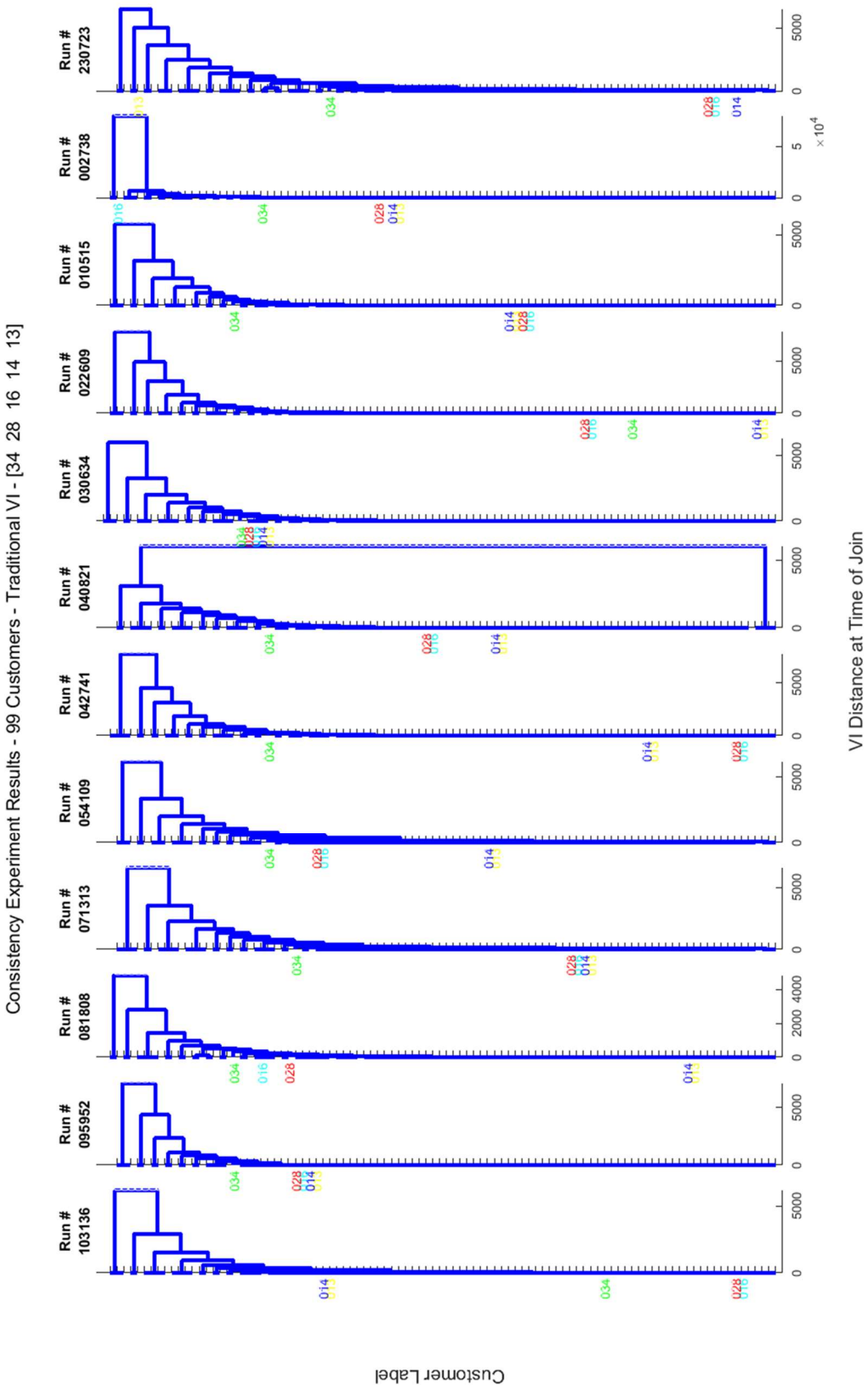

Figure 4.16 Consistency experiment results 5 of 20

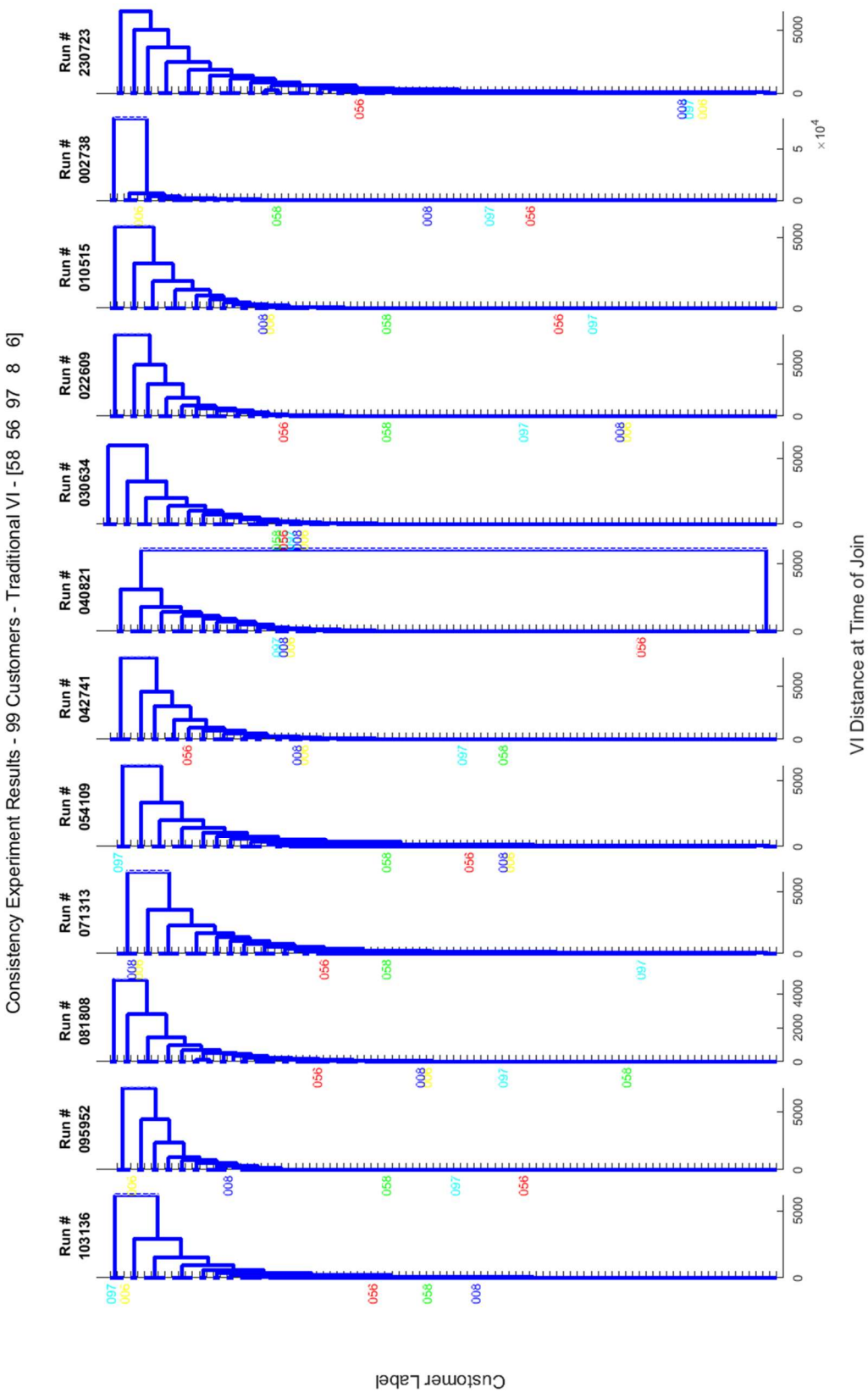

Figure 4.17 Consistency experiment results 6 of 20

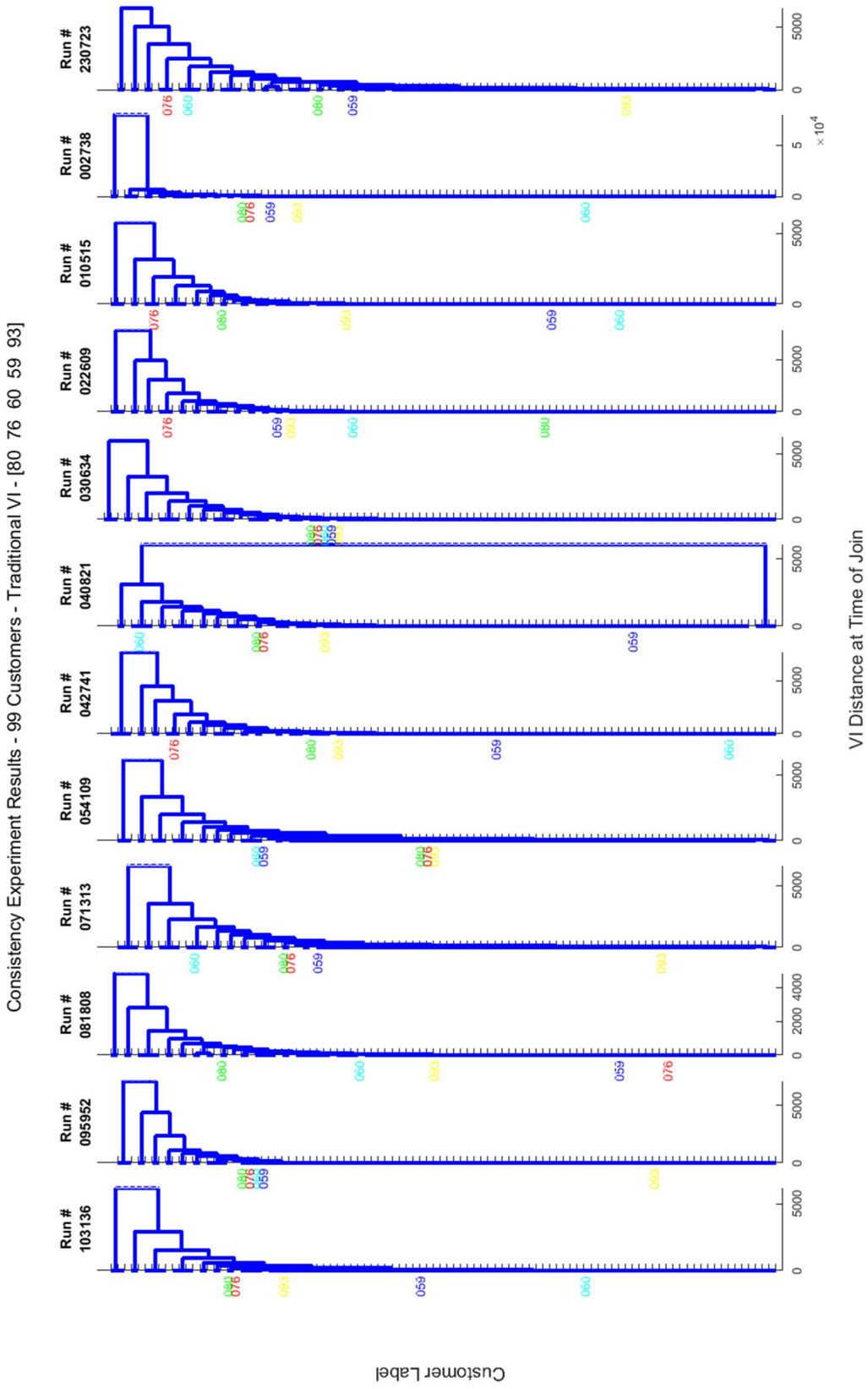

Figure 4.18 Consistency experiment results 7 of 20

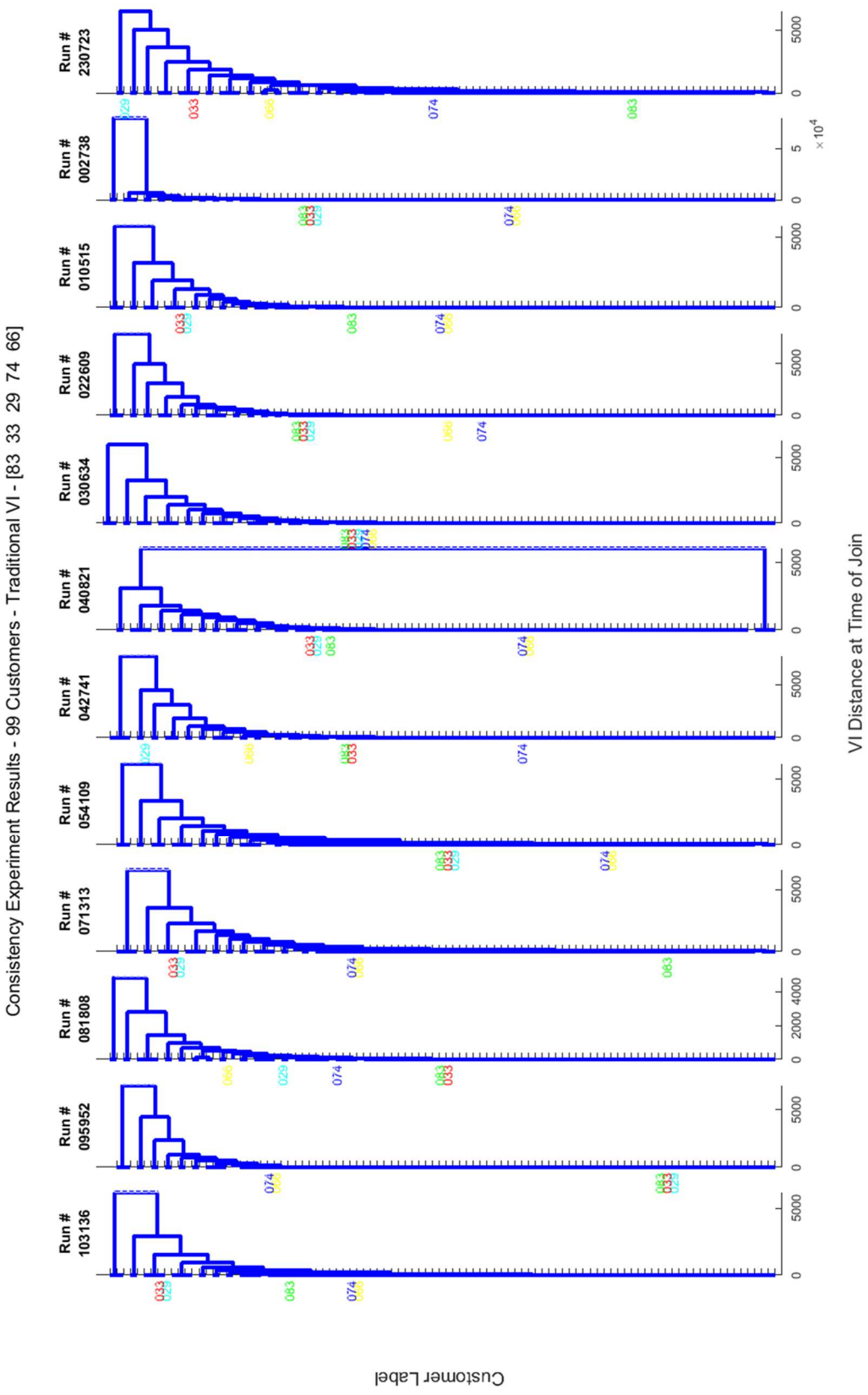

Figure 4.19 Consistency experiment results 8 of 20

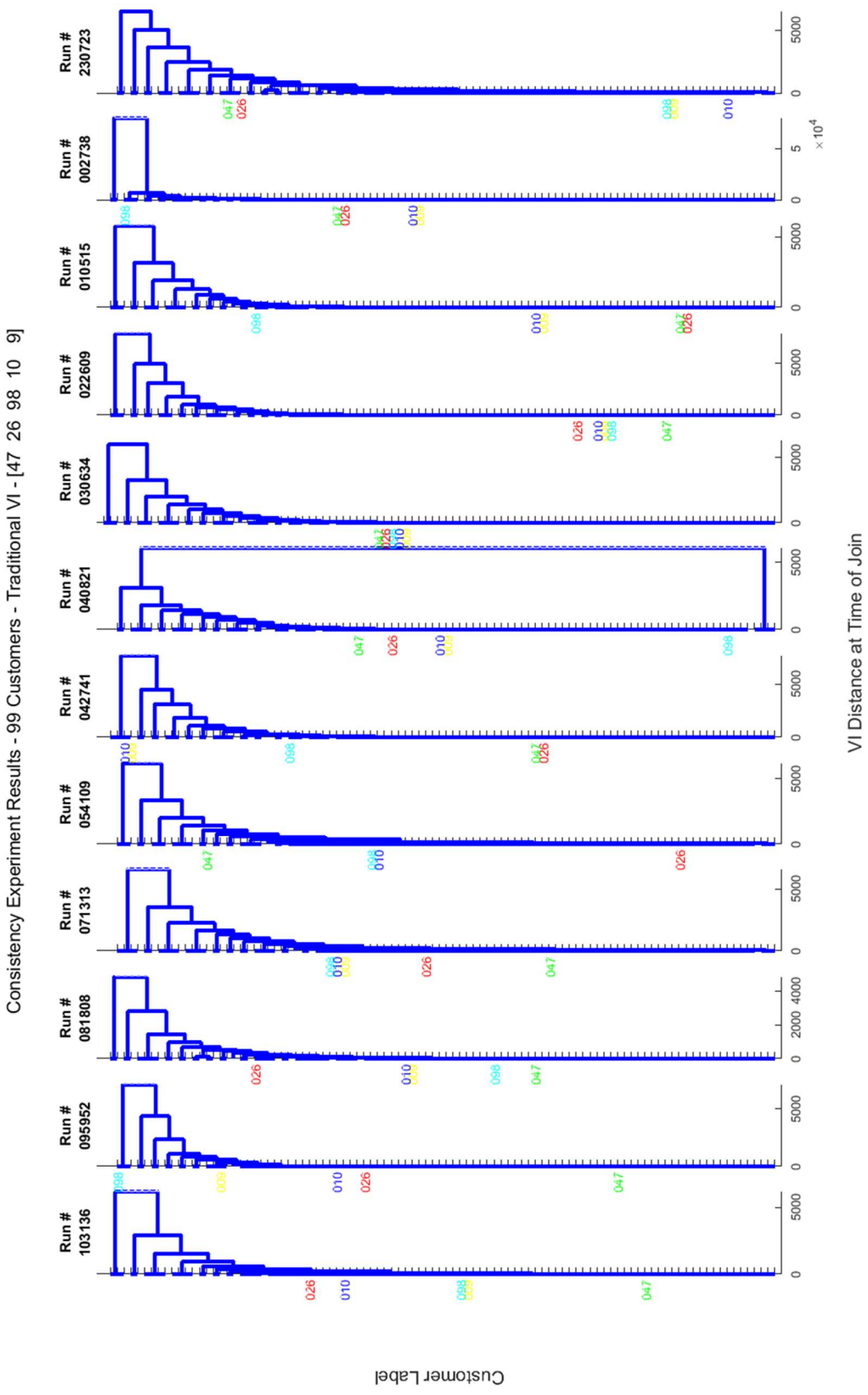

Figure 4.20 Consistency experiment results 9 of 20

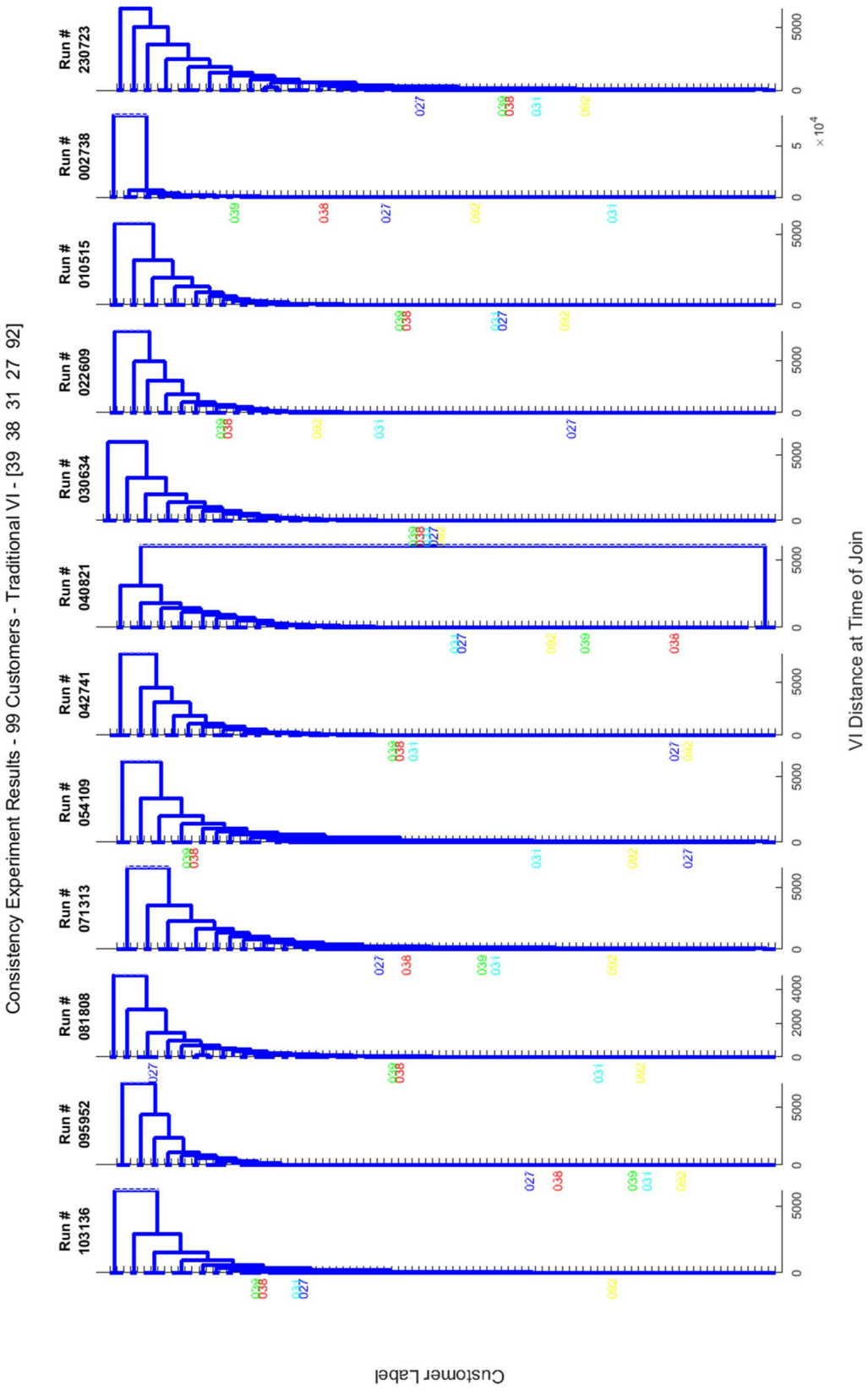

Figure 4.21 Consistency experiment results 10 of 20

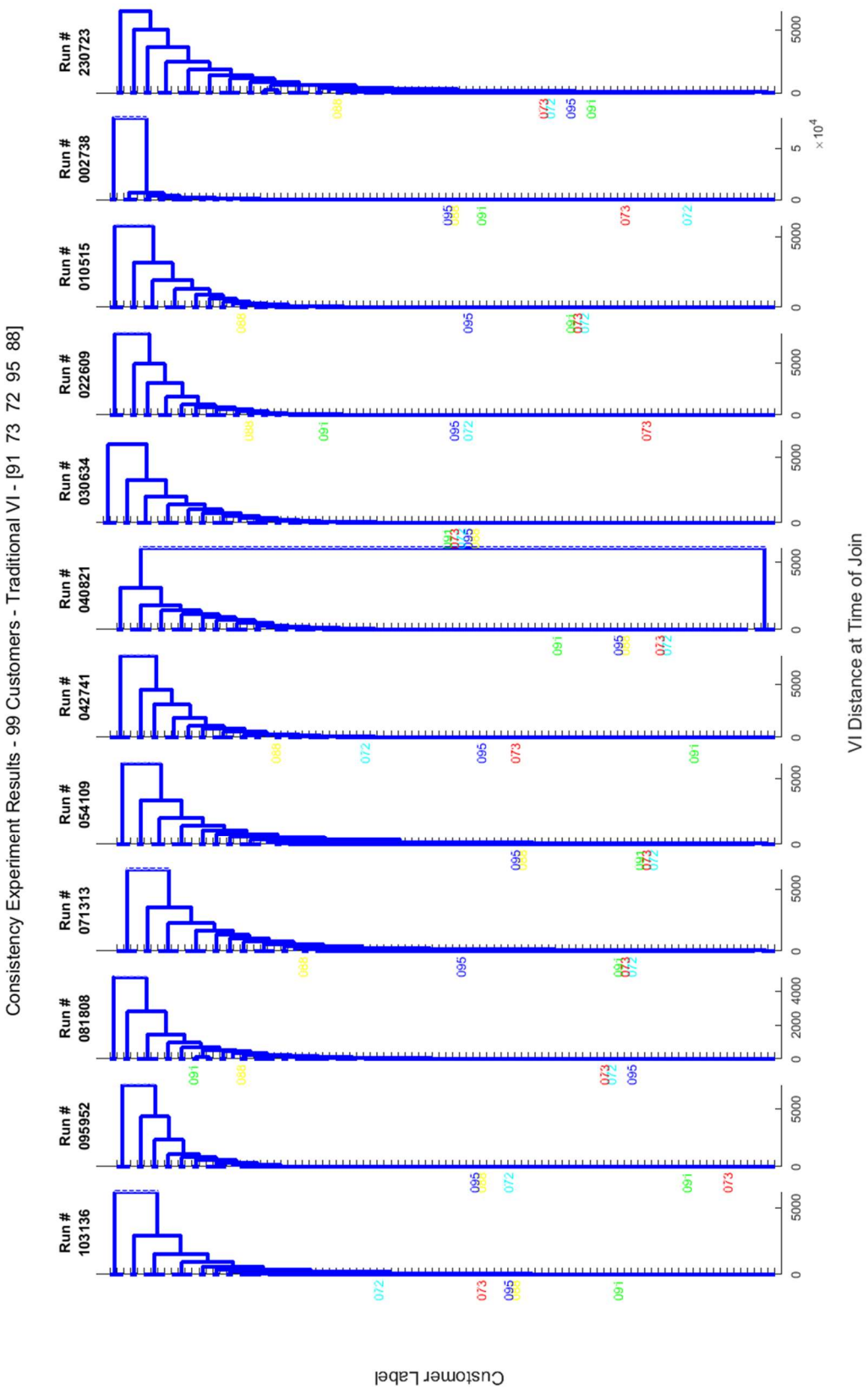

Figure 4.22 Consistency experiment results 11 of 20

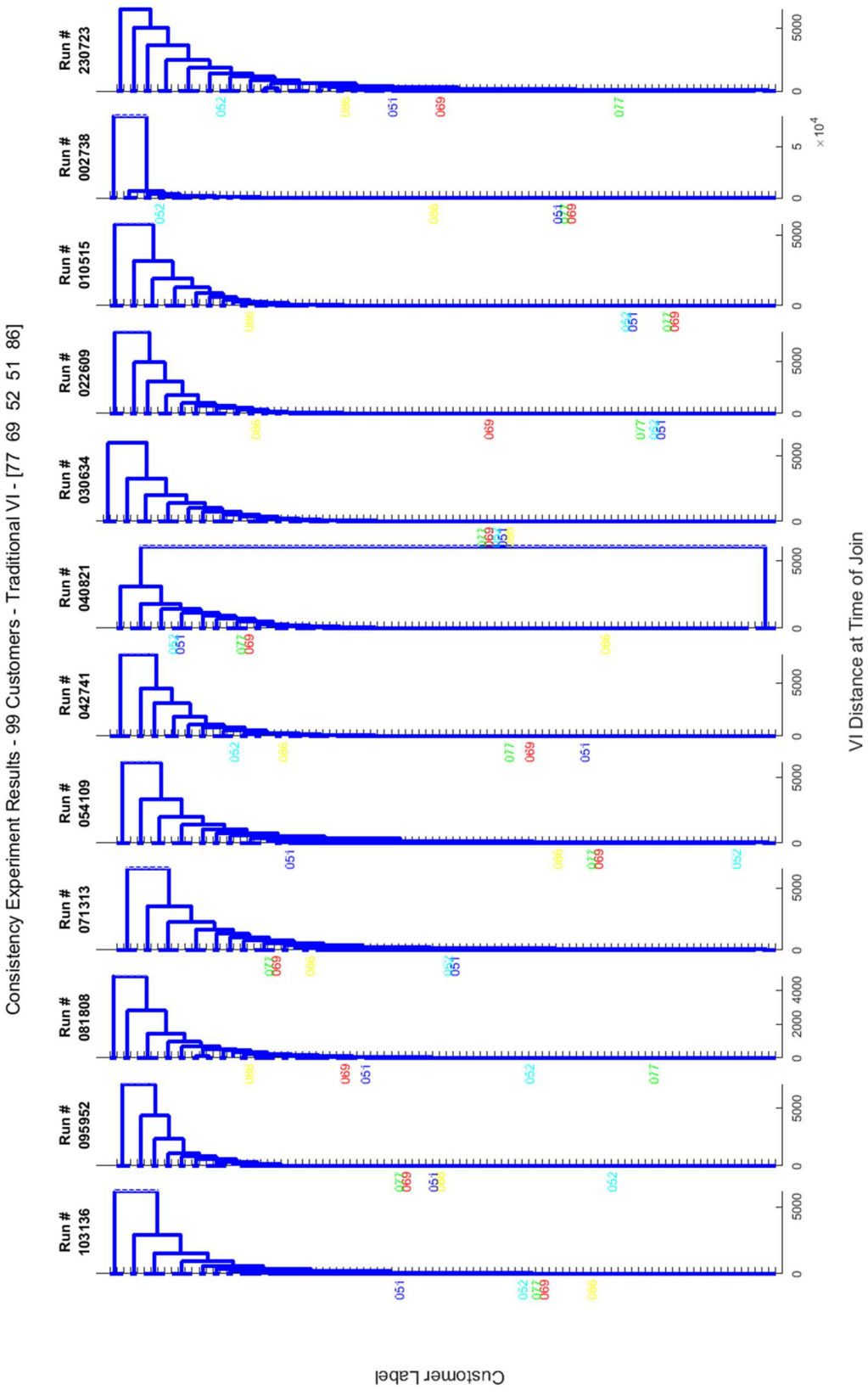

Figure 4.23 Consistency experiment results 12 of 20

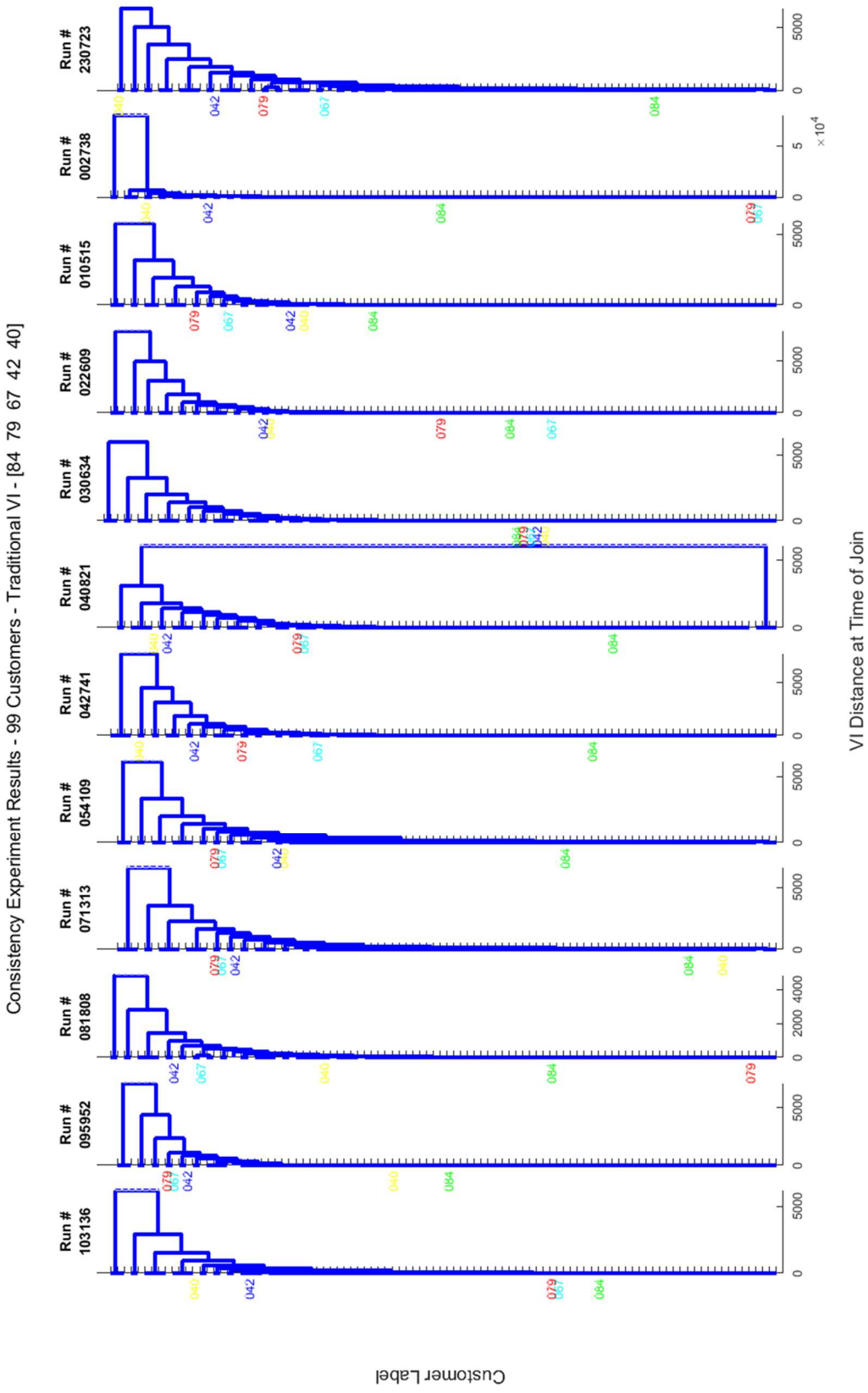

Figure 4.24 Consistency experiment results 13 of 20

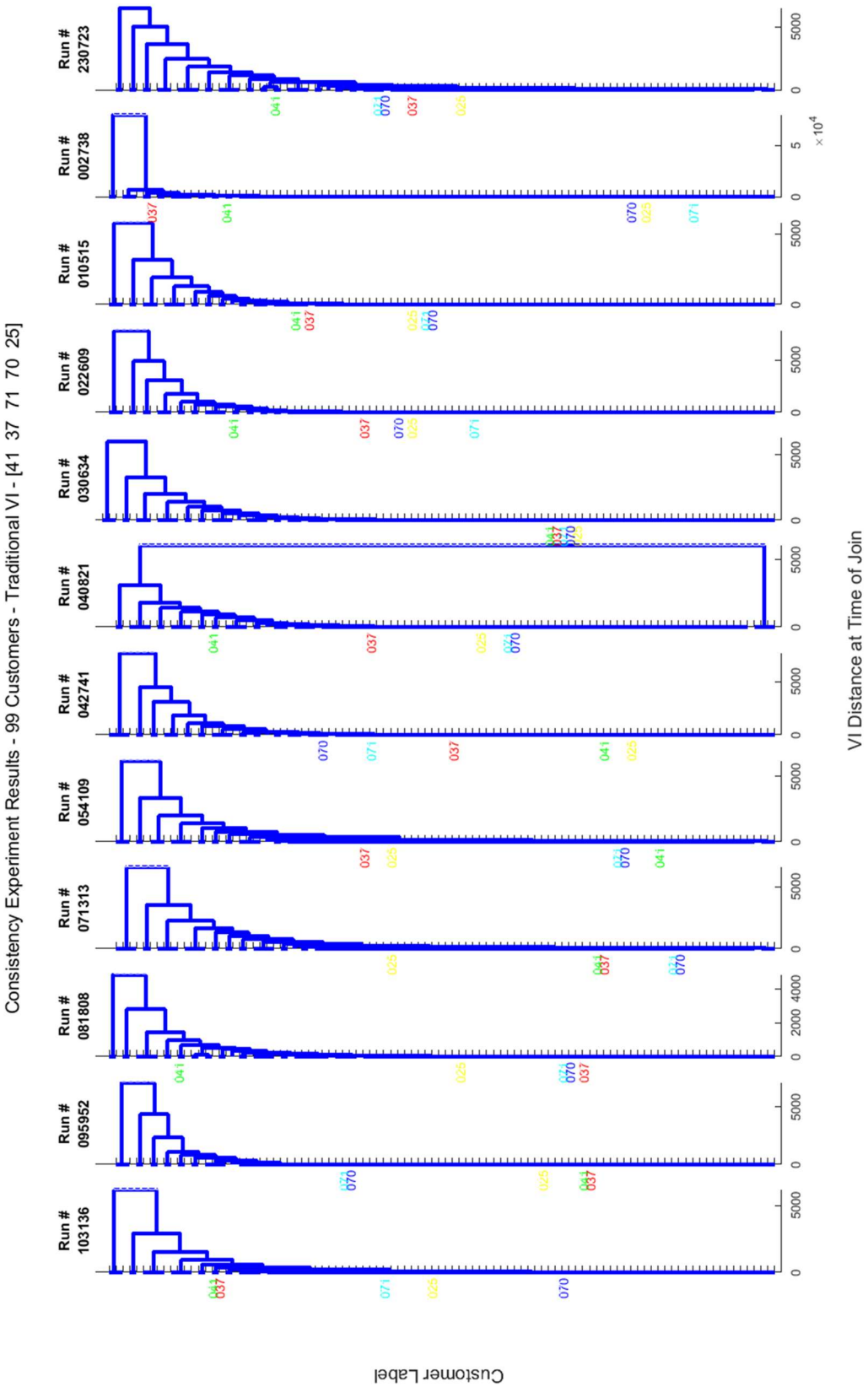

Figure 4.25 Consistency experiment results 14 of 20

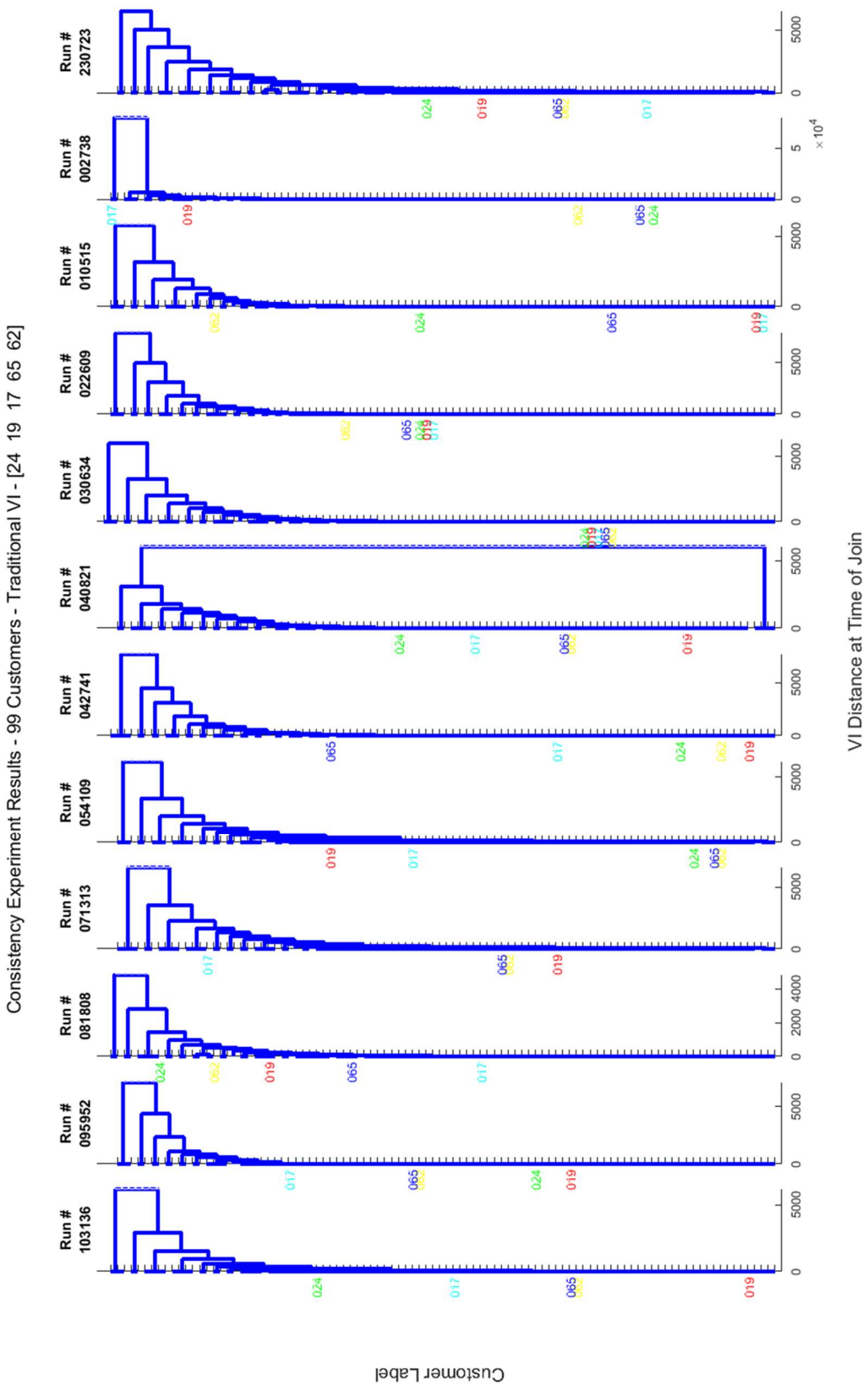

Figure 4.26 Consistency experiment results 15 of 20

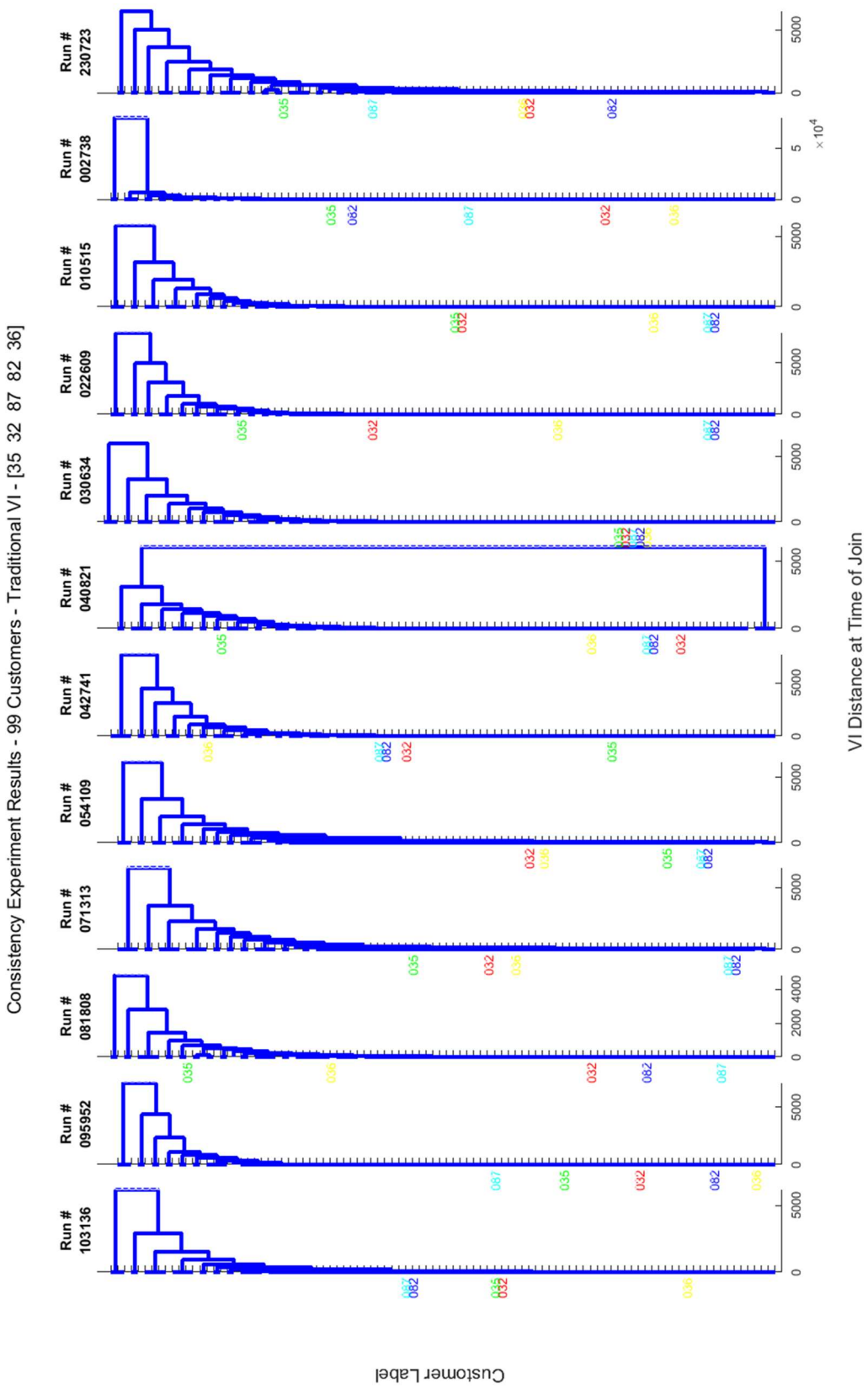

Figure 4.27 Consistency experiment results 16 of 20

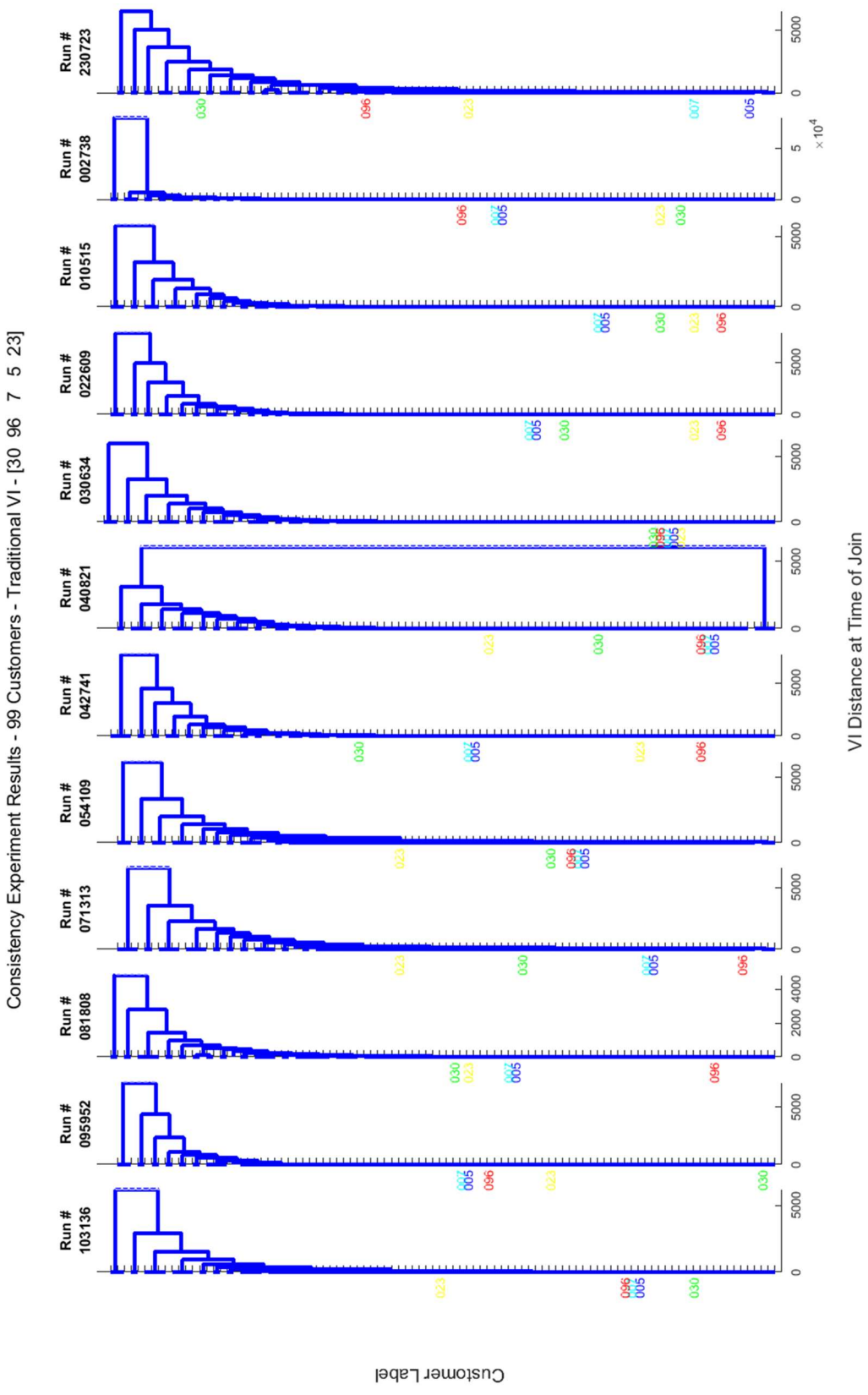

Figure 4.28 Consistency experiment results 17 of 20

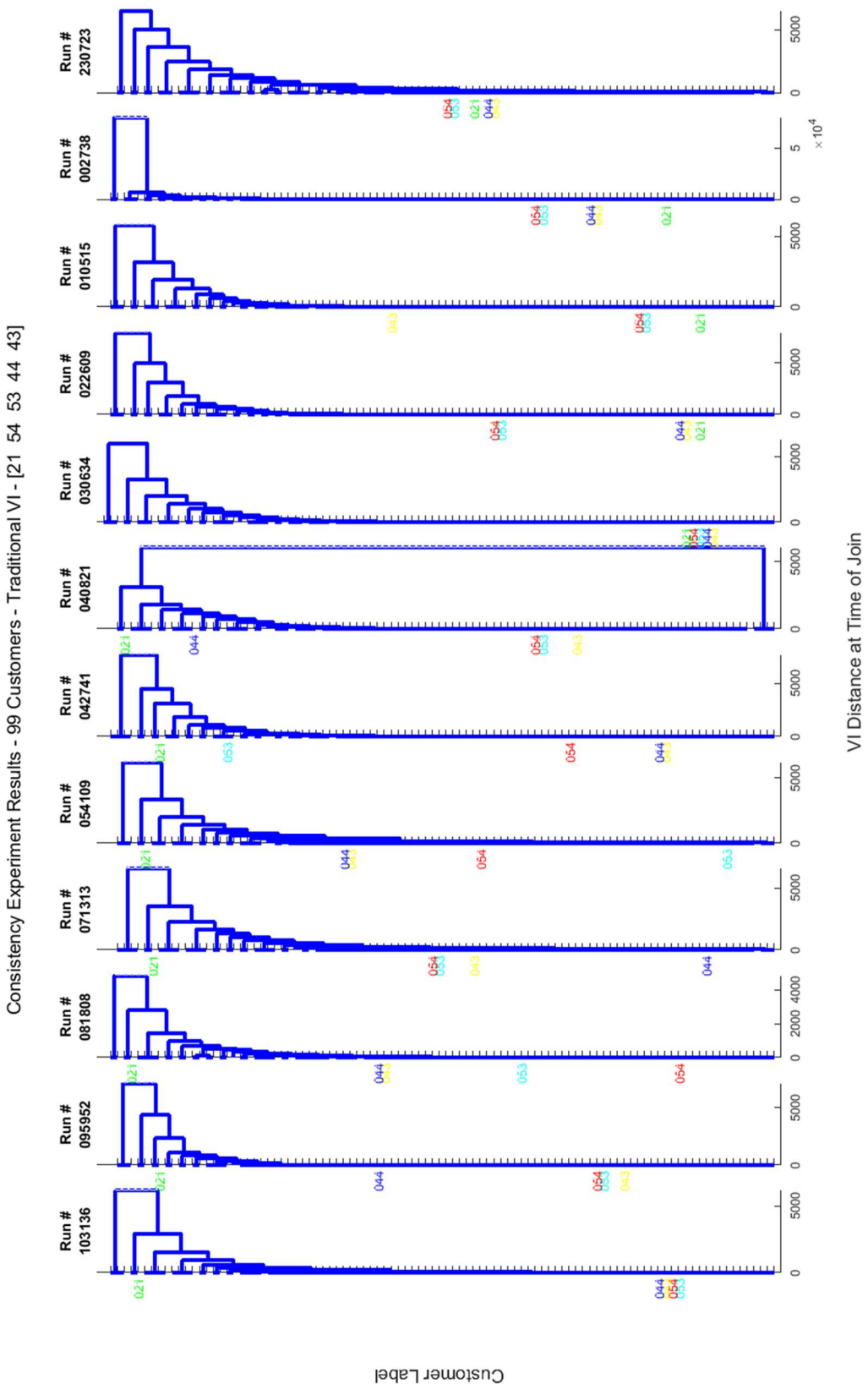

Figure 4.29 Consistency experiment results 18 of 20

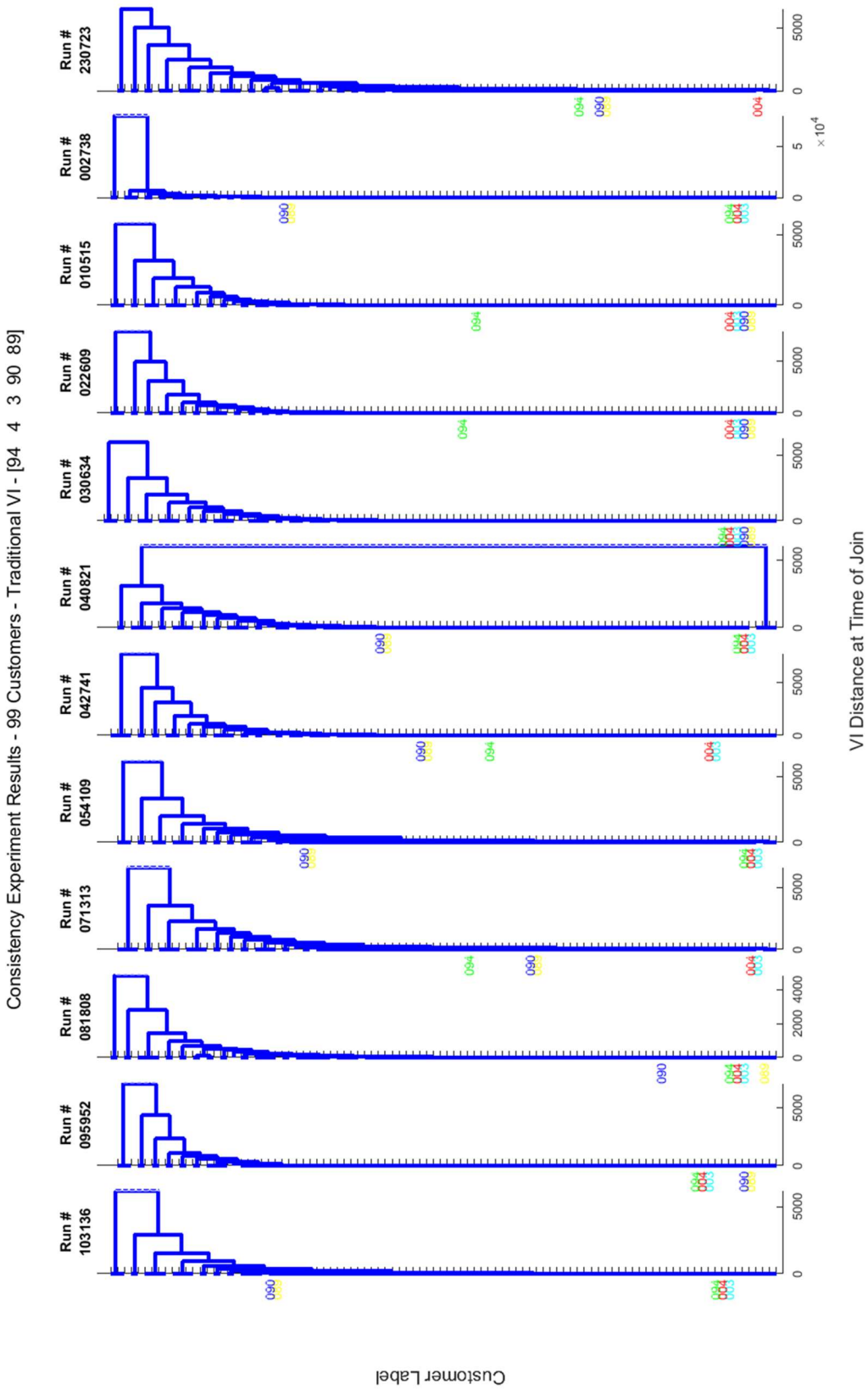

Figure 4.30 Consistency experiment results 19 of 20

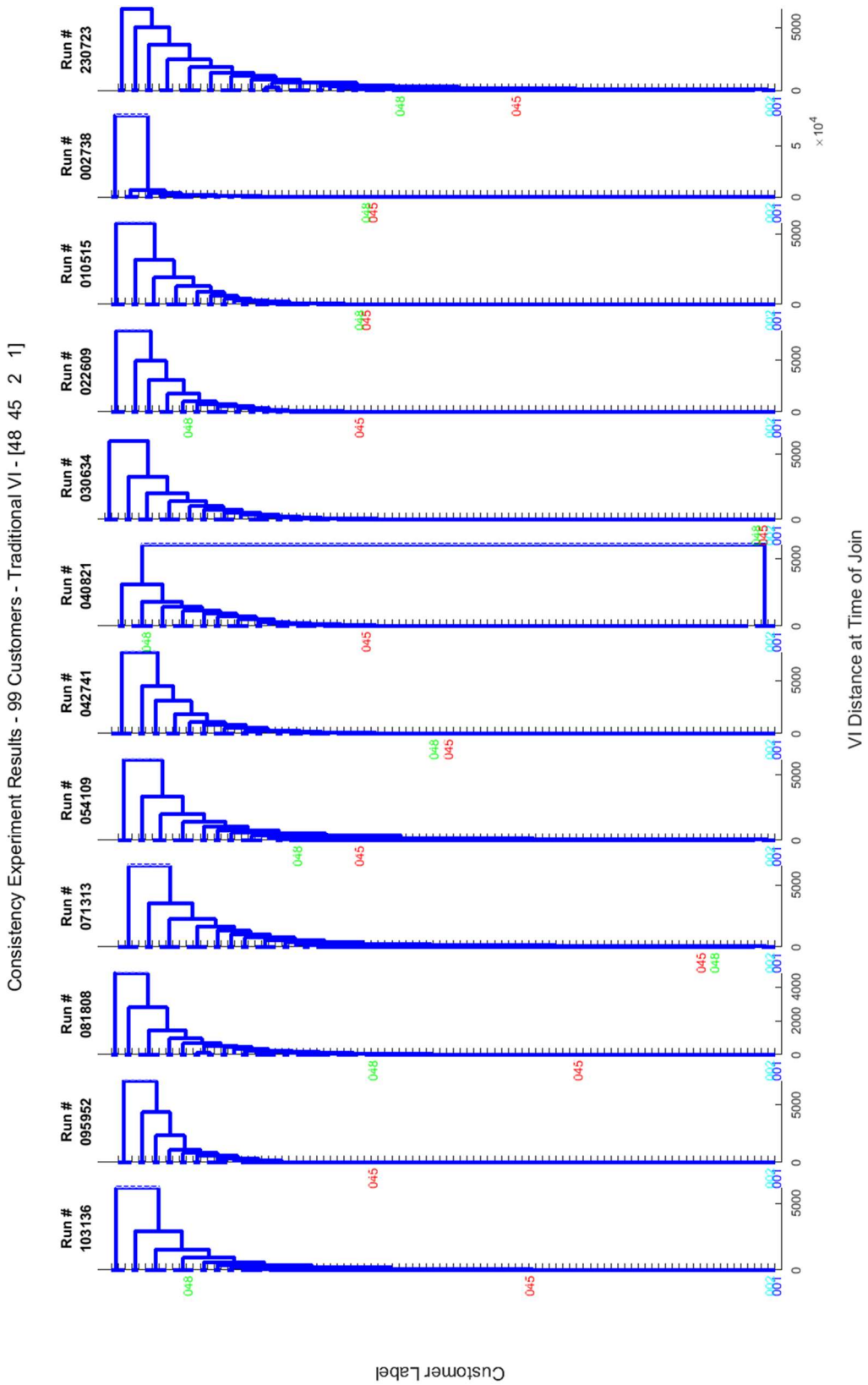

Figure 4.31 Consistency experiment results 20 of 20

4.3.3 Discussion of Consistency Experiment Results

As shown in Section 4.3.3, the clustering of many customers occurs similarly regardless of the trial. Figure 4.32 illustrates several of the most consistently placed customers in the dendrogram. These customers have lower volatility when comparing multiple trials of the same experiment. The utility can expect these customers to be classified in the same manner most of the time.

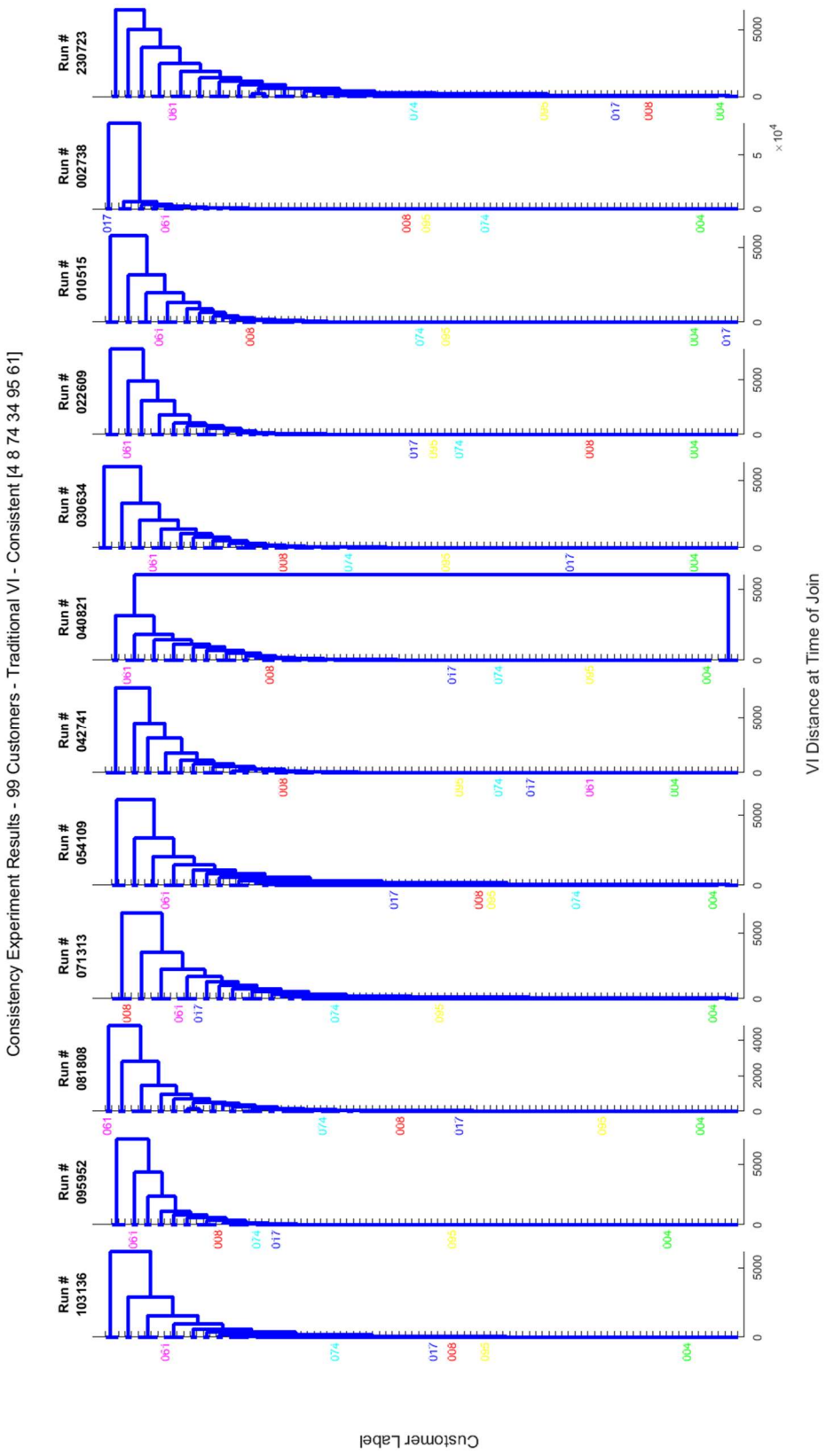

Figure 4.32 Consistency experiment results - consistent customers

A subset of customers, however, are not clustered consistently over the multiple trials. Figure 4.33 highlights those customers with the most inconsistent clustering results. These customers exhibit high volatility in their classification. As the utility runs the clustering algorithm multiple times, there is no expectation these customers will remain in the same groups. This volatility will increase the burden on the utility to understand the behaviors of these customers in-depth.

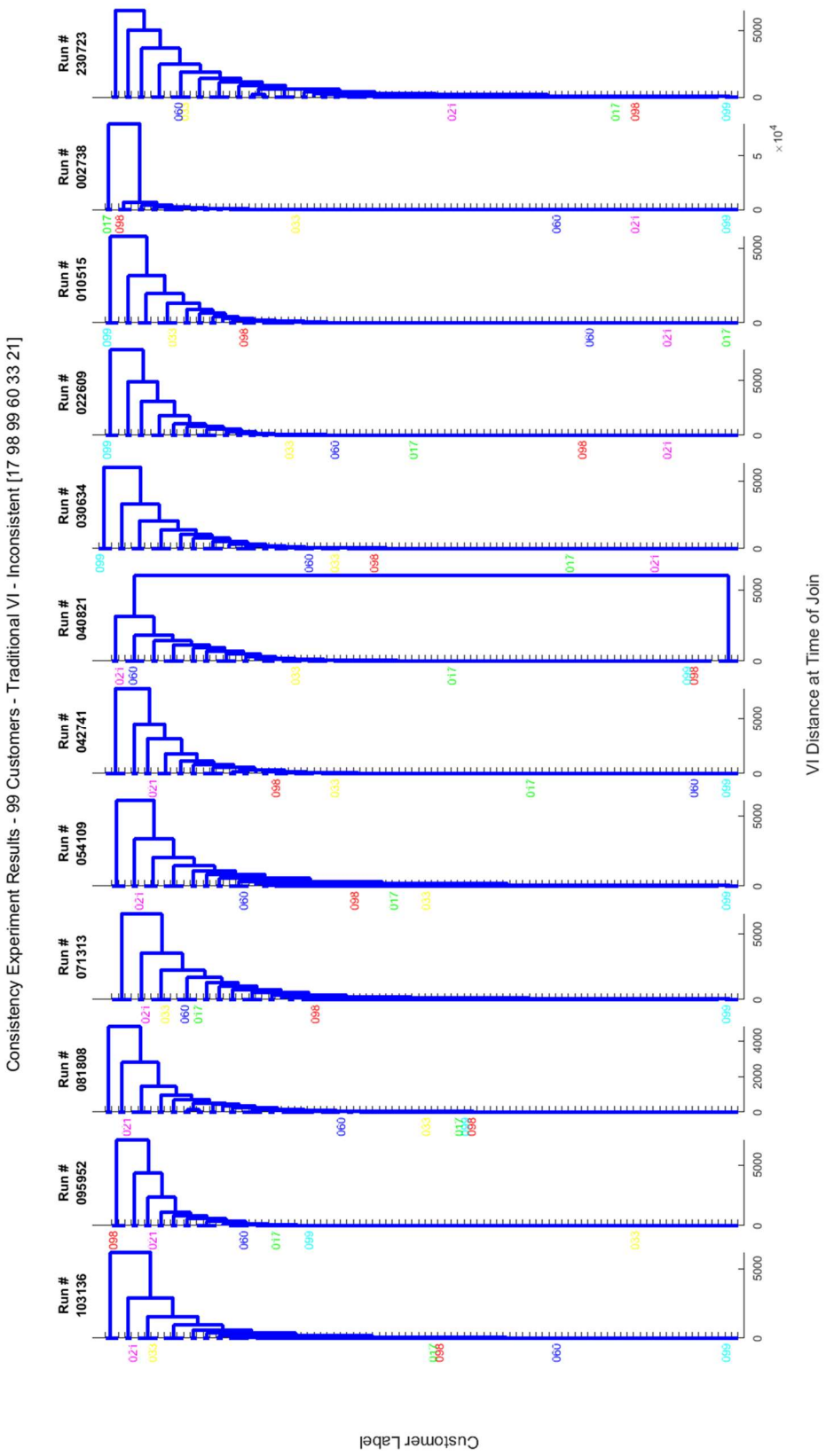

Figure 4.33 Consistency experiment results - inconsistent customers

This inconsistency will lead to certain customers being grouped differently depending on the particular run of the clustering experiment. Inconsistency is an undesirable result for the utility and costs additional time or money to investigate. Rather than using the traditional VI distance to cluster customer models, a novel component-weighting scheme has been developed to improve the consistency of this clustering process. The next chapter describes the foundation and mathematical theory of the weighted VI in detail and presents the results of consistency experiments on the same data using the new distance measure.

5 NOVEL DISTANCE MEASURE BASED ON WEIGHTED GAUSSIAN MIXTURE MODEL COMPONENTS

As the previous chapter illustrated, the results from the consistency testing were mixed. Some customers had consistent behavior in the clustering between multiple trials, while others showed a large volatility in clustering behavior. To combat this variation in results, this chapter presents a novel component-weighting scheme, discounting behaviors with large variation and preferring behaviors with highly repeatable patterns. For reference, Figure 5.1 outlines the entire process of methods and experiments used in this research. This chapter defines a weighted variation of information (wVI) distance for clustering probabilistic models as an improvement over the traditional variation of information (VI) distance used in the previous chapter. The results using the wVI distance are far more consistent across many trials for every meter in the data set.

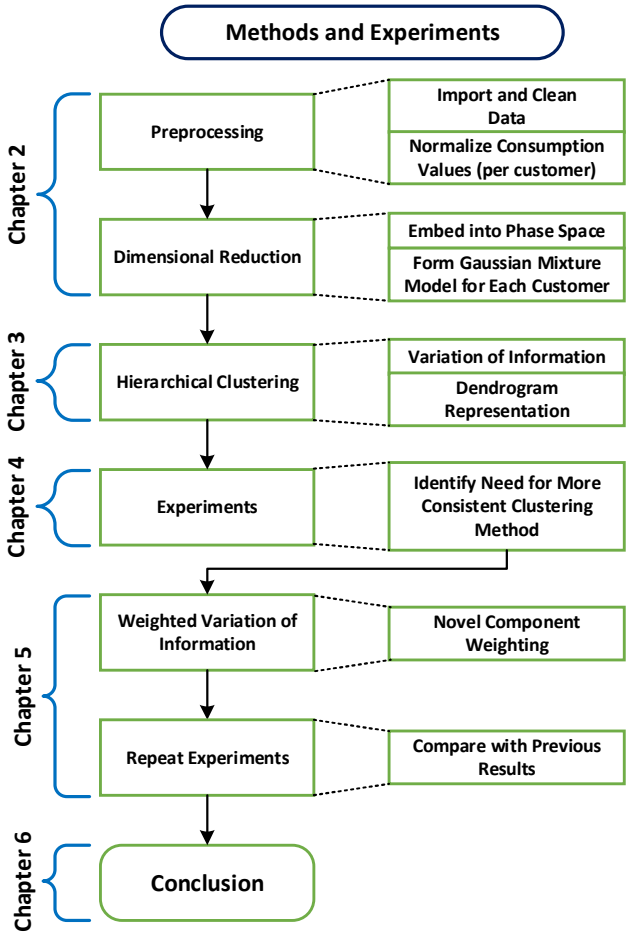

Figure 5.1 Flow diagram of the methods and experiments in this research

The experiments in Chapter 4 apply equal weight to every component within a Gaussian Mixture Model (GMM). When combining distributions, uniform weighting is not necessarily the most accurate and repeatable choice. Expanding upon work combining probabilistic forecasts in [105]–[107], a similar method is developed to weight the individual components of the Gaussian Mixture Models. For clarity, the weighting method is described in one dimension first, and then generalized to multiple dimensions for application to this research.

5.1 Accuracy and Precision of a Gaussian Distribution

The terms accuracy and precision are often treated interchangeably, but they describe two different measurements [108]. The slingshot and target shown in Figure 5.2 shows the

relationship between a Gaussian distribution fit along one dimension and the accompanying strikes on the target. The grouping of strikes shows a high accuracy, with the grouping centered on the x-ring (or bullseye) within the target, but a low precision, with a wide cluster of strikes.

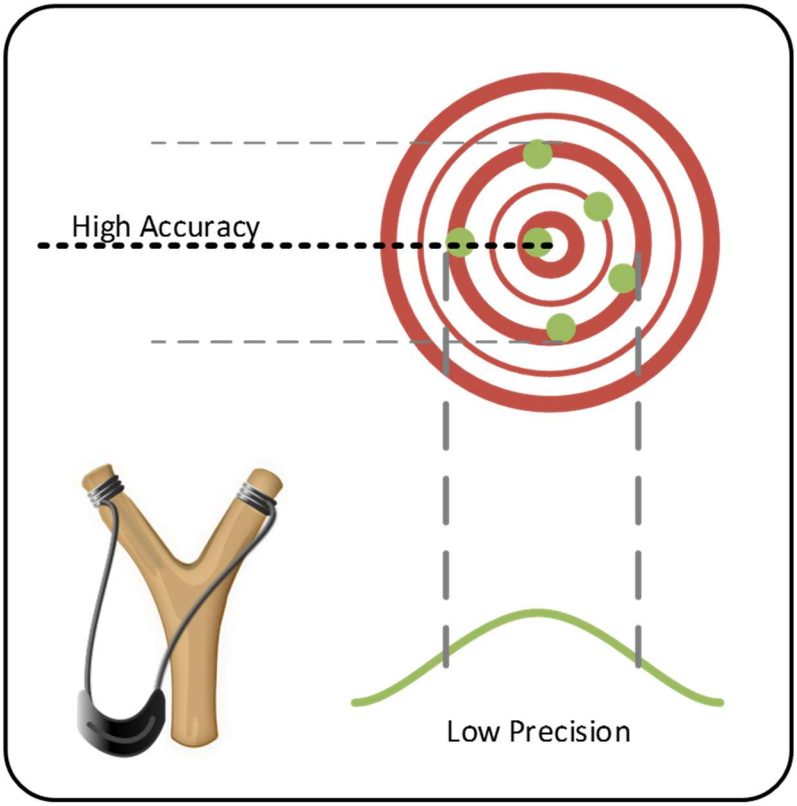

Figure 5.2 Comparing accuracy and precision of strikes to a Gaussian distribution fit on the target

With two different groupings of strikes, shown in Figure 5.3, the differences in accuracy and precision are apparent. The left grouping shows a tight cluster with a high precision but low accuracy. The right grouping shows a wide cluster with a high accuracy but low precision. When evaluating these groupings, the tight cluster shows a more repeatable performance, even though the accuracy is not centered on the target. While the grouping on the right is more accurate, centered on the x-ring of the target, any particular strike is less likely to provide useful information about the rest of the strikes in the cluster, or the ability of the individual to perform consistently [108].

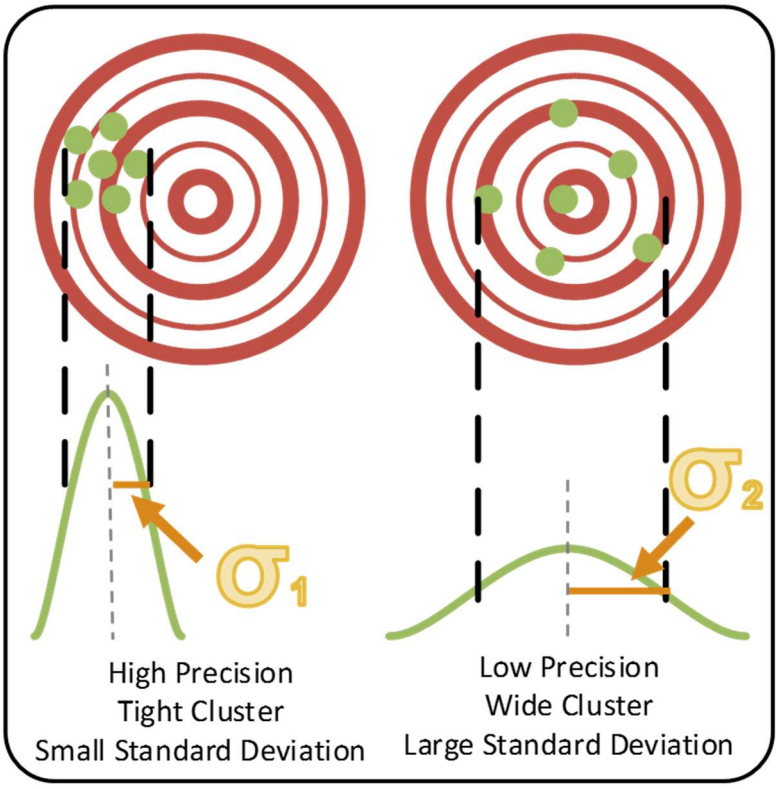

Figure 5.3 Two different Gaussian distributions fit to strikes on targets

5.2 Gaussian Mixture Models of Water Consumption Patterns

A two-component Gaussian mixture model of a daily measured flow pattern is shown in Figure 5.4. The model is fit to weekday flow measurements from a residential water meter. It is representative of many household daily flow measurement patterns with a narrow morning "before work" peak indicative of a short time between the alarm clock and departure. The wide afternoon/evening "after work" peak indicates a more flexible afternoon/evening water consumption pattern based on something other than a fixed deadline such as the beginning of the workday. Other patterns exist, but this pattern and variations of it are pervasive in residential data and are useful to discuss the limitations of using traditional variation of information as a clustering distance.

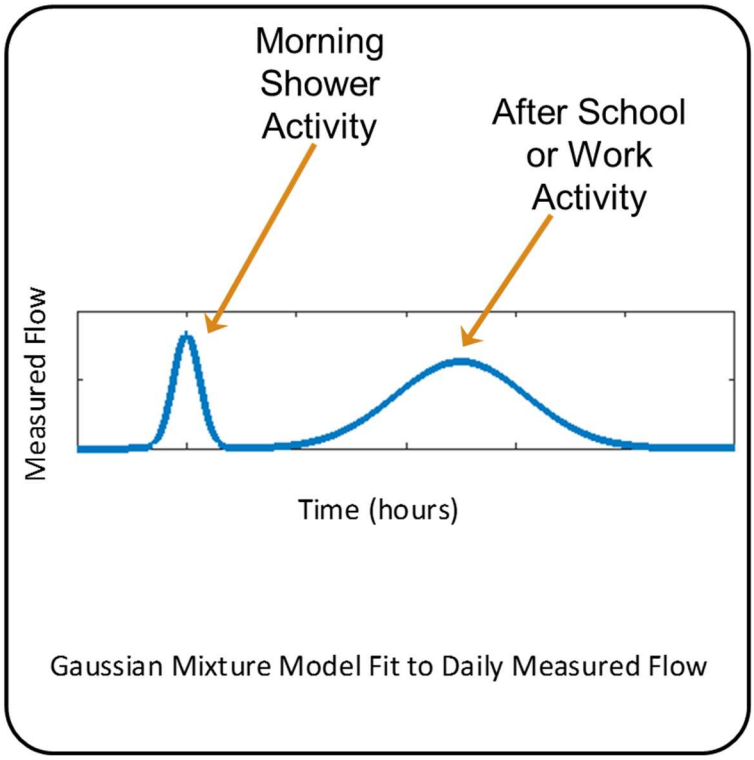

Figure 5.4 Gaussian mixture model fit to residential weekday measured flow

Patterns like the one in Figure 5.4 are seen throughout the data. Figure 5.5 shows three such customers represented by three different Gaussian mixture models, each with two Gaussian components. For the purposes of this research, the customers A and B are the most similar, having a narrow peak in the morning indicative of a highly repeatable behavior. The wide peak in the afternoon shows significant total consumption, but lacks the repeatability of a habitual behavior, creating the large standard deviation shown in the mixture models. Comparing customers A and C, the large peak in the afternoon is very similar, but the morning peaks appear at different times. This can skew the traditional variation of information (VI) distance calculation in favor of grouping AC prior to grouping AB.

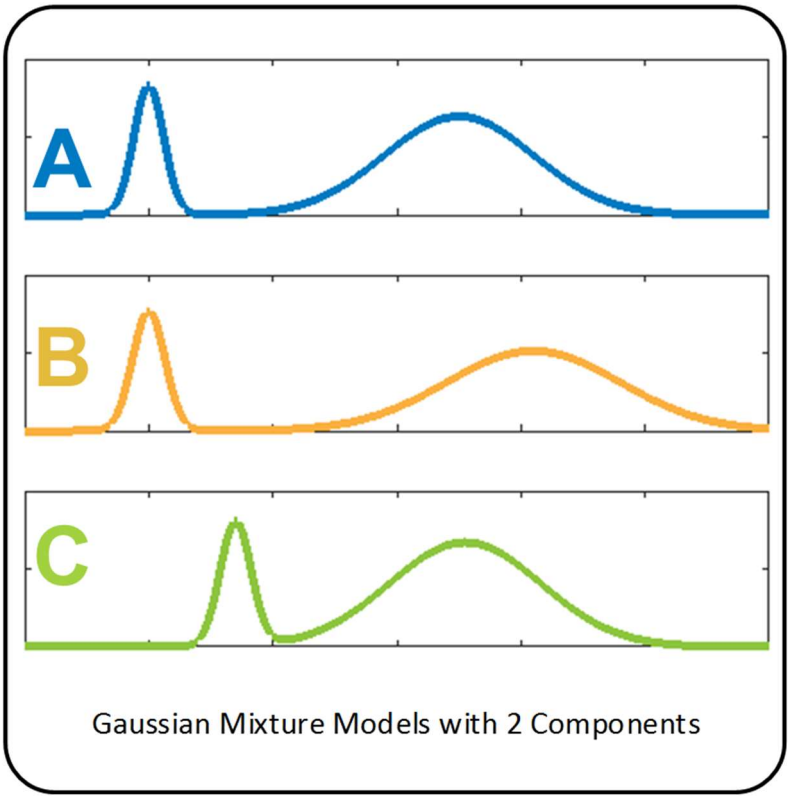

Figure 5.5 Three customers represented as 2-component GMMs

To illustrate the different clustering results, Figure 5.6 shows each possible clustering with the traditional VI area shaded in red. While not immediately obvious, the mathematical value of VI for the third clustering, AC, is the smallest of the three, and would be chosen for a traditional VI based clustering. A distribution with a wide standard deviation shows less precision of the mean time value represented by the model component. Thus, the standard deviation of the distribution for each component brings important information regarding the precision of the measurements to the weighting of a multi-component model.

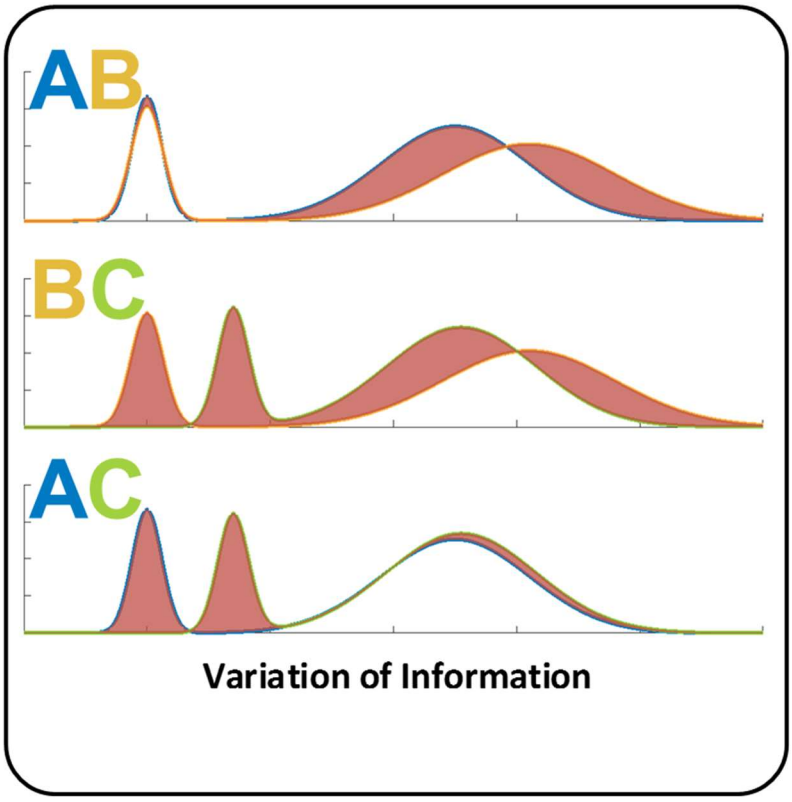

Figure 5.6 Three possible clusterings of the customers A, B, and C, with shaded area indicating variation of information

5.3 Component Weighting of the Gaussian Mixture Models

Adding component weighting to the VI computation provides the desired results. Each component in the mixture model has its own standard deviation, as illustrated in Figure 5.7. These standard deviations are used to compute the weights for every component in each customer GMM [109].

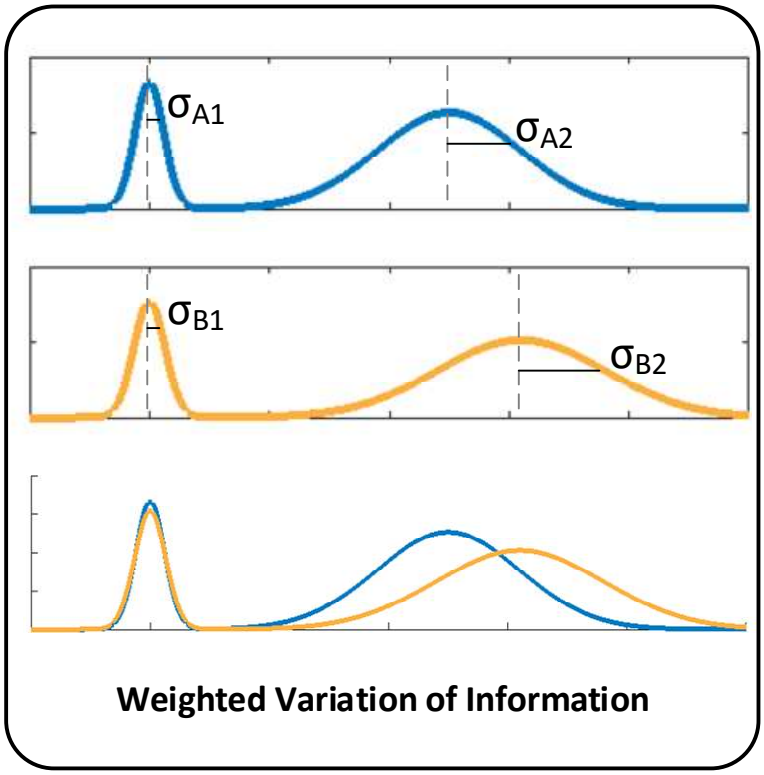

Figure 5.7 Component standard deviations used for computation of weighted variation of information

The relationship between the weights and the standard deviation of components is

$$w_{A1} = \frac{\frac{1}{\sigma_{A1}}}{\frac{1}{\sigma_{A1}} + \frac{1}{\sigma_{A2}}} \ . \tag{5.1}$$

Then, these weights are applied to the different entropies of the components, and the weighted component entropies summed to determine the overall entropy of the customer

$$wH(A) = H(A_1)w_{A1} + H(A_2)w_{A2} \ . \tag{5.2}$$

After the remaining weights and entropies are computed for every customer, the weighted mutual information and weighted variation of information between two customers are computed using

$$wMI(A,B) = MI(A_1,B_1)w_{A1}w_{B1} + MI(A_1,B_2)w_{A1}w_{B2} + MI(A_2,B_1)w_{A2}w_{B1} + MI(A_2,B_2)w_{A2}w_{B2} \quad (5.3)$$

and

$$wVI(A,B) = wH(A) + wH(B) - 2\left[wMI(A,B)\right]. \quad (5.4)$$

In the case illustrated in Figure 5.7, the components of the GMM are cleanly separated, and the products of $MI(A_1,B_2)w_{A1}w_{B2}$ and $MI(A_2,B_1)w_{A2}w_{B1}$ are small or zero, but this is not always the case when comparing two GMMs

5.4 Weighting of Models with Varying Number of Components

The different customers need not have the same number of components in the GMM, although the same dimension is required for comparison. Figure 5.8 demonstrates comparing customers with different numbers of components and includes the computed VI and wVI measurements.

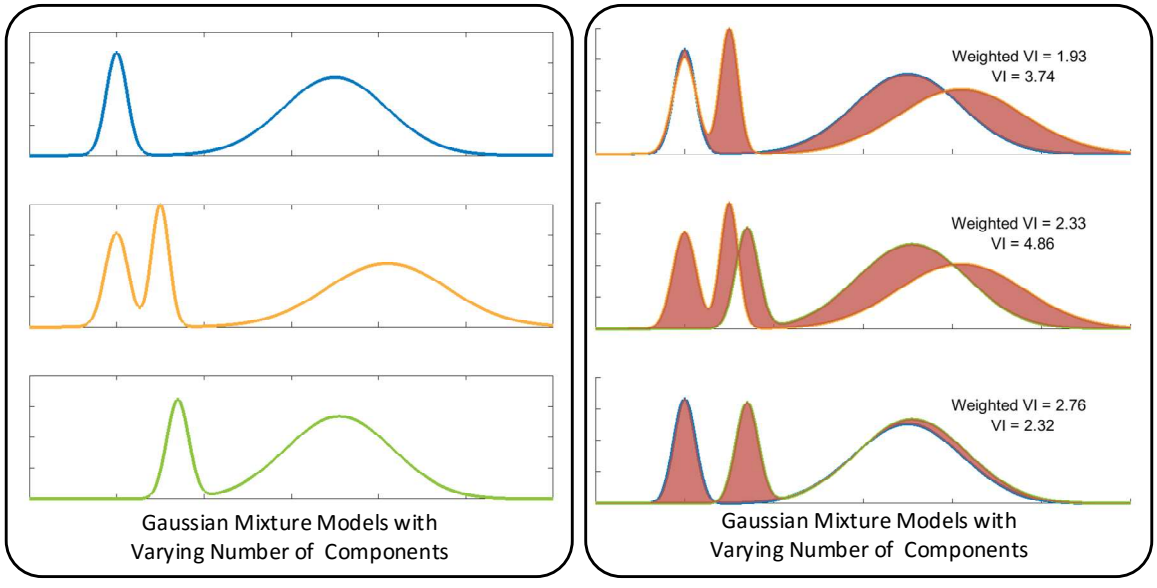

Figure 5.8 Comparing the VI and weighted VI of customers with varying number of components in the GMM

This example has been reduced in dimension for clarity and ease of visual illustration, but the procedure can be extrapolated into an n-dimensional space with any number of components to the GMM. For the current research using reconstructed phase space of water flow measurements, the arithmetic mean of standard deviation in the three axial directions is used to create an average standard deviation for the component,

$$\sigma_{1avg} = \frac{\sigma_{1x} + \sigma_{1y} + \sigma_{1z}}{3}. \tag{5.5}$$

These could be weighted individually if one dimension is more important than others in the underlying model. All dimensions are equally weighted in this work. Next, compute the weight relative to the standard deviations of every component within the model:

$$w_1 = \frac{\frac{1}{\sigma_{1avg}}}{\frac{1}{\sigma_{1avg}} + \frac{1}{\sigma_{2avg}} + \frac{1}{\sigma_{3avg}} + \frac{1}{\sigma_{4avg}}}. \tag{5.6}$$

Necessarily, the sum of all weights of all components of a model is unity,

$$\sum_{i=1}^{n} w_i = 1. \tag{5.7}$$

Finally, apply these weights when computing the entropy, MI, and VI distance between two cluster models (cluster A and cluster B)

$$H(A) = H(A_1)w_{A1} + H(A_2)w_{A2} + H(A_3)w_{A3} + H(A_4)w_{A4} = \sum_{i=1}^{n} H(A_i)w_{Ai}, \tag{5.8}$$

$$H(B) = H(B_1)w_{B1} + H(B_2)w_{B2} + H(B_3)w_{B3} + H(B_4)w_{B4} = \sum_{k=1}^{m} H(B_k)w_{Bk}, \tag{5.9}$$

mutual information

$$MI\left(A,B\right)=\sum_{k=1}^{m}\sum_{i=1}^{n}MI\left(A_i,B_k\right)w_{Ai}w_{Bk}\,, \tag{5.10}$$

and variation of information distance

$$\begin{aligned} &wVI\left(A,B\right)=wH\left(A\right)+wH\left(B\right)-2\left[wMI\left(A,B\right)\right], \text{ or} \\ &wVI\left(A,B\right)=\sum_{i=1}^{n}H\left(A_i\right)w_{Ai}+\sum_{k=1}^{m}H\left(B_k\right)w_{Bk}-2\left(\sum_{k=1}^{m}\sum_{i=1}^{n}MI\left(A_i,B_k\right)w_{Ai}w_{Bk}\right), \end{aligned} \tag{5.11}$$

between two cluster models (cluster A and cluster B). Weight w_{A1} represents the weight of mixture component 1 for cluster A. This weighting allows the clustering algorithm to emphasize smaller cluster components, those with a tighter standard deviation and reduces emphasis of the components with a large average standard deviation.

5.5 Comparing Weighted Variation of Information with the Traditional Variation of Information

In some circumstances, the weighted variation of information will be equivalent to the traditional variation of information. For Gaussian mixture models comprised of identically shaped components, the weights will be equal. Recall the definition of wVI between two cluster models A and B,

$$wVI\left(A,B\right)=\sum_{i=1}^{n}H\left(A_i\right)w_{Ai}+\sum_{k=1}^{m}H\left(B_k\right)w_{Bk}-2\left(\sum_{k=1}^{m}\sum_{i=1}^{n}MI\left(A_i,B_k\right)w_{Ai}w_{Bk}\right), \tag{5.12}$$

with the weight of any component of A,

$$w_{Ai} = \frac{\frac{1}{\sigma_{Ai}}}{\sum_{i=1}^{n} \frac{1}{\sigma_{Ai}}}, \tag{5.13}$$

and the weight of any component of B,

$$w_{Bk} = \frac{\frac{1}{\sigma_{Bk}}}{\sum_{k=1}^{m} \frac{1}{\sigma_{Bk}}}. \tag{5.14}$$

In cases where the standard deviations of all components within the cluster (A or B) are equal, the weights of each component within the cluster are also equal. That is,

$$\begin{aligned} &w_A = w_{Ai} = \cdots = w_{An} = 1, \text{ and} \\ &w_B = w_{Bi} = \cdots = w_{Bn} = 1. \end{aligned} \tag{5.15}$$

With equal component weights, $wVI(A,B)$ is equal to $VI(A,B)$,

$$\begin{aligned} &wVI(A,B) = \\ &\quad = \sum_{i=1}^{n} H(A_i) w_{Ai} + \sum_{k=1}^{m} H(B_k) w_{Bk} - 2\left(\sum_{k=1}^{m} \sum_{i=1}^{n} MI(A_i, B_k) w_{Ai} w_{Bk} \right) \\ &\quad = \sum_{i=1}^{n} H(A_i) + \sum_{k=1}^{m} H(B_k) - 2\left(\sum_{k=1}^{m} \sum_{i=1}^{n} MI(A_i, B_k) \right) \\ &\quad = VI(A,B). \end{aligned} \tag{5.16}$$

The models with components varying minimally in standard deviation (in three dimensions this is shape and volume) will have nearly identical, but not equal weights. When a model has considerable variation in the standard deviations (shape, volume), the component weights will be substantially different. This is reflected in the computation of individual weights by Equations (5.13) and (5.14).

The wVI will represent models with varying components differently than the traditional VI measurement. The emphasis on narrower distributions (compact volumes in three dimensions), results in a smaller distance between customers that match on these highly repetitive behaviors.

5.6 Consistency Testing Using Weighted Variation of Information

The consistency tests of Section 4.3 have been repeated using the weighted variation of information (wVI) distance measure. This experiment compares multiple Gaussian mixture models from the same 99 customers, and each has the same anonymized label number as it had in results presented in Sections 4.3.2 and 4.3.3. The colorized groupings presented in each of the 20 output figures match those in the previous results as well, to aid comparisons between the experimental results.

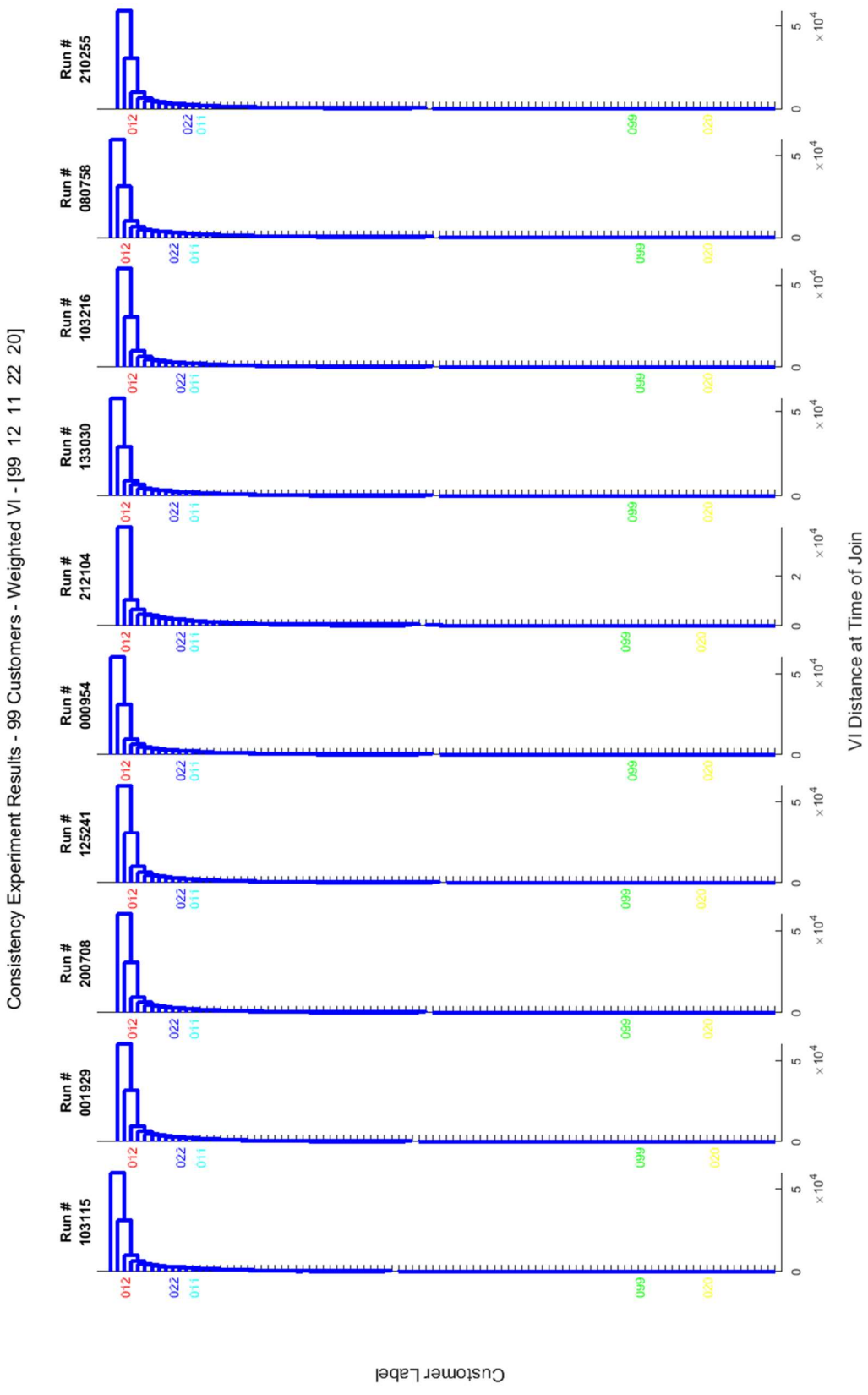

Figure 5.9 Consistency experiment results using wVI distance 1 of 20

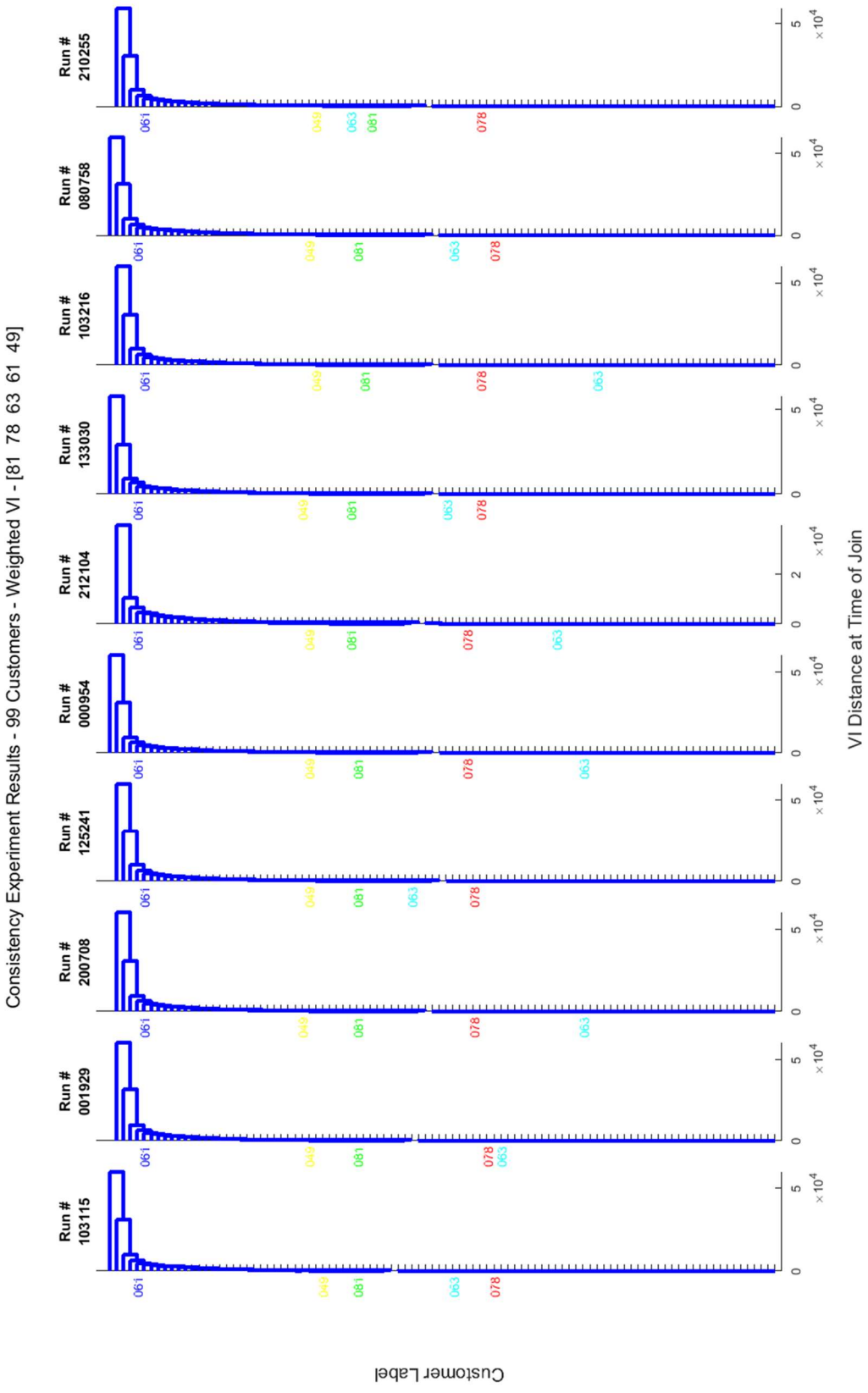

Figure 5.10 Consistency experiment results using wVI distance 2 of 20

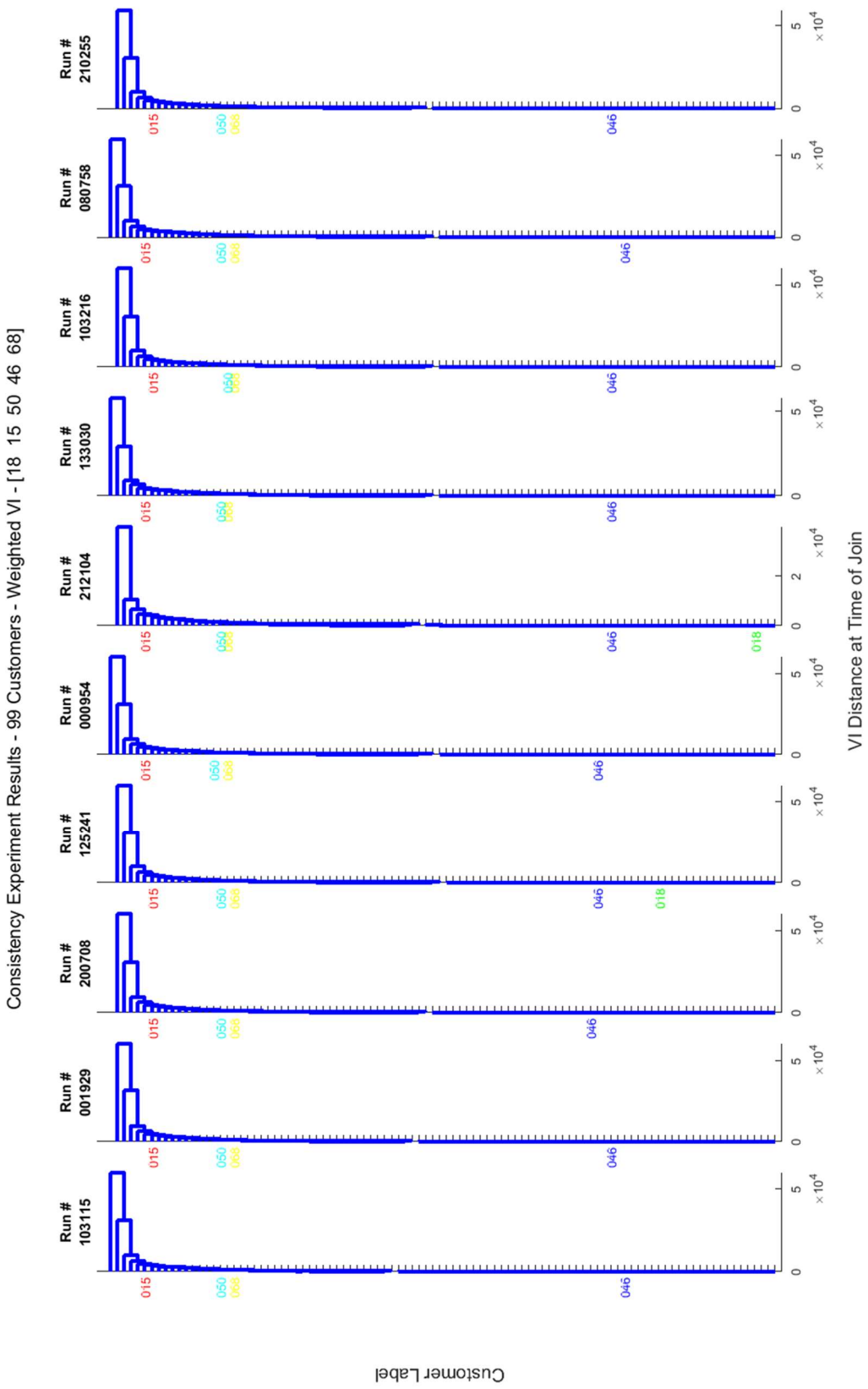

Figure 5.11 Consistency experiment results using wVI distance 3 of 20

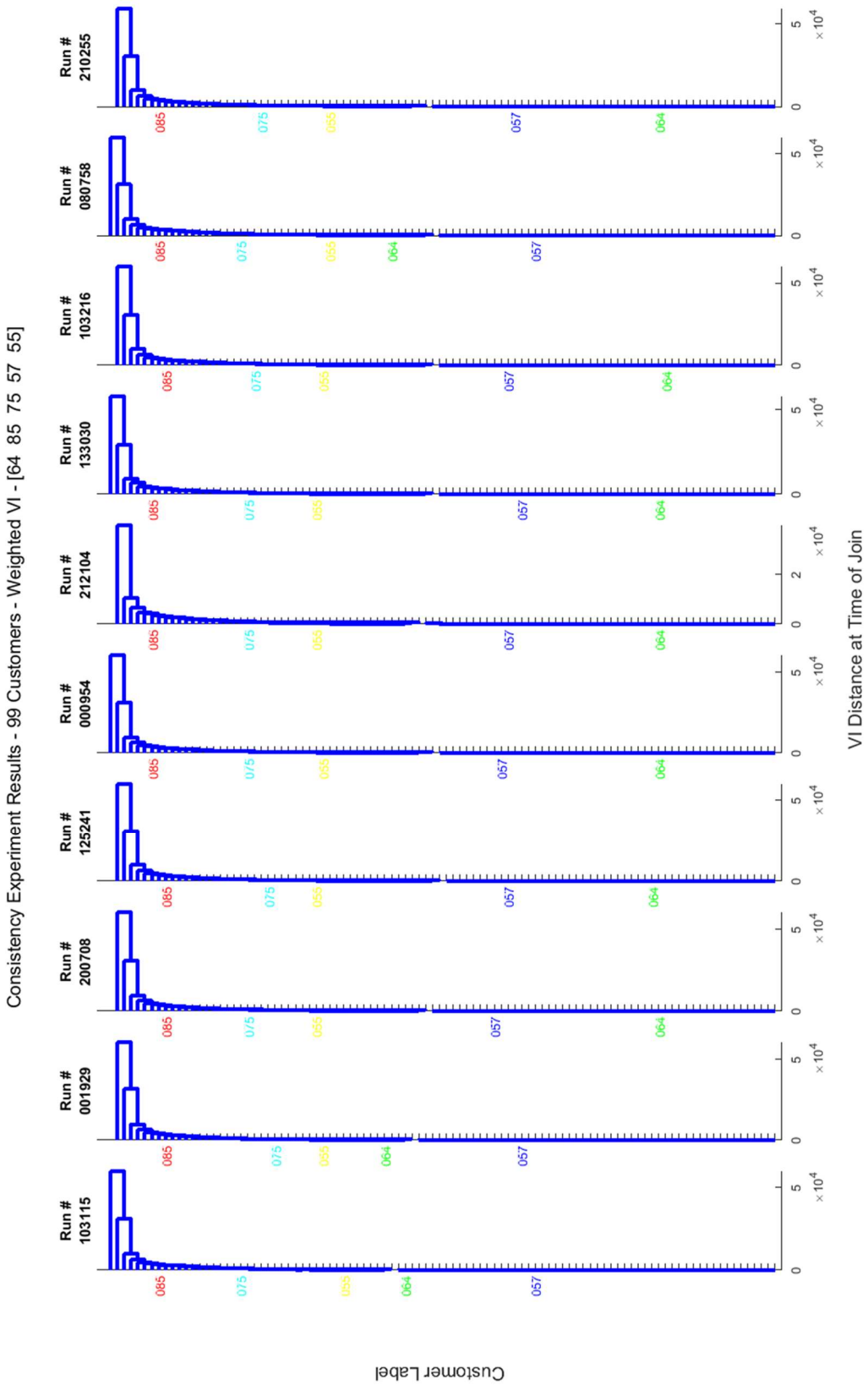

Figure 5.12 Consistency experiment results using wVI distance 4 of 20

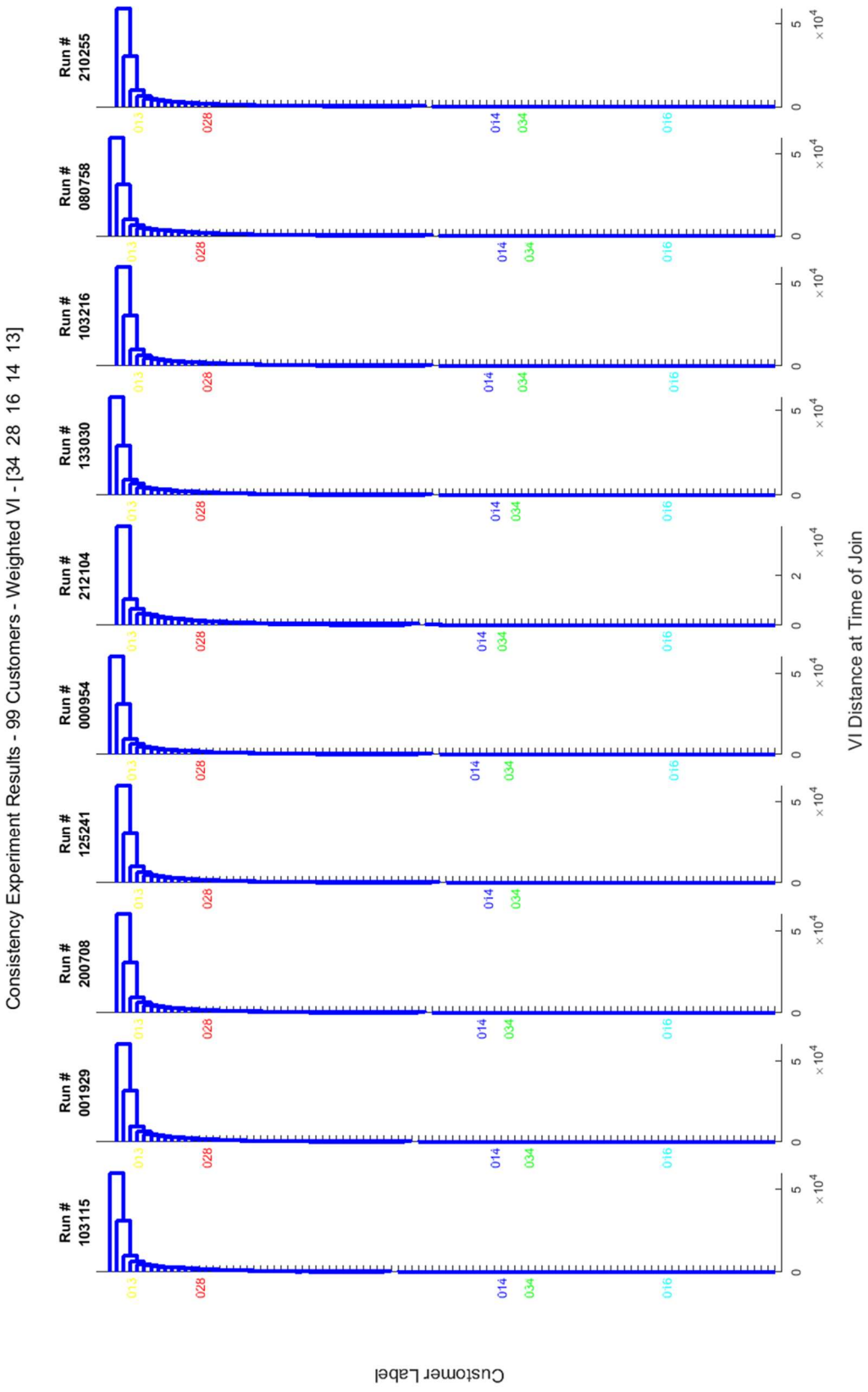

Figure 5.13 Consistency experiment results using wVI distance 5 of 20

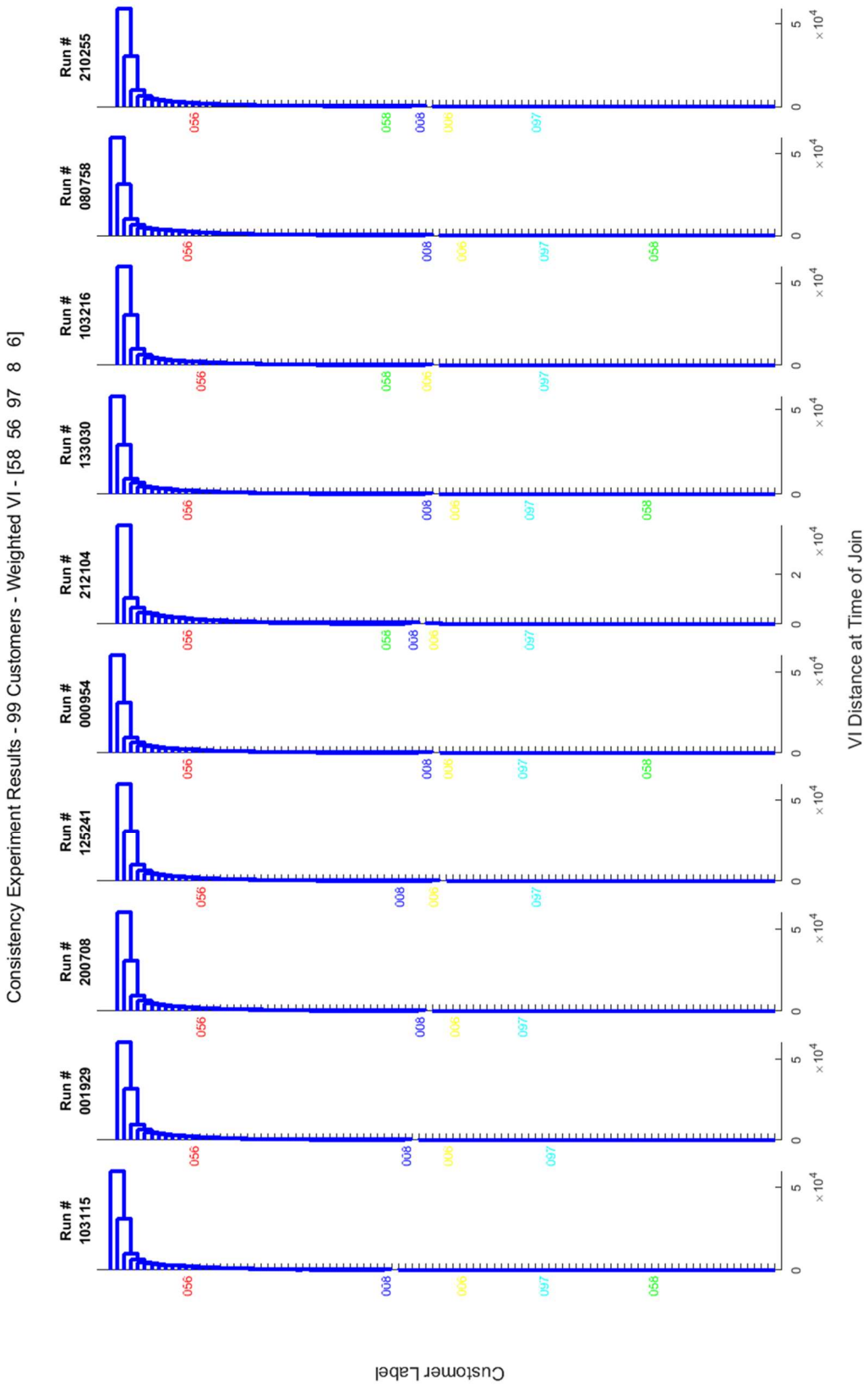

Figure 5.14 Consistency experiment results using wVI distance 6 of 20

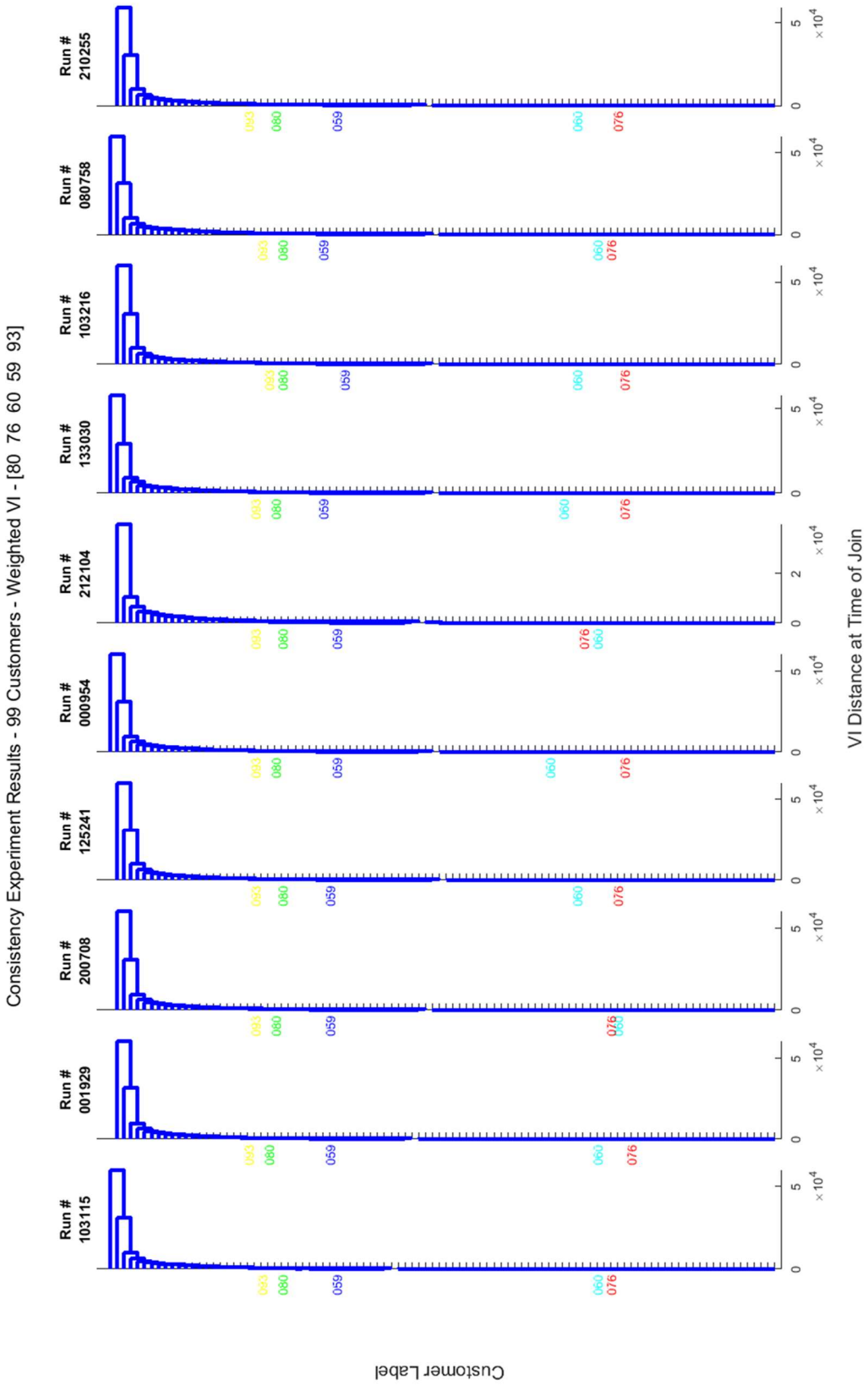

Figure 5.15 Consistency experiment results using wVI distance 7 of 20

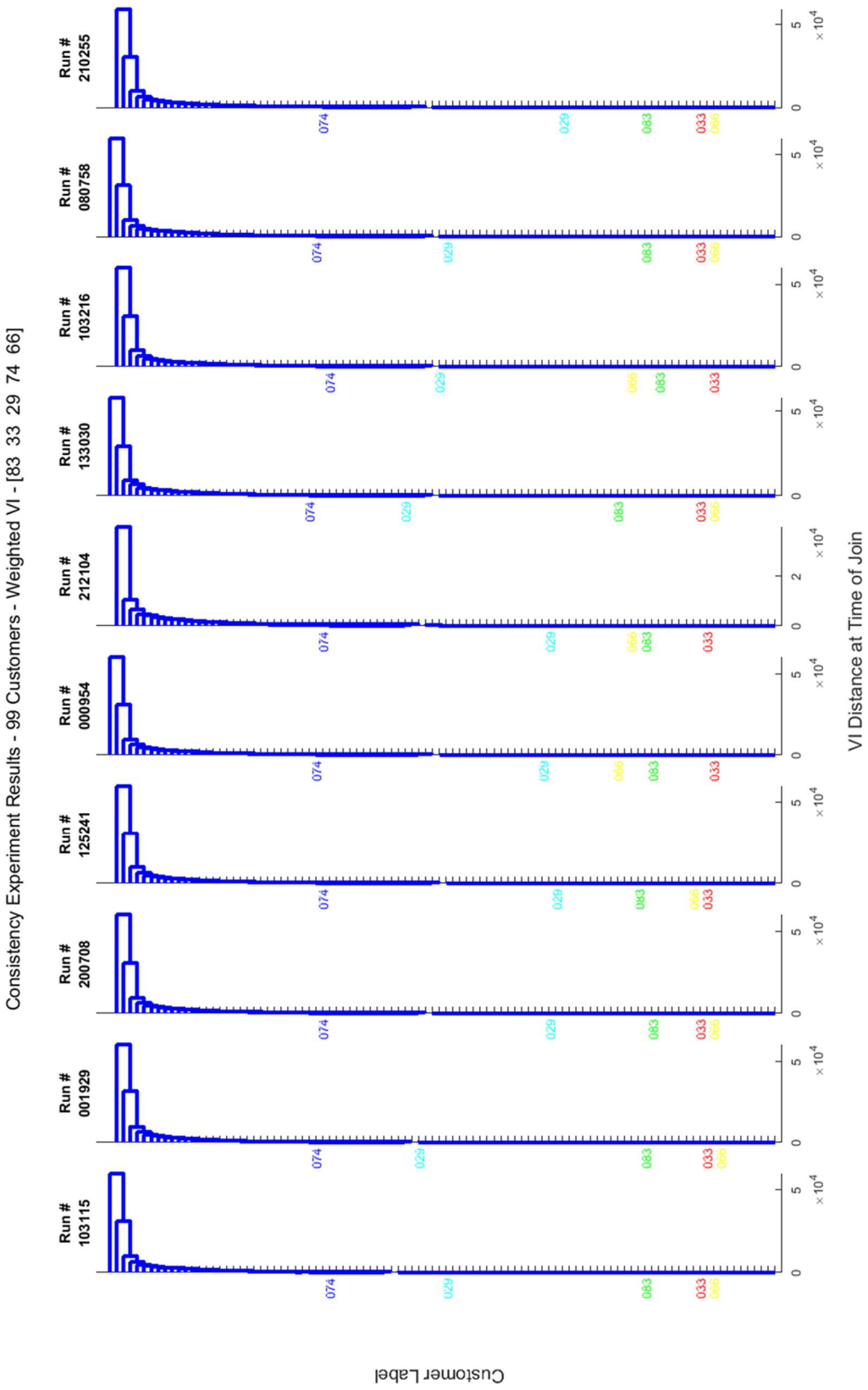

Figure 5.16 Consistency experiment results using wVI distance 8 of 20

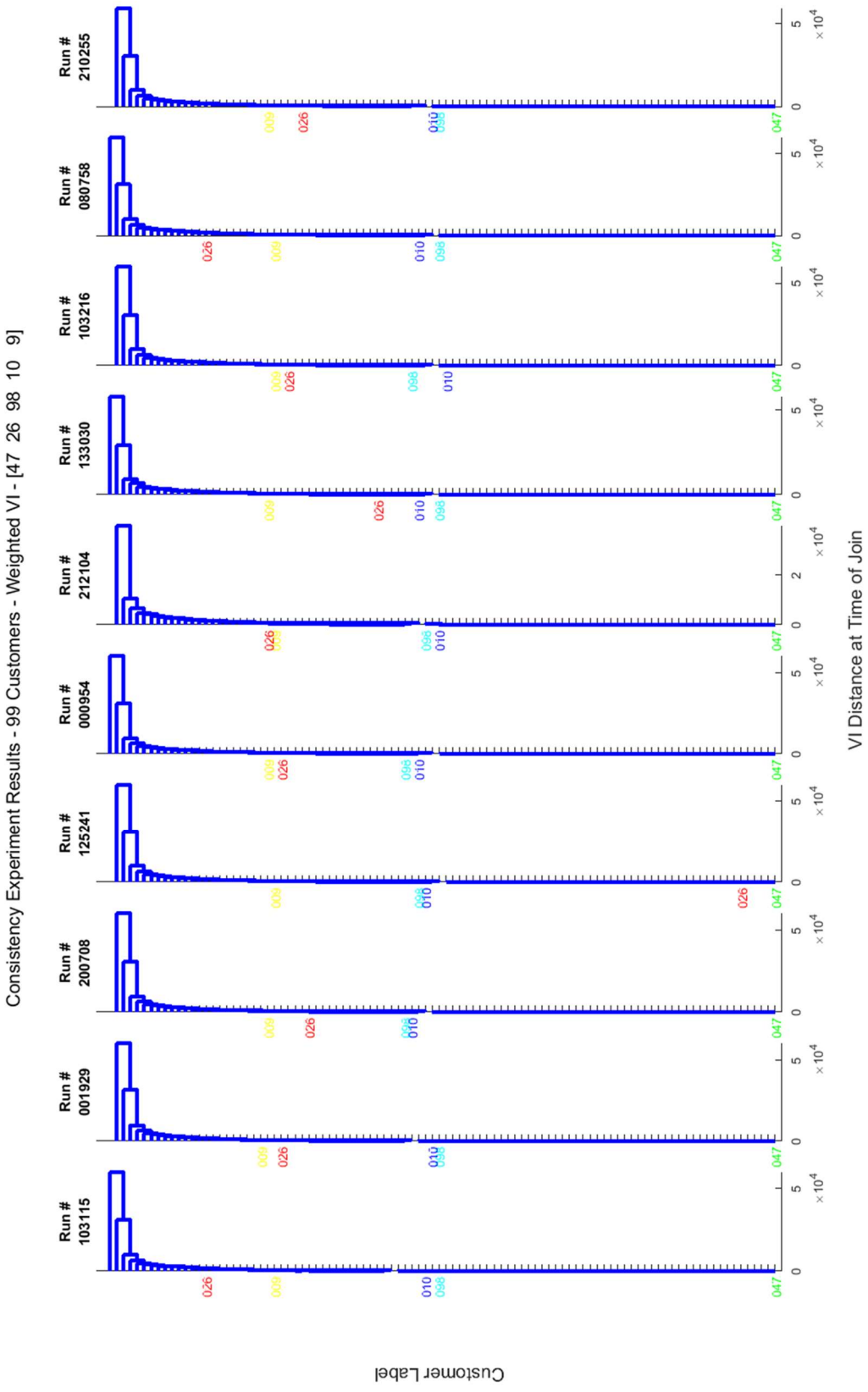

Figure 5.17 Consistency experiment results using wVI distance 9 of 20

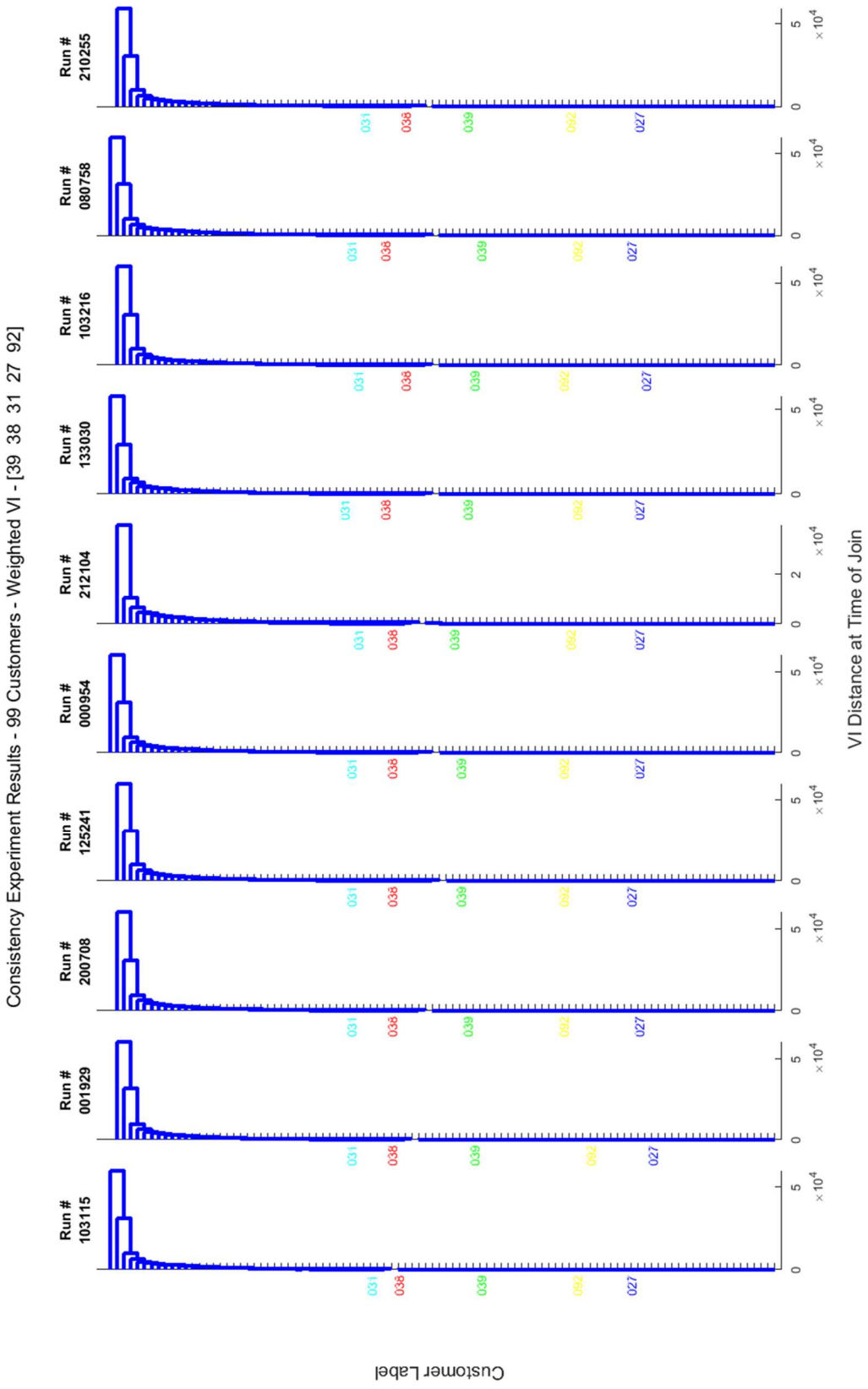

Figure 5.18 Consistency experiment results using wVI distance 10 of 20

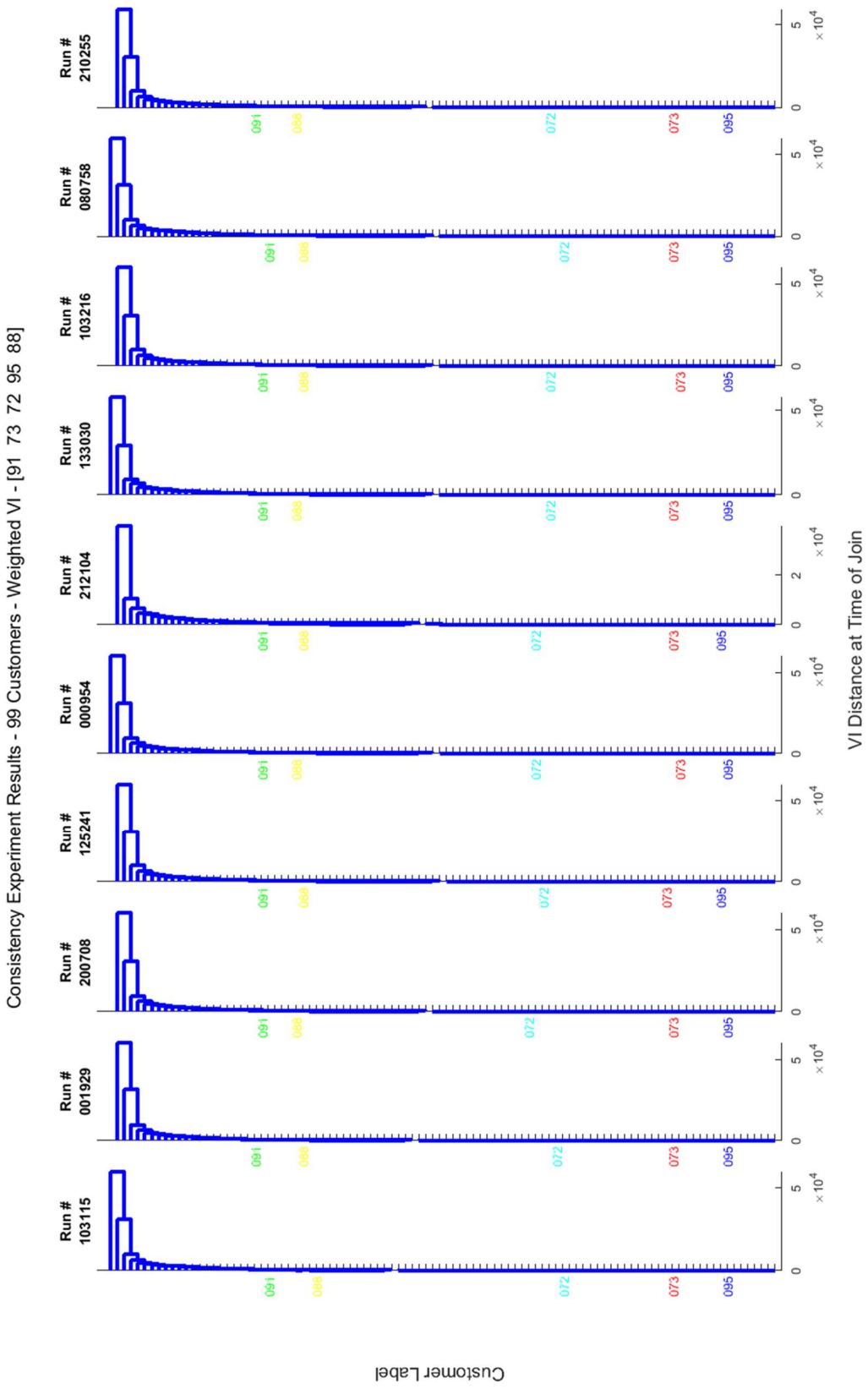

Figure 5.19 Consistency experiment results using wVI distance 11 of 20

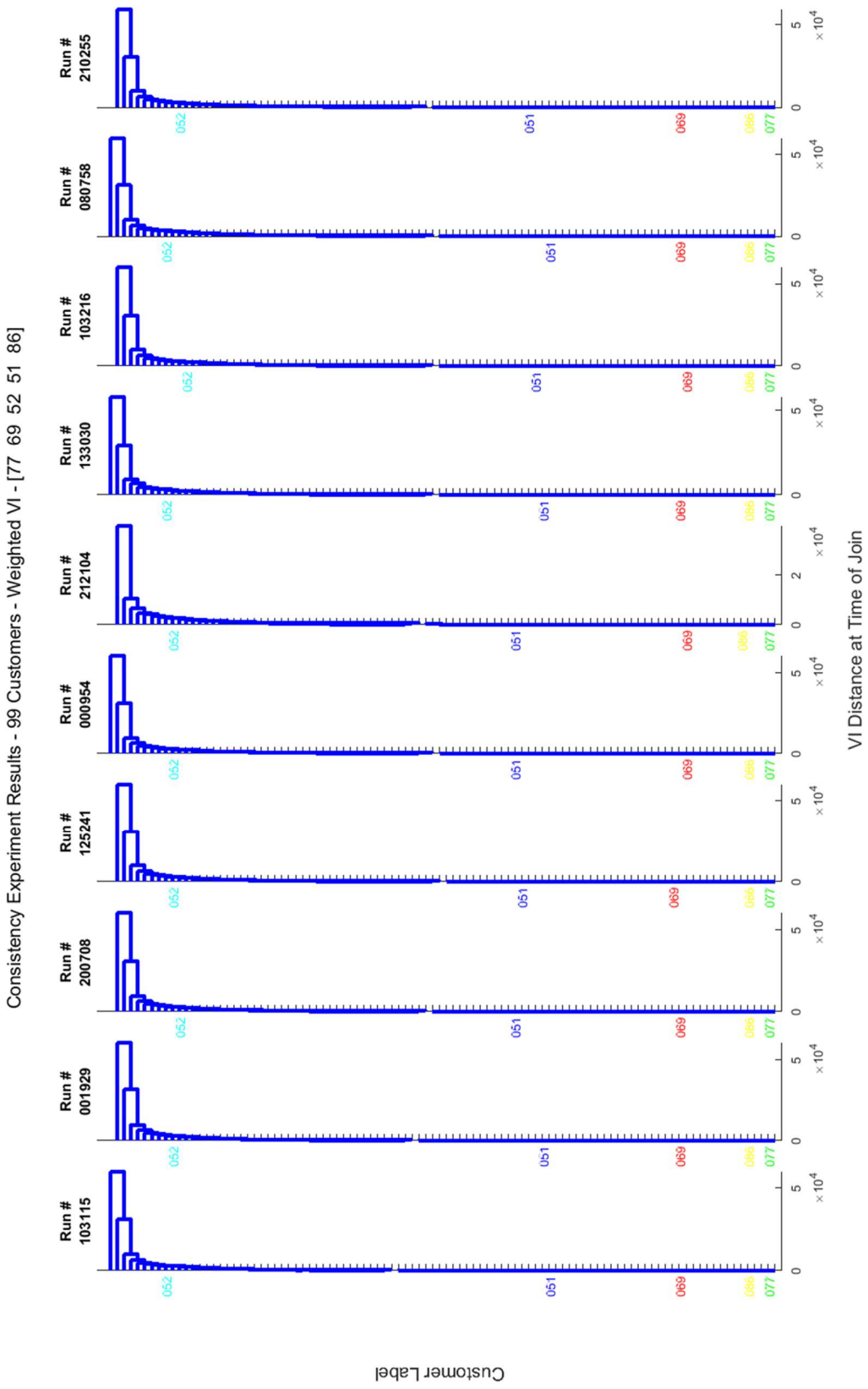

Figure 5.20 Consistency experiment results using wVI distance 12 of 20

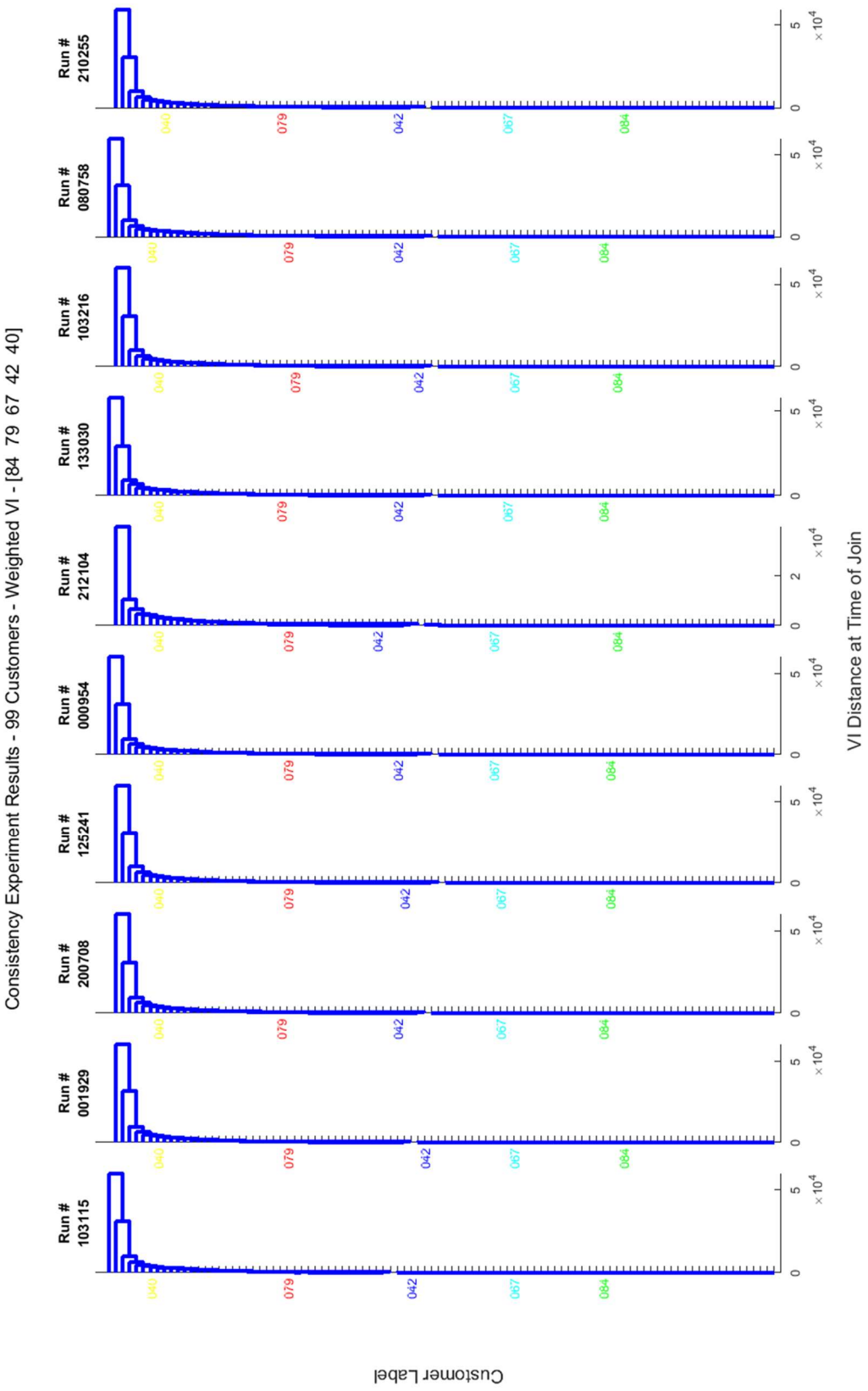

Figure 5.21 Consistency experiment results using wVI distance 13 of 20

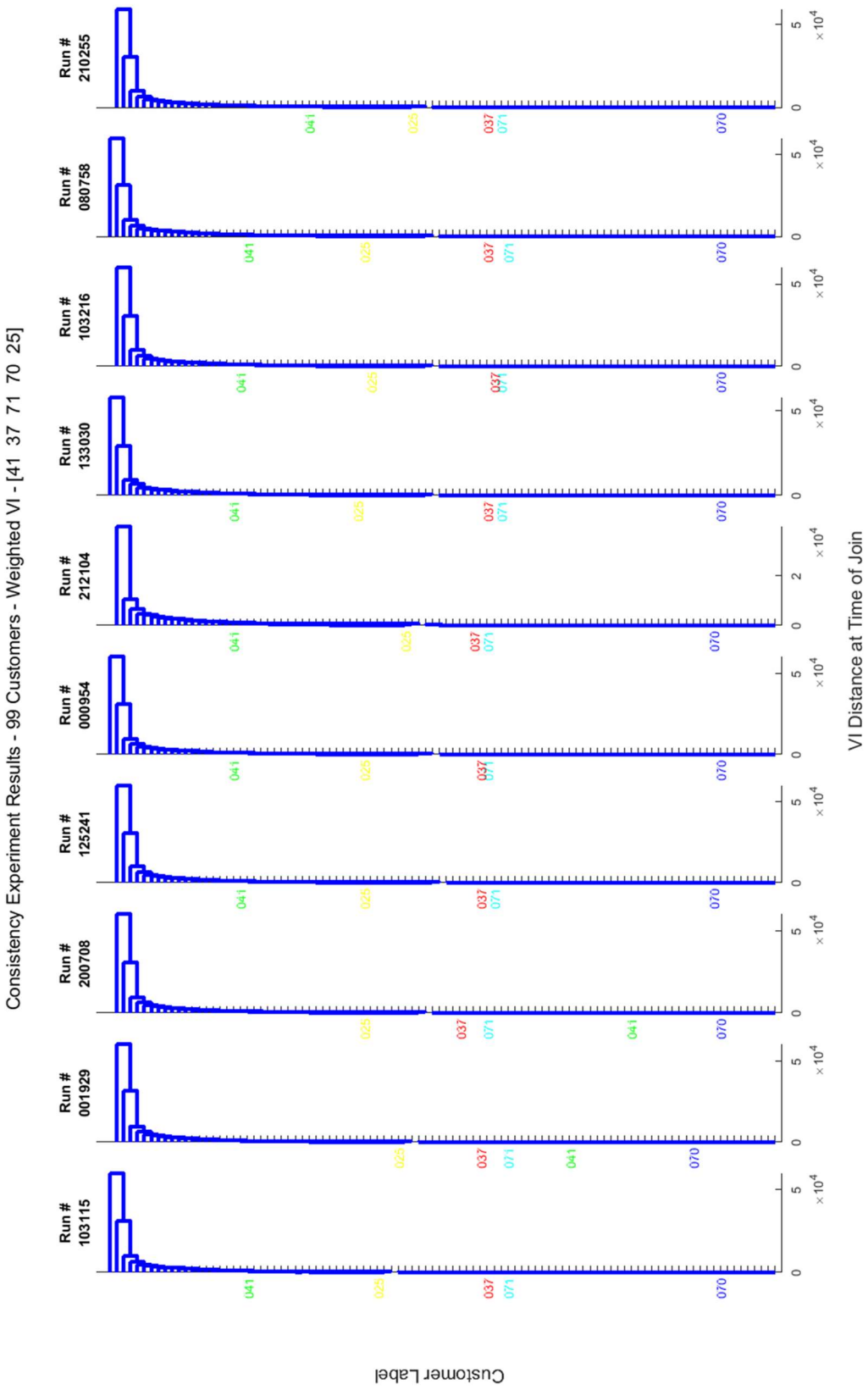

Figure 5.22 Consistency experiment results using wVI distance 14 of 20

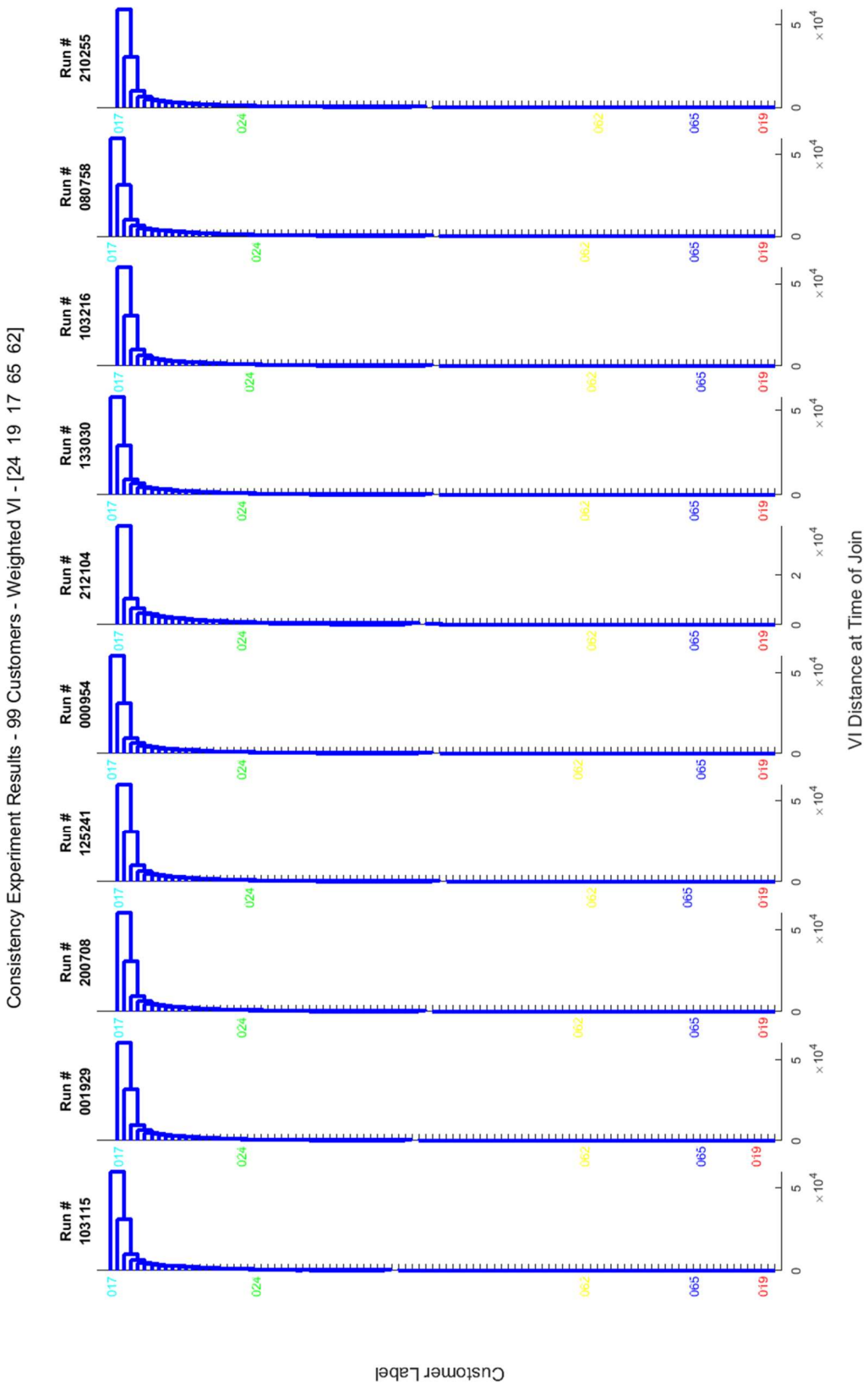

Figure 5.23 Consistency experiment results using wVI distance 15 of 20

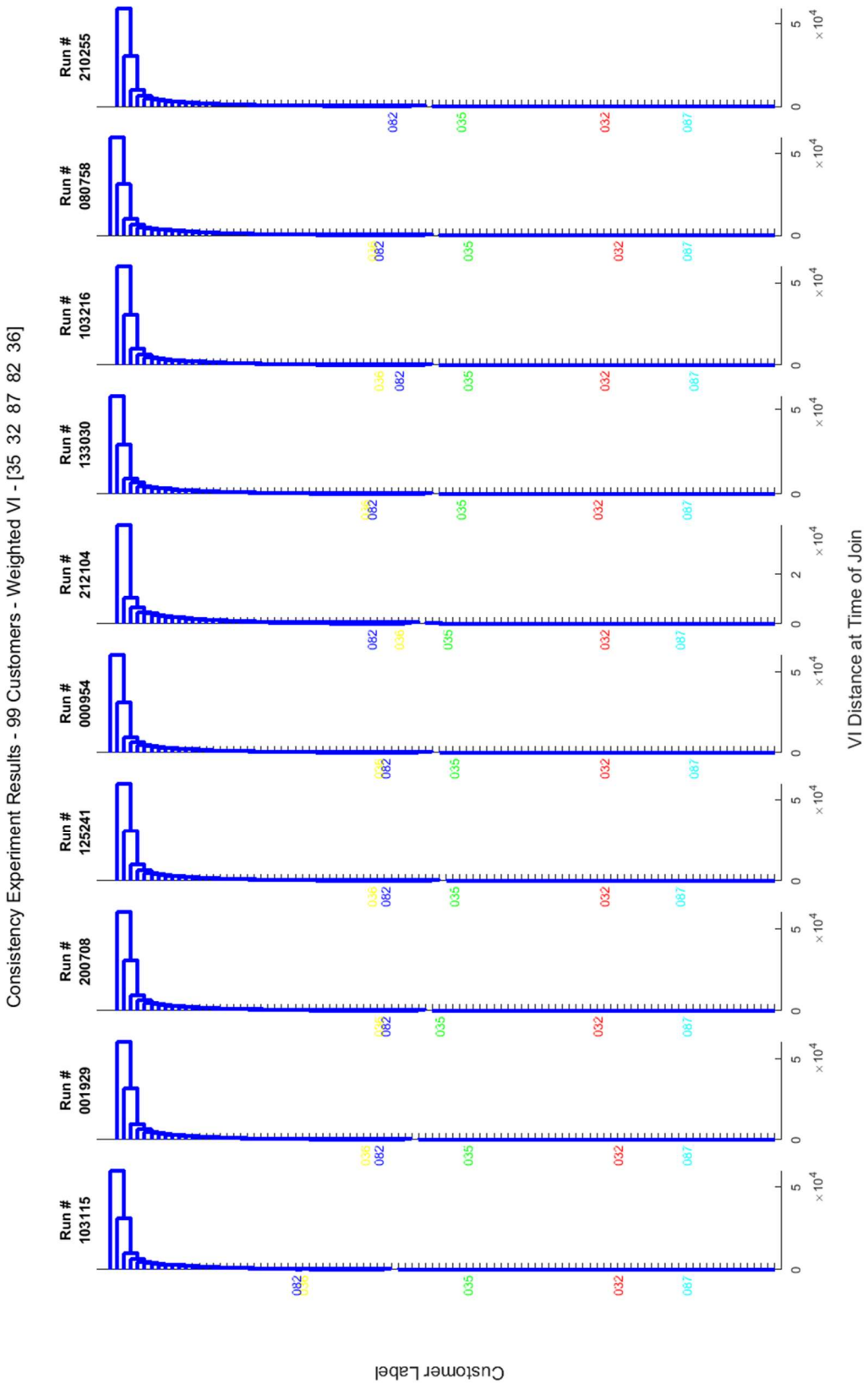

Figure 5.24 Consistency experiment results using wVI distance 16 of 20

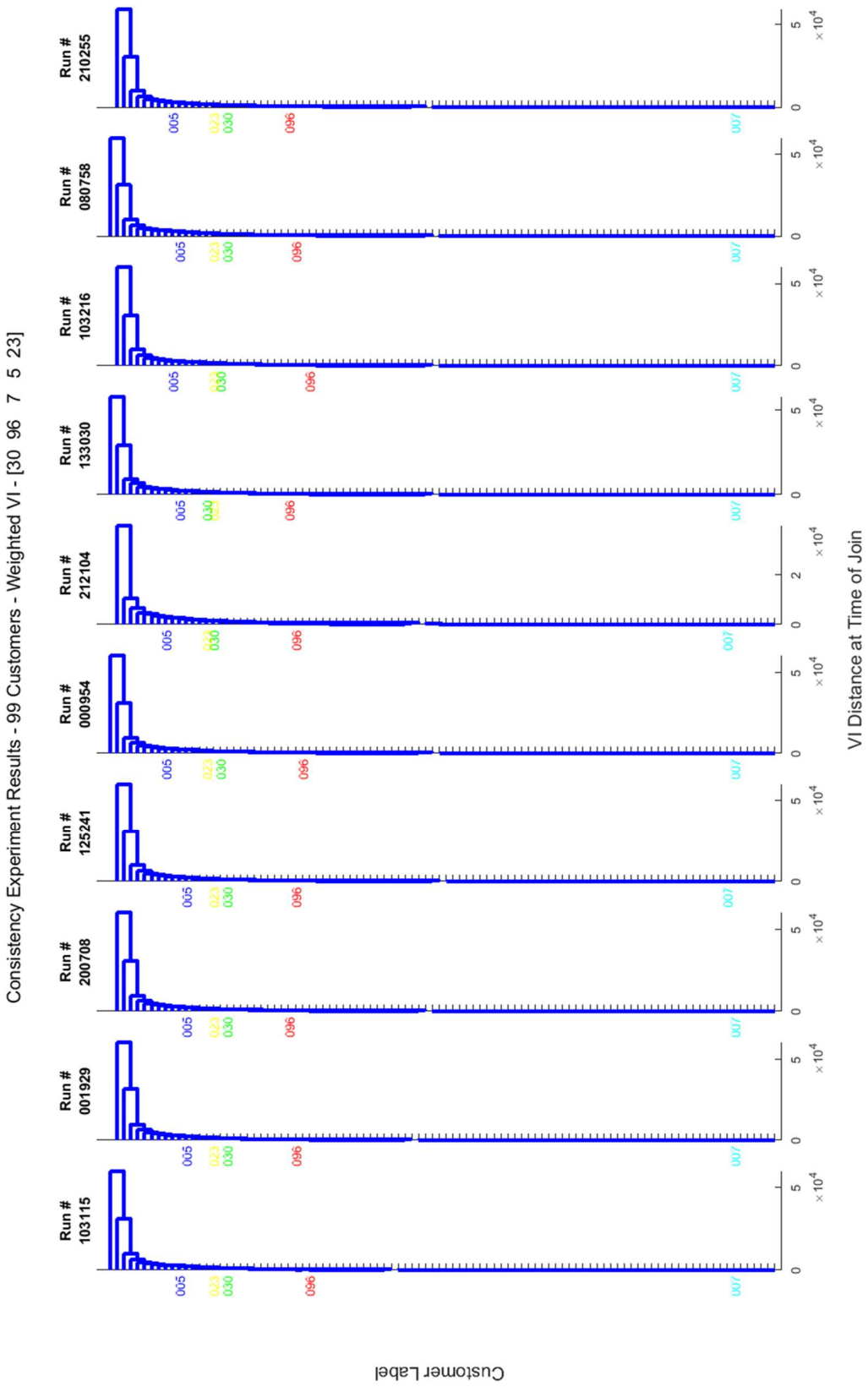

Figure 5.25 Consistency experiment results using wVI distance 17 of 20

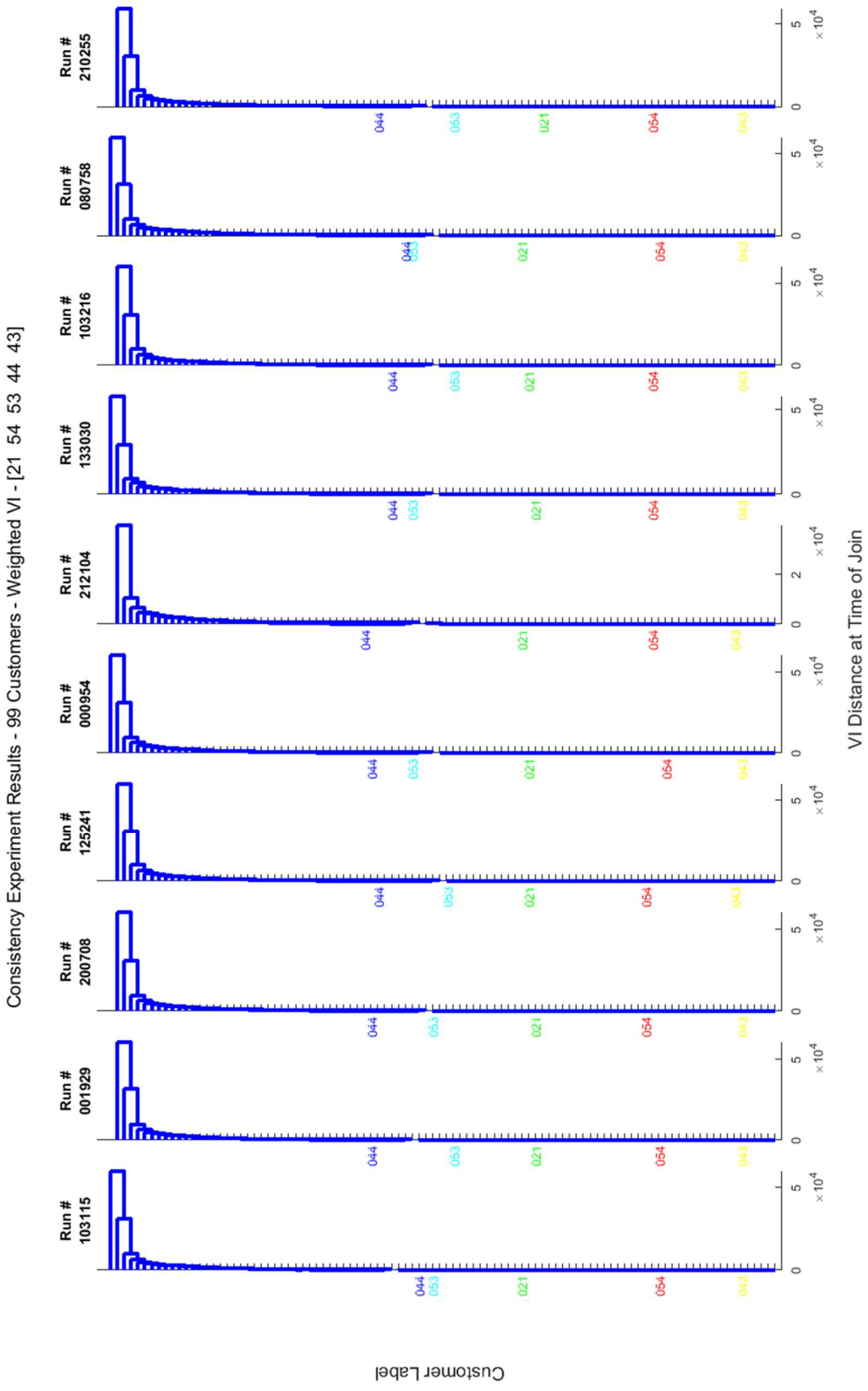

Figure 5.26 Consistency experiment results using wVI distance 18 of 20

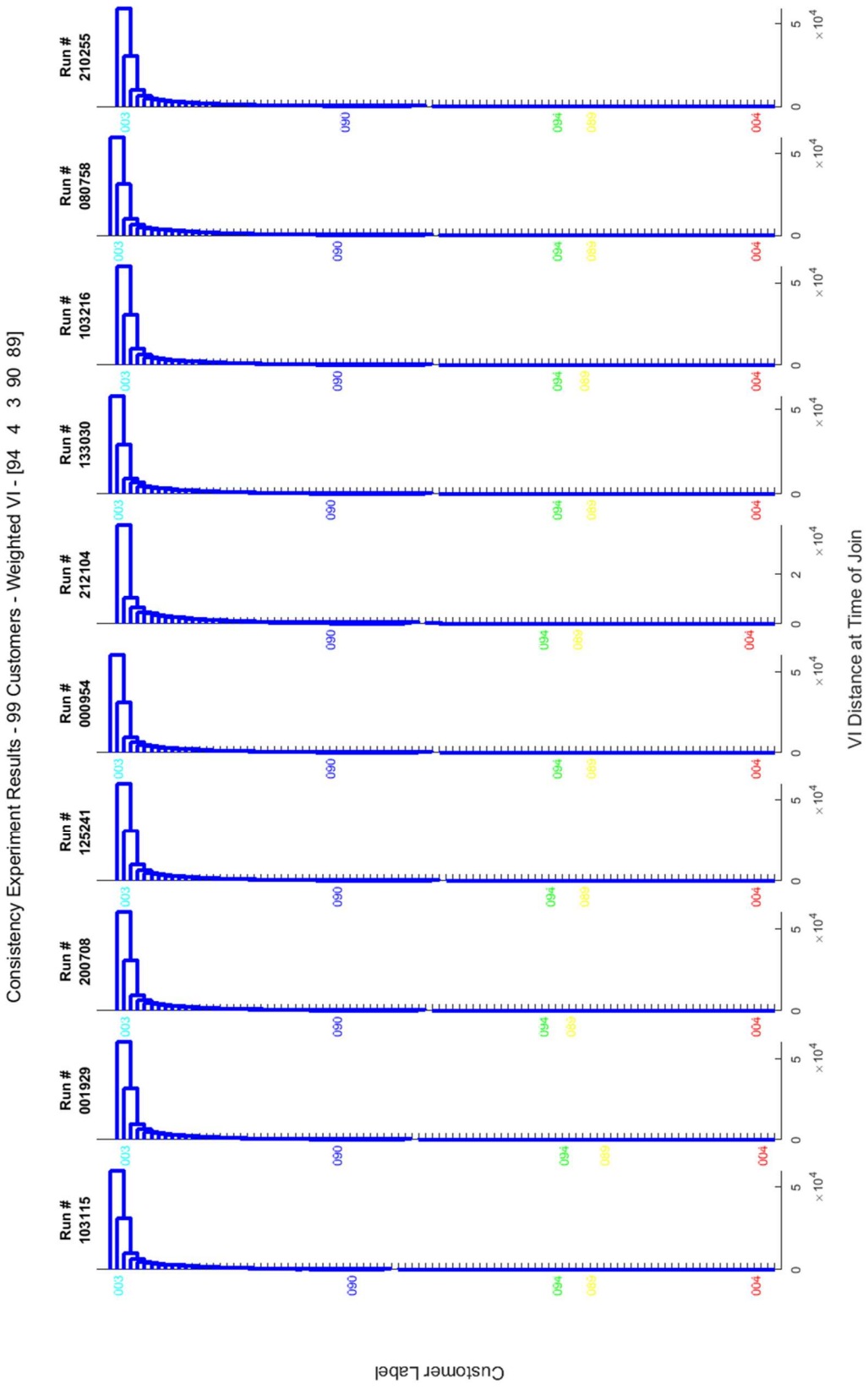

Figure 5.27 Consistency experiment results using wVI distance 19 of 20

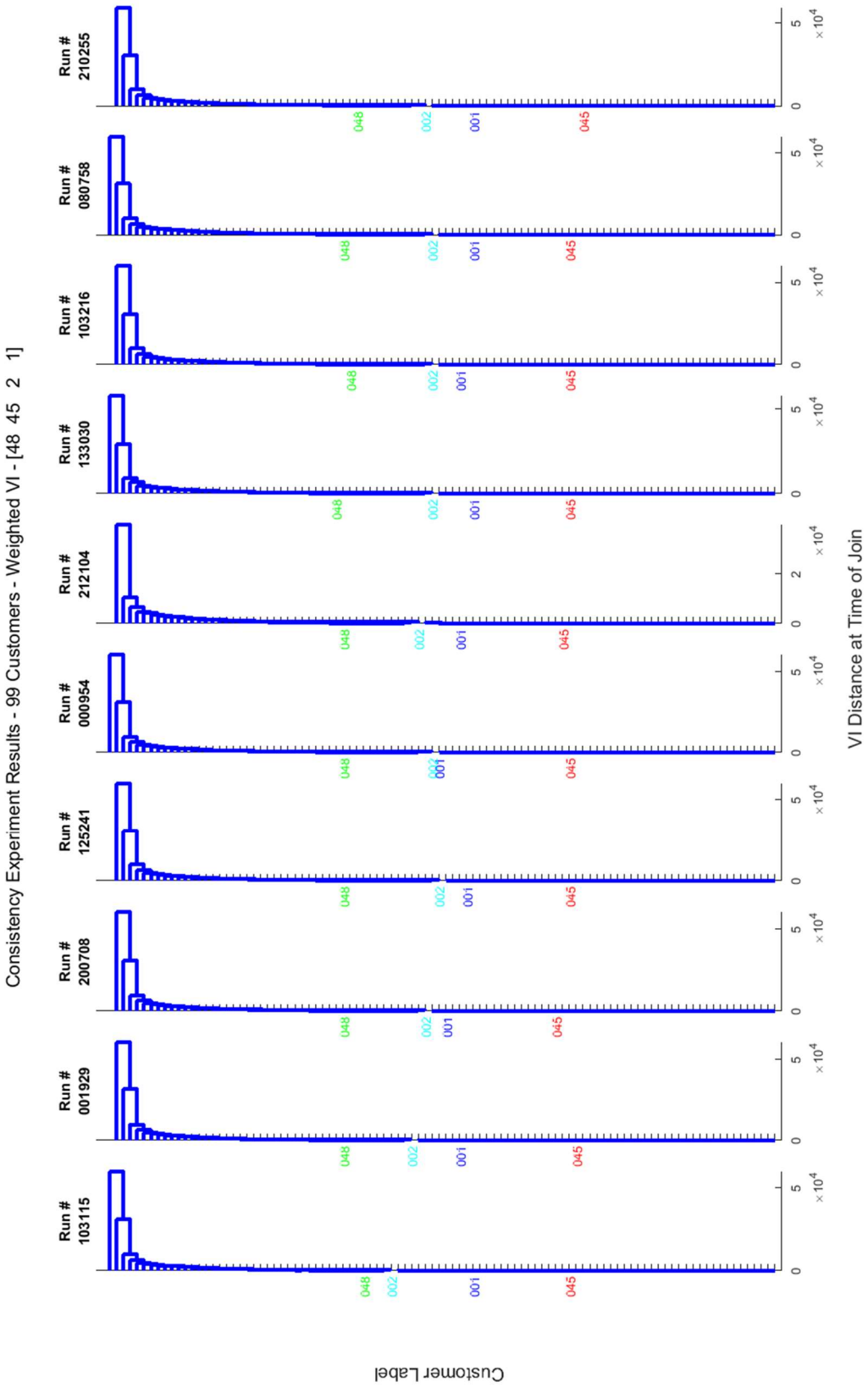

Figure 5.28 Consistency experiment results using wVI distance 20 of 20

5.7 Discussion of Consistency Experiment Results with Weighted Variation of Information

The primary motivation for developing the new weighting method is to improve the consistency of clustering results over multiple trials of the experiment. The customers who showed somewhat consistent results in Figure 4.32 are presented again in Figure 5.29, and show even less volatility between clustering trials. Using the wVI measurement produces results that are much less volatile across all trials and all customers.

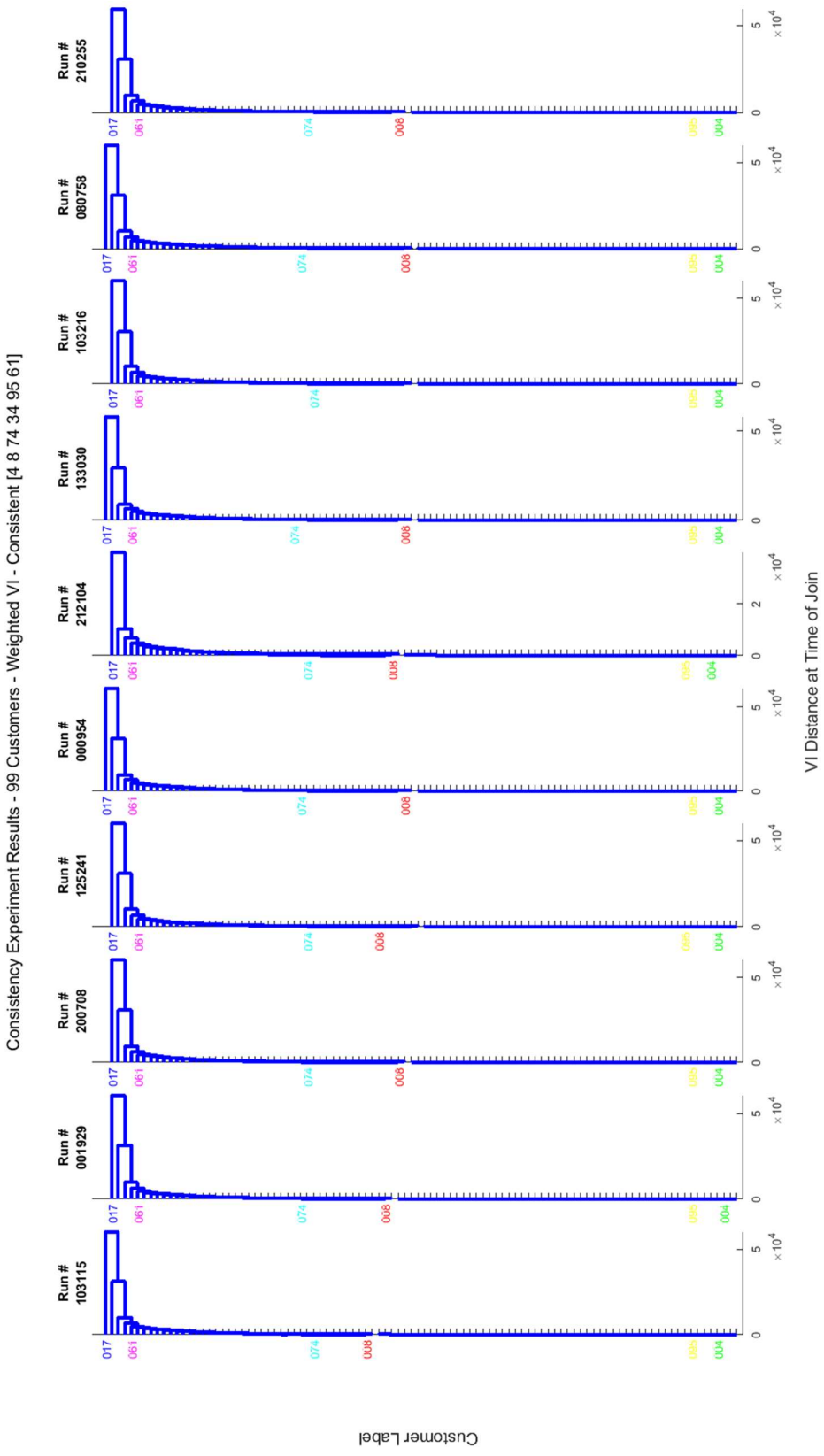

Figure 5.29 Consistency experiment results using wVI distance - consistent customers

Recall the results in Section 4.3.3 showed a group of customers with highly volatile placements between multiple trials. Those same customers are highlighted in Figure 5.30, using the clustering results from the wVI distance method. As Figure 5.30 shows clearly, using the wVI distance measure for clustering the GMMs reduces the volatility of all individual customers across many trials. This improves the consistency seen when running the clustering multiple times with the same data, and reduces the volatility caused by random differences in the Gaussian mixture models. For a practical application, the repeatability of results is critical to performance. A utility must be confident the same data produces nearly the same clusters, regardless of the randomness within the models.

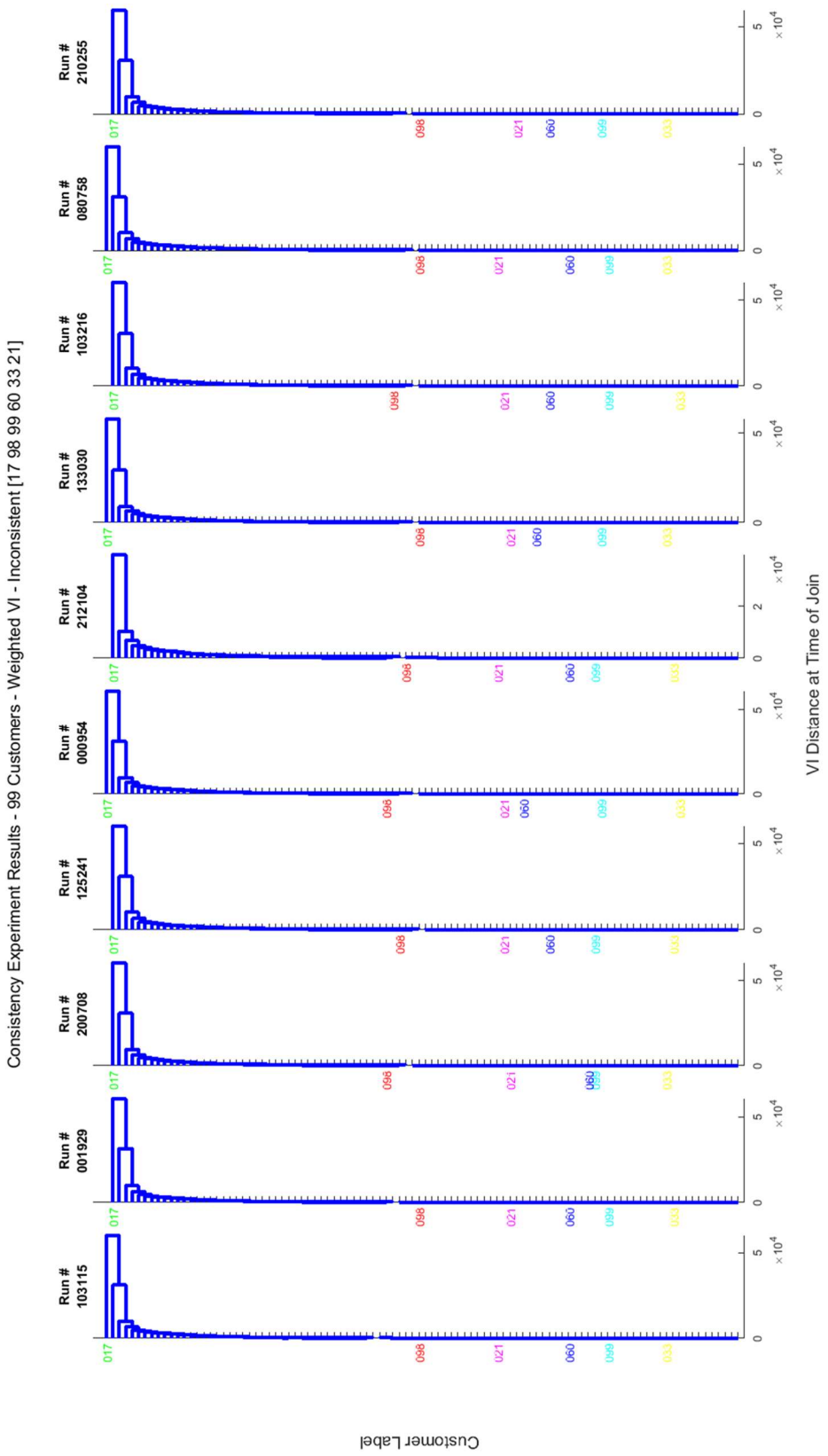

Figure 5.30 Consistency experiment results using wVI distance - inconsistent customers

While the wVI distance measure greatly reduces the volatility of all customers, some still show inconsistencies. Figure 5.31 shows the customers with the most volatility (by inspection) using the wVI distance measure. The customers in this set remain within the large grouping of customers in the lower 80% of the dendrogram, regardless of the specific order they have been clustered. This indicates the volatility is not as severe as those customers moving from “far distance” to “near distance,” seen in Figure 4.33 using the traditional VI distance measure.

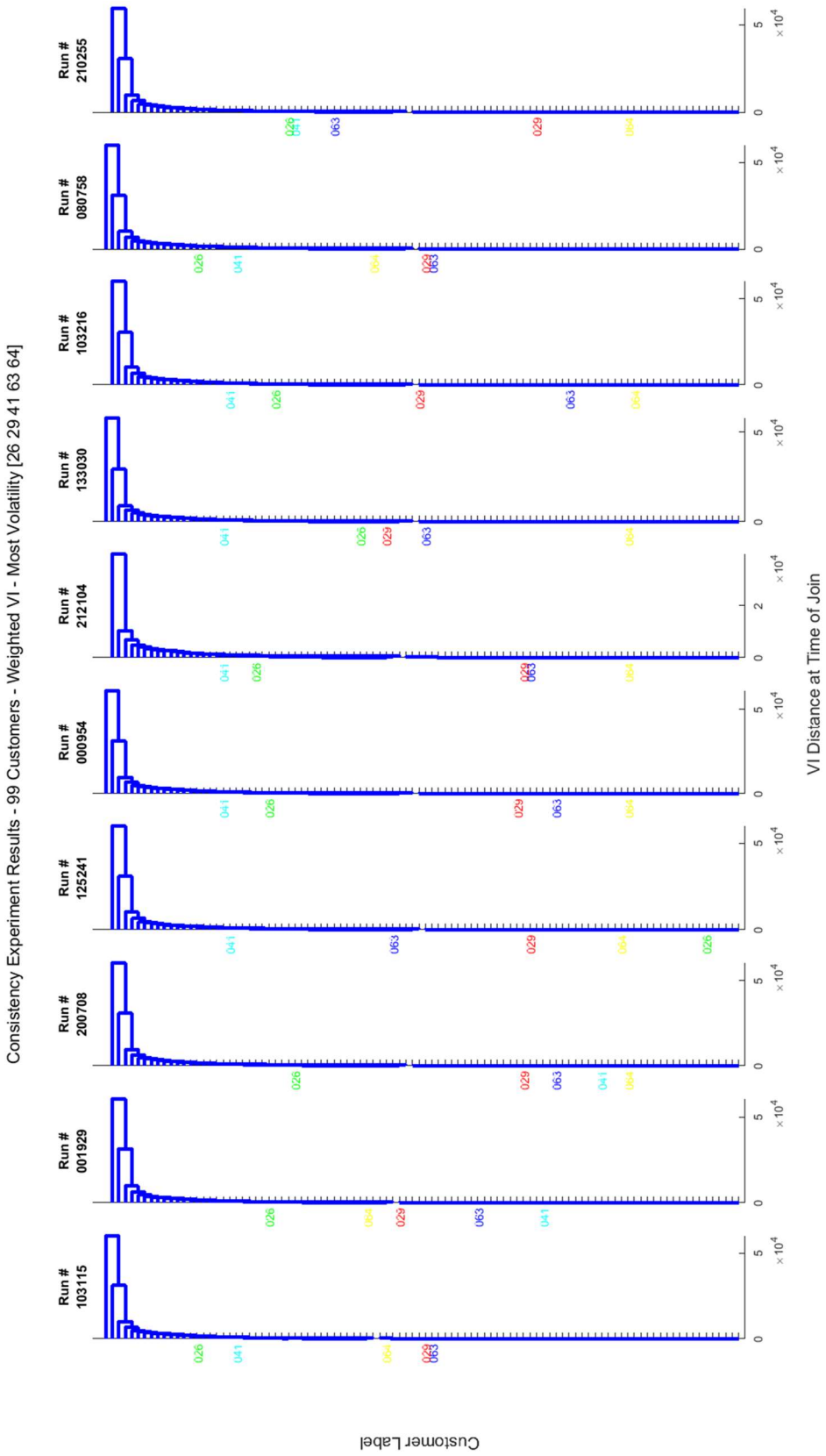

Figure 5.31 Consistency experiment results using wVI distance - highest volatility using wVI

Those customers with the furthest distances were repeatedly clustered in the same manner, with approximately the same distance relative to the rest. Figure 5.32 highlights the six customers on the top end of the dendrograms. Note the order of these does not change between all ten experimental trials, with the exception of customer 003 not appearing in Run # 212104 due to an infinite distance from the rest.

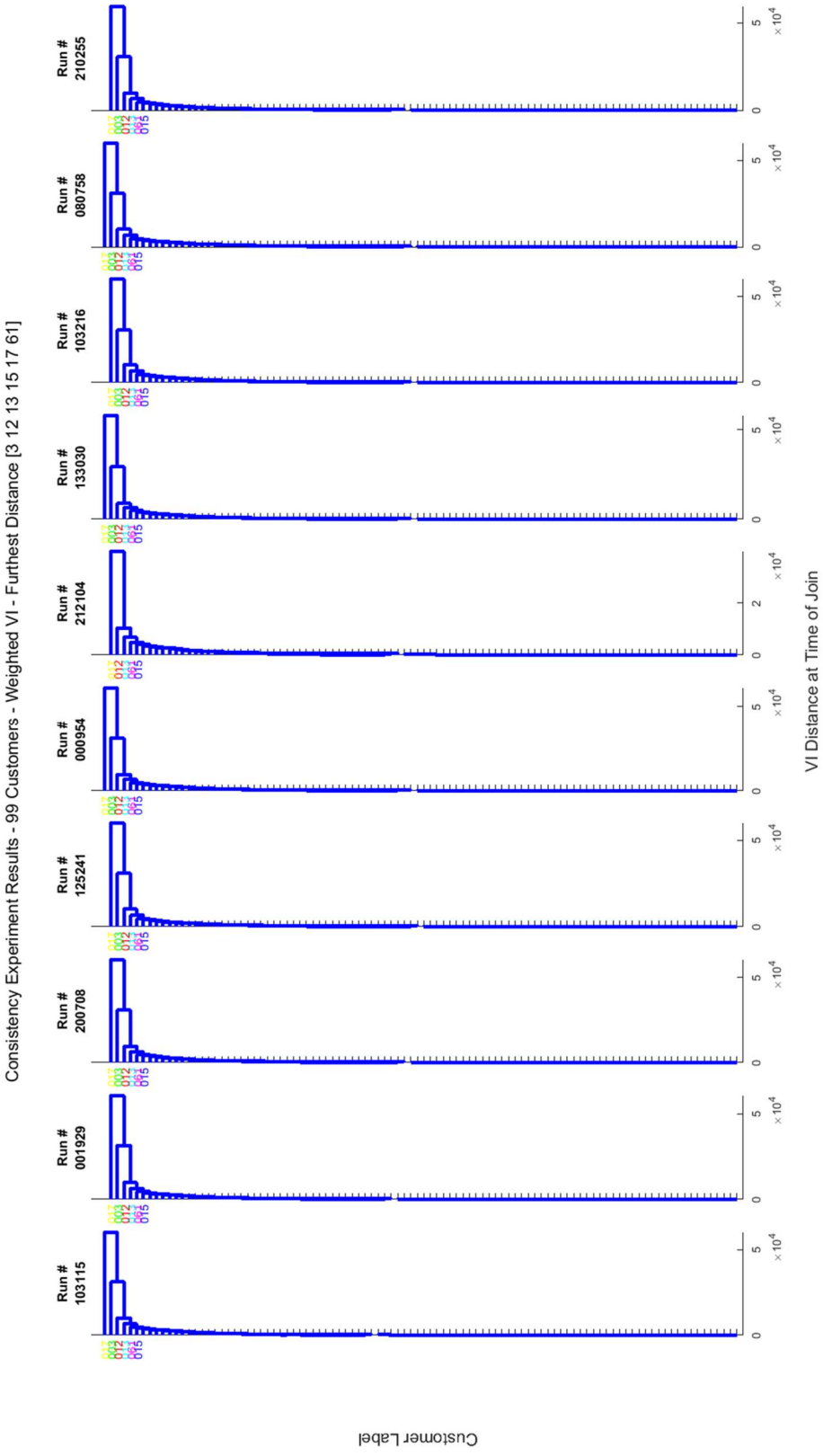

Figure 5.32 Consistency experiment results using wVI distance - largest distance from main cluster using wVI

This chapter has introduced a novel component-weighting scheme used to compute the weighted variation of information distance between two Gaussian mixture models. The new distance shows consistent clustering results when run on the same original data multiple times, using new Gaussian mixture models for each trial. Even the most volatile customers and those customers with the furthest distance from the remainder of the data remained more consistent than when using a traditional variation of information distance measure. The repeatability of these experiments makes the weighted variation of information distance more useful in a practical application when clustering utility customers, reducing the number of customers that "move" between clusters and improving the confidence in identifying customers with a great distance from the remainder of the data.

6 CONCLUSION

This dissertation addresses the need to divide a population of utility customers into groups based on their similarities and differences using only the measured flow data collected by water meters. The method of hierarchical agglomerative clustering of these utility customers based on an information theoretic distance measure is explored and tested using both the traditional variation of information and the novel weighted variation of information distance measures. Results indicate more consistent clustering occurs while using the weighted variation of information distance measure.

6.1 Summary of Methods

The work described by this dissertation is outlined in Figure 6.1. As data is collected from water meters, the measurements of flow in gallons are recorded at hourly intervals. The records are stored as time series entries in the Badger Meter, Inc. BEACON® Advanced Metering Analytics system. Chapter 1 provides context of the water meter domain and details about the specific equipment used for collecting the data. Prior to any clustering, the data requires preprocessing to eliminate anomalies and errors that will invalidate the clustering results, as described in Section 2.3. The data also are normalized per customer by the non-zero median value, leaving only the behavioral patterns and relative magnitude of flow. Finally, a triangular filter is applied to smooth out the human behavioral jitter and systematic time drift within the sensor network.

Upon completion of the preprocessing step, dimensional reduction is performed in Section 2.4. Within this procedure, the quantity of data is reduced from individual records at each time interval to a Gaussian mixture model of the data within a reconstructed phase space with time lags of 0, 24, and 168 hours. The purpose of the reconstructed phase space is to generate areas within the space related to daily and weekly habitual water consumption behaviors. The

Gaussian mixture models reduce the space required to store a representation of a single meter, and allow the direct comparison of multiple meters with different quantities of historical data.

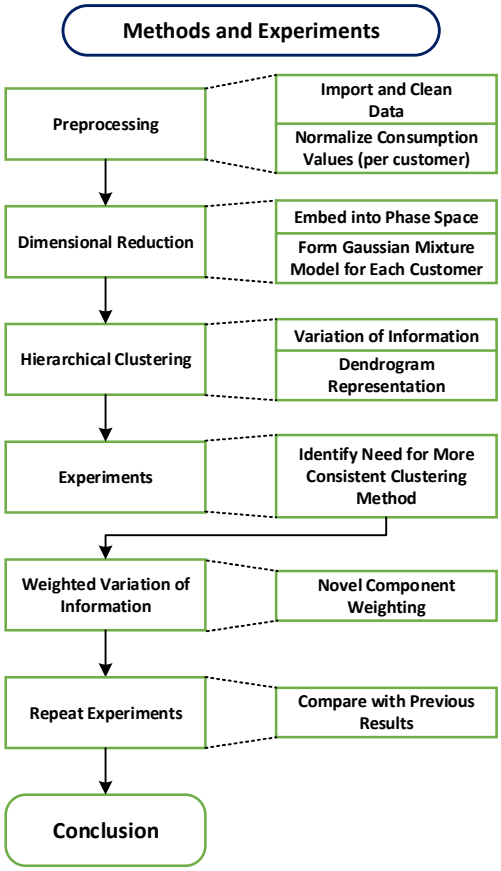

Figure 6.1 Flow diagram of the methods and experiments in this research

Following the dimensional reduction and preprocessing, the hierarchical clustering process can begin. Chapter 3 describes the clustering process in detail. Specific unsupervised clustering techniques are discussed. Distance measures are defined and compared, with supporting examples to illustrate advantages and shortcomings. Finally, the implementation of this method using MATLAB® is explained in Section 3.4, with specific functions used for each stage of the process.

Experimental procedures for testing the performance of this method are examined in Chapters 4 and 5. Chapter 4 explores the basic method, discussing whether the clustering

approach is appropriate for the water meter domain, and provides initial results from the hierarchical clustering based on variation of information distance measure. The results show some volatility in clustering outcomes between subsequent trials of the same data set, illustrating the need for a new distance measure. Chapter 5 introduces the weighted variation of information distance, and defines the weighting of individual Gaussian components within a Gaussian mixture model. The experiments discussed in Chapter 5 show the greatly improved consistency between experimental trials, supporting the efficacy of this new weighting scheme.

6.2 Future Work

This work can be expanded through enhancements to pre- and post-processing methods, exploring the reconstructed phase space further, identification of evolutionary customer behavior, and practical improvements for commercial application. Additionally, the weighted variation of information distance measure can be generalized further for more flexibility in applications.

6.2.1 Handling of Missing Reads and Gaps in the Time Series

The data cleaning implemented in this work extracts the longest consecutive set of measurements with no gaps, disaggregated records, or negative values. This brute-force approach simply excludes data that will cause the clustering method to crash. A more robust method would identify the anomalies and modify the clustering method to accommodate them. The customer model currently stores only a single time series as input to generate the model. A different approach could accommodate multiple, nonconsecutive time series to accommodate large gaps in the recorded data. This would involve making several initial Gaussian mixture models for a particular customer, one for each time series segment. The collection of sub-models would need to be combined into one master-model used for clustering amongst all customers, weighting the contribution to the master by the amount of data used to create that particular sub-model.

In another implementation, the preprocessing of large gaps in the data may include disaggregating the sum of consumption over the missing time. The system would determine the expected value during the missing periods, based on previously collected data. For instance, the weekday expected value pattern composed of two Gaussian distributions illustrated in Figure 2.13 could be used as the function to scale the known missing volume. This pattern is unique to every customer and day types (day of week, weekday, weekend, or other schedules). Once the scaled expected values are recreated, the disaggregated data can be entered into the former gap in the time series. This will not provide additional insight in the model (creating a model from itself is moot) but will allow the data system to handle a single continuous time series rather than several smaller time series. The advantage will be simplified implementation, data storage, and handling of the data by the program compared to the previous suggestion of storing many time series individually for one meter.

6.2.2 Detecting and Correcting Negative Flow Measurements

Any data showing negative values has been omitted from this work. However, water meter readings sometimes indicate negative flow. True reverse flow occurs when water passes through a meter in the reverse direction, causing a mechanical meter to spin backwards or an electronic meter to detect negative flow. Erroneous reverse flow is caused by sensor noise, mechanical jitter, or communication errors. True reverse flow measurements are considered alarm conditions by many utilities as a backflow event may introduce contaminants into the water supply [69], [110]. Due to this alarm condition, true reverse flow episodes must be maintained in the data for alerting the system managers and customers with a model indicating reverse flow need further investigation. Erroneous measurements are a nuisance, and a pre-processing step to filter these from the data models is desirable, such as replacing the negative flow with imputed values suggested by [47]. In some cases, the sensor error or mechanical jitter is obvious – a very large magnitude negative flow followed by a very large magnitude positive flow as the next

reading does not have this error. The data cleaning procedure must identify the error type and apply corrective action if necessary before continuing to model creation. Care must be taken to preserve the underlying actual usage obscured by the much larger magnitude of the jitter.

6.2.3 Describing Clusters by Typical Flow Profiles

Post-processing techniques can improve the usefulness of the clustering method. Certain clusters with high distances from the main group can be evaluated further to identify underlying undesirable behavior such as a leak or abnormally heavy usage. The post-processing can implement a second stage to generate a typical flow profile similar to [42] based on the behavior of the members of a particular cluster. Utility experts then can review this flow profile, labeling that group of customers based on their behavior. One example of this labeling process is identification of customers with irrigation patterns. A utility expert can identify an irrigation pattern by inspection due to the large irrigation flow volume occurring during specific months of the year and times of the day when irrigation is permitted by the municipality [3], [17], [49], [111]. Customers with irrigation patterns are potentially targets for conservation campaigns, as drought conditions often influence the local rules governing irrigation behavior.

6.2.4 Individual Meter Migration Between Clusters

A particular meter will not have the same usage pattern forever due to underlying changes in the behavior of the occupants [43], [46], [112]. In a residential application, the occupants may sell the home. This can cause weeks or months of abnormal usage and then a new pattern once the new occupants settle into the house. An owner may install a water feature, swimming pool, or change their landscaping, creating large water flow events where none had occurred previously. A family may change habits over the years, as children grow into teenagers and bathing activity changes. Then, as those same teenagers become adults, the habits change again as the occupancy decreases. The research presented here separated volume from behavior through normalization of the data, but other research could include non-normalized data.

During any of these common events, the model created for a particular meter will change. The grouping of customers will reflect these behaviors as a meter migrates from one cluster to a different. A separate system could monitor wVI distances compared to other meters and identify the migratory behaviors of individuals over time. This may be a slow migration such as children growing and changing the bathing habits gradually or an abrupt migration such as an occupancy change. The migratory patterns may follow a meta-model within the utility, for example, occupancy changes may have a pattern of behavior that is identifiable and repeats throughout many neighborhoods and properties.

Further migratory patterns may show the recorded flow deviating from the meter's expected model. In some cases, the meter may exhibit a leak pattern in addition to the normal consumption pattern [58], [113]. This offset for a leak may be masked by other usage and not easily identified as a leak through traditional detection methods of continuous flow, especially when the leak is smaller than the lowest resolution of the meter. A meter migrating to a leak pattern should trigger an alert for additional action to notify the owner and correct the problem.

6.2.5 Adding a New Model to the Existing Clustering

Two known shortcomings of this work are the unknown amount of data required to create a viable model and the lack of an easy method to add a new customer without re-clustering the entire data set. The individual meters in this work are treated as if there are no changes in ownership or commercial usage, a naïve assumption. A change in residential ownership may include new family habits and behaviors [49], [111]. Commercial usage changes may indicate a new owner, or a change in the business processes that occur within the location. One example is a gas station that expands to include a car wash facility. This new model must be generated from a reasonable period of data collection and then added to the existing dendrogram in a suitable position.

The amount of data required to make a suitable model should be identifiable through robust experimentation. Supplementing the existing data set with synthetic customers having progressively shorter time series durations will create many customers with expected distances near to the donor customer. Clustering on a large group of these will identify the minimum time series duration required for a representative model based on the synthetic customers that no longer have a close distance to the donor. The experiment should be repeated on a large scale, across all types of commercial and residential meters, as the threshold for required time series duration may be affected by end use and meter size or resolution.

Upon creating a new model for insertion to the data space, the current method requires clustering of the entire data set. Exploiting the metric properties of the variation of information [77], a new mathematical method can be implemented to identify the distance to each of the original customer models. Then, using the triangle inequality [114] or ultra-metric strategy [115], a suitable location for insertion can be computed. This strategy will allow the insertion of some customers into the existing dendrogram without repeating the entire clustering process, but good practice suggests periodically refreshing the entire clustering.

6.2.6 Improvements to the Weighted Variation of Information Distance

The weighted variation of information measure presented here may also be improved by future work. The method currently assumes each dimension within a component of the Gaussian mixture model is equally important, but additional research may determine specific weights based on the orientation of the Gaussian component within the phase space relative to the dimensional axes. Higher order models may have more complex weighting schemes defined by combinations of dimensions. This preference for one dimension or a weighted combination of dimensions may be related to work such as principal component analysis within a reconstructed phase space [57], [116].

6.2.7 Hierarchical Clustering Evaluation Measure for Unsupervised Applications

The experiments presented in this work illustrate the need for a new evaluation measure specifically used for comparing hierarchical clustering when labeled data is unavailable. The existing evaluation techniques are unsuitable for this application due to limitations on data or the required flexibility of number of clusters due to the commercial application.

Measures such as the Rand [86], adjusted Rand (ARI) [117], or hierarchy agreement indices (HAI) [118], [119] require labeled data or a ground truth clustering for comparison. In absence of these labels, the measurements only determine the agreements between two clusterings, not including any ability to scale the disagreement based on the distance between where a customer was assigned in one clustering versus the other. The lack of labels presents a further problem in that expert labels for water consumption behaviors have not been defined by the field. Individual fixture signatures have been explored [43]–[46], but nothing has been defined for assigning behaviors at the meter level. These population-based measures, (Rand, ARI, HAI) could possibly be used in conjunction with some form of cross-validation. This approach would require holding each clustering trial as the ground truth, and comparing all other trials to it. Repeating this for every trial to be the ground truth provides a set of pairwise indices that can be combined in some manner to provide a measure describing the overall similarity across many trials of the same experiment.

Other popular evaluation measures such as silhouettes [73] or Dunn's index [120] require a fixed number of clusters to be defined. The hierarchical clustering process was specifically chosen to accommodate varying numbers of clusters related to constraints outside the data itself – financial, human, and material resources of the commercial application. While these measures will identify compact and separated clusters, they will not provide assurance that the clustering procedure will result in similar clusters forming among multiple trials. Further, the commercial application is often more concerned with a solution that can be replicated and works better than

the current approach, rather than being focused upon the exact optimal solution at a cost of resources or time. These measures of separation and compactness might be applied at every step of the linkage in the hierarchy, but may clutter the results with more information than can easily be interpreted.

Another possible solution to the evaluation is adapting the variation of information to compare the linkages themselves. This method requires defining a suitable calculation of entropy based on the linkage itself, combining the individual joins and the distances thereof. Care must be taken when defining this entropy such that those few individuals with great distances from the others do not skew the resulting measurement.

Finally, some measure might be created to indicate the individual volatility for each customer within the set of trials. This necessarily requires accounting for the distance at time of join and some measure of the average join location. Perhaps the measure for each customer multiplies the difference between the maximum and minimum join positions with the wVI of the average join position. Some representative value computed from this set of customer measures then defines the overall performance of the clustering technique.

6.3 Discussion of Contributions

This research contributes a method for processing water meter time series data as well as a novel approach to weighing components within a model. The method of unsupervised hierarchical clustering using information-theoretic distance measures is flexible enough to accommodate different numbers of clusters as the individual application requires, and needs no training set of labeled customers to determine which individuals have similar behavior to each other. These advantages make the method appropriate for implementation in water utilities where resources of time, finances, equipment, or staff are limited. The weighted variation of information distance measure presented here improves the clustering consistency to engender confidence in

the results, with customers assigned similarly throughout multiple experiment trials. The weighted variation of information focuses on flow event behaviors with a tight variation in time and volume and relies less upon behaviors that vary widely from day to day.

REFERENCES

[1] "U.S. Drought Portal." [Online]. Available: http://drought.gov/drought/. [Accessed: 07-Aug-2014].

[2] "California Department of Water Resources Groundwater Information Center," 2017. [Online]. Available: http://www.water.ca.gov/groundwater/gwinfo/. [Accessed: 21-Apr-2017].

[3] *Water Conservation Ordinance*. City of Visalia, California, Municipal Code § 13.20.

[4] U.S. Census Bureau, "Current Housing Reports, Series H150/07," *American Housing Survey for the United States : 2007*, 2008.

[5] *WSO Water Treatment Grade 1*. American Water Works Association, 2016.

[6] M. Farhaoui and M. Derraz, "Review on Optimization of Drinking Water Treatment Process," *Journal of Water Resource and Protection*, vol. 8, pp. 777–786, 2016.

[7] *M22 Sizing Water Service Lines and Meters, Third Edition*. American Water Works Association, 2014.

[8] M. S. Crainic, "A Short History of Residential Water Meters Part I: Mechanical Water Meters with Moving Parts," in *Installation for Buildings and Ambiental Comfort Conference XXI Edition*, 2012, pp. 27–35.

[9] M. S. Crainic, "A Short History of Residential Water Meters Part II: Water Meters with no Moving Parts," in *Installation for Buildings and Ambiental Comfort Conference XXI Edition*, 2012, pp. 36–43.

[10] M. Crainic, "A Short History of Residential Water Meters Part III: Improvements of Water Meters," in *Installation for Buildings and Ambiental Comfort Conference XXI Edition*, 2012, pp. 44–52.

[11] "Chapter I — Early History of Water Measurement and the Development of Meters," *American Water Works Association*, vol. 51, no. 6, pp. 791–799, 1959.

[12] J. W. Ferguson, "Replacing Water Meters to Cut Costs Across Texas," *The New York Times*, 04-Aug-2012.

[13] C. E. Boyle, S. Eskaf, M. W. Tiger, J. A. Hughes, E. Christine, M. Wyatt, and A. Jeffrey, "Mining Water Billing Data to Inform Policy and Communication Strategies," *American Water Works Association. Journal*, vol. 103, no. 11, p. 12,45-58, Nov. 2011.

[14] "WaterSense - Understanding Your Water Bill," *United States Environmental Protection Agency*, 2017. [Online]. Available: https://www.epa.gov/watersense/understanding-your-water-bill. [Accessed: 10-Feb-2018].

[15] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-Series Clustering - A Decade Review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[16] A. Albert, “Problems, Models, and Algorithms in Data-Driven Energy Demand Management,” Stanford University, 2014.

[17] S. K. Amponsah, D. Otoo, and C. A. K. Todoko, “Time Series Analysis of Water Consumption in the Hohoe Municipality of the Volta Region, Ghana,” *International Journal of Applied Mathematical Research*, vol. 4, no. 2, pp. 393–403, 2015.

[18] I. P. Panapakidis, M. C. Alexiadis, and G. K. Papagiannis, “Three-Stage Clustering Procedure for Deriving the Typical Load Curves of the Electricity Consumers,” in *2013 IEEE Grenoble Conference PowerTech, POWERTECH 2013*, 2013.

[19] I. P. Panapakidis, M. C. Alexiadis, and G. K. Papagiannis, “Electricity Customer Characterization Based on Different Representative Load Curves,” in *9th International Conference on the European Energy Market (EEM)*, 2012, pp. 1–8.

[20] F. Rasheed and R. Alhajj, “A Framework for Periodic Outlier Pattern Detection in Time-Series Sequences.,” *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 569–582, 2014.

[21] D. Vijayasenan, F. Valente, and H. Bourlard, “Agglomerative Information Bottleneck for Speaker Diarization of Meetings Data,” *2007 IEEE Workshop on Automatic Speech Recognition Understanding ASRU*, vol. 12, no. 5, pp. 250–255, 2007.

[22] D. Reynolds, “Gaussian Mixture Models,” *Encyclopedia of Biometric Recognition*, vol. 31, no. 2, pp. 1047–64, 2008.

[23] B. Logan and A. Salomon, “A Music Similarity Function Based on Signal Analysis,” in *IEEE International Conference on Multimedia and Expo 2001*, 2001, pp. 952–955.

[24] E. Pampalk, “Speeding Up Music Similarity,” in *MIREX 2005, 2nd Annual Music Information Retrieval Evaluation eXchange*, 2005.

[25] T. W. Liao, “Clustering of Time Series Data - A Survey,” *Pattern Recognition*, vol. 38, pp. 1857–1874, 2005.

[26] R. Xu and D. Wunsch II, “Survey of Clustering Algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[27] T. Fu, “A Review on Time Series Data Mining,” *Engineering Applications of Artificial Intelligence*, vol. 24, pp. 164–181, 2011.

[28] J. Lin, E. Keogh, L. Wei, and S. Lonardi, “Experiencing SAX: A Novel Symbolic Representation of Time Series,” *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, Apr. 2007.

[29] Y. Sun, J. Li, J. Liu, B. Sun, and C. Chow, “An Improvement of Symbolic Aggregate Approximation Distance Measure for Time Series,” *Neurocomputing*, vol. 138, pp. 189–198, 2014.

[30] G. Zaib, U. Ahmed, and A. Ali, “Pattern Recognition Through Perceptually Important Points in Financial Time Series,” *WIT Transactions on Modelling and Simulation*, vol. 38, no. 12, 2004.

[31] G. Chicco, R. Napoli, and F. Piglione, "Comparisons Among Clustering Techniques for Electricity Customer Classification," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 933–940, 2006.

[32] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems 14*, pp. 849–856, 2001.

[33] C.-L. Liu, "A Tutorial of the Wavelet Transform," *NTUEE*, vol. 2, pp. 1–72, 2010.

[34] Z. Ghahramani, "An Introduction to Hidden Markov Models and Bayesian Networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1. pp. 9–42, 2001.

[35] A. C. Lindgren, M. T. Johnson, and R. J. Povinelli, "Speech Recognition Using Reconstructed Phase Space Features," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003*, 2003.

[36] C. Laspidou, E. Papageorgiou, K. Kokkinos, S. Sahu, A. Gupta, and L. Tassiulas, "Exploring Patterns in Water Consumption by Clustering," *Procedia Engineering*, vol. 119, pp. 1439–1446, 2015.

[37] J.-S. Chou and A. S. Telaga, "Real-Time Detection of Anomalous Power Consumption," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 400–411, 2014.

[38] A. Albert, R. Rajagopal, and R. Sevlian, "Power Demand Distributions: Segmenting Consumers Using Smart Meter Data," in *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings - BuildSys '11*, 2011, p. 49.

[39] C. Beckel, L. Sadamori, T. Staake, and S. Santini, "Revealing Household Characteristics from Smart Meter Data," *Energy*, vol. 78, pp. 397–410, 2014.

[40] C. Flath, D. Nicolay, T. Conte, C. van Dinther, and L. Filipova-Neumann, "Cluster Analysis of SmartMetering Data: An Implementation in Practice," *Business and Information Systems Engineering*, vol. 4, no. 1, pp. 31–39, 2012.

[41] F. Fusco, M. Wurst, and W. J. Yoon, "Mining Residential Household Information from Low-Resolution Smart Meter Data," in *21st International Conference on Pattern Recognition (ICPR 2012)*, 2012, pp. 3545–3548.

[42] Y. Il Kim, J. M. Ko, and S. H. Choi, "Methods for Generating TLPs (Typical Load Profiles) for Smart Grid-Based Energy Programs," in *2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*, 2011, pp. 1–6.

[43] R. M. Willis, R. Stewart, D. P. Giurco, M. R. Talebpour, and A. Mousavinejad, "End Use Water Consumption in Households: Impact of Socio-Demographic Factors and Efficient Devices," *Journal of Cleaner Production*, vol. 60, pp. 107–115, 2013.

[44] R. Cardell-Oliver and G. Peach, "Making Sense of Smart Metering Data: A Data Mining Approach for Discovering Water Use Patterns," *Australian Water Association Water Journal*, vol. 40, pp. 124–128, 2013.

[45] R. Cardell-Oliver, “Discovering Water Use Activities for Smart Metering,” in *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2013, pp. 171–176.

[46] R. Cardell-Oliver, “Water Use Signature Patterns for Analyzing Household Consumption Using Medium Resolution Meter Data,” *Water Resources Research*, vol. 49, no. 12, pp. 8589–8599, Dec. 2013.

[47] H. Akouemo, “Data Cleaning in the Energy Domain,” Marquette University, 2015.

[48] R. Brown and R. J. Povinelli, “Personal Communication.” 2016.

[49] W. DeOreo, P. W. Mayer, L. Martien, M. Hayden, A. Funk, M. Kramer-Duffield, R. Davis, J. Henderson, B. Raucher, P. Gleick, and M. Heberger, “California Single Family Water Use Efficiency Study.” 2011.

[50] F. Takens, *Detecting Strange Attractors in Turbulence*. Berlin, Heidelberg: Springer, 1981.

[51] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, Second. Cambridge: Cambridge University Press, 2004.

[52] M. T. Johnson and R. J. Povinelli, “RPS Toolbox,” 2003. [Online]. Available: http://povinelli.eece.mu.edu/itr-speech/download/. [Accessed: 15-Jan-2018].

[53] P. D. McNicholas, *Mixture Model-Based Classification*. Boca Raton, Florida: CRC Press, 2017.

[54] J. Mei, M. Liu, Y.-F. Wang, and H. Gao, “Learning a Mahalanobis Distance-Based Dynamic Time Warping Measure for Multivariate Time Series Classification,” *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1363–1374, 2016.

[55] Sajama and A. Orlitsky, “Supervised Dimensionality Reduction Using Mixture Models,” in *22nd International Conference on Machine Learning*, 2005, pp. 768–775.

[56] M. Qiao and J. Li, “Two-Way Gaussian Mixture Models for High Dimensional Classification,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 3, no. 4, pp. 259–271, 2010.

[57] R. J. Povinelli, M. T. Johnson, A. C. Lindgren, and J. Ye, “Time Series Classification Using Gaussian Mixture Models of Reconstructed Phase Spaces,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 779–783, 2004.

[58] S. A. McKenna, F. Fusco, and B. J. Eck, “Water Demand Pattern Classification from Smart Meter Data,” in *Procedia Engineering*, 2014, vol. 70, pp. 1121–1130.

[59] MATLAB®, “fitgmdist,” *MATLAB®*. [Online]. Available: https://www.mathworks.com/help/stats/fitgmdist.html. [Accessed: 01-Jan-2018].

[60] G. Vallabha, “plot_gaussian_ellipsoid,” *MATLAB® Central File Exchange*, 2016. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/16543-plot-

gaussian-ellipsoid. [Accessed: 01-Jan-2018].

[61] I. Bose and X. Chen, “Detecting Temporal Changes in Customer Behavior,” *Proceedings of the International Electrical Engineering Congress*, pp. 3–6, 2014.

[62] I. Bose and X. Chen, “A Fuzzy Clustering Based Analysis of Migratory Customer Behavior,” *2011 International Conference on Computational and Information Sciences*, pp. 480–483, 2011.

[63] J. H. Jensen, D. P. W. Ellis, M. G. Christensen, and S. H. Jensen, “Evaluation of Distance Measures Between Gaussian Mixture Models of MFCCs,” *International Symposium on Music Information Retrieval (ISMIR 2007)*, vol. 2, pp. 107–108, 2007.

[64] T. Jebara, Y. Song, and K. Thadani, “Spectral Clustering and Embedding with Hidden Markov Models,” in *European Conference on Machine Learning*, 2007, pp. 164–175.

[65] R. J. Povinelli, M. T. Johnson, A. C. Lindgren, and F. M. Roberts, “Statistical Models of Reconstructed Phase Spaces for Signal Classification,” *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2178–2186, 2006.

[66] A. Di Nardo, M. Di Natale, G. F. Santonastaso, V. Tzatchkov, and V. H. Alcocer Yamanaka, “Divide and Conquer Partitioning Techniques for Smart Water Networks,” *Procedia Engineering*, vol. 89, pp. 1176–1183, 2014.

[67] A. Candelieri and F. Archetti, “Identifying Typical Urban Water Demand Patterns for a Reliable Short-Term Forecasting - The Icewater Project Approach,” *Procedia Engineering*, vol. 89, pp. 1004–1012, 2014.

[68] A. Candelieri, D. Soldi, and F. Archetti, “Layered Machine Learning for Short-Term Water Demand Forecasting,” *Environmental Engineering and Management Journal*, vol. 14, no. 9, pp. 2061–2072, 2015.

[69] D. Garcia, D. González Vidal, J. Quevedo, V. Puig, and J. Saludes, “Water Demand Estimation and Outlier Detection from Smart Meter Data Using Classification and Big Data Methods,” in *2nd New Developments in IT & Water Conference*, 2015, pp. 1–8.

[70] A. Candelieri, D. Soldi, D. Conti, and F. Archetti, “Analytical Leakages Localization in Water Distribution Networks Through Spectral Clustering and Support Vector Machines. The Icewater Approach,” in *16th Conference on Water Distribution System Analysis*, 2014, vol. 89, pp. 1080–1088.

[71] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, “Efficient Agglomerative Hierarchical Clustering,” *Expert Systems with Applications*, vol. 42, no. 5, pp. 2785–2797, 2015.

[72] B. Rezaee, “A Cluster Validity Index for Fuzzy Clustering,” *Fuzzy Sets and Systems*, vol. 161, no. 23, pp. 3014–3025, 2010.

[73] P. J. Rousseeuw, “Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[74] R. C. de Amorim and C. Hennig, “Recovering the Number of Clusters in Data Sets with Noise Features Using Feature Rescaling Factors,” *Information Sciences*, vol. 324, pp. 126–145, 2015.

[75] M. Srinivas and C. Krishna Mohan, “Efficient Clustering Approach using Incremental and Hierarchical Clustering Methods,” in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–7.

[76] H. Yu, Z. Liu, and G. Wang, “An Automatic Method to Determine the Number of Clusters Using Decision-Theoretic Rough Set,” *International Journal of Approximate Reasoning*, vol. 55, pp. 101–115, 2014.

[77] M. Meila, “Comparing Clusterings—An Information Based Distance,” *Journal of Multivariate Analysis*, vol. 98, no. 5, pp. 873–895, 2007.

[78] E. H. S. Humaid, “A Data Mining Based Fraud Detection Model for Water Consumption Billing System in MOG,” Islamic University of Gaza, 2012.

[79] G. Chicco, “Overview and Performance Assessment of the Clustering Methods for Electrical Load Pattern Grouping,” *Energy*, vol. 42, no. 1, pp. 68–80, 2012.

[80] S. P. Rao and D. J. Cook, “Identifying Tasks and Predicting Action in Smart Homes using Unlabeled Data,” in *Proceedings of the Machine Learning Workshop on The Continuum from Labeled to Unlabeled Data*, 2003.

[81] M. Steinbach, L. Ertöz, and V. Kumar, “The Challenges of Clustering High Dimensional Data,” in *New Directions in Statistical Physics*, Springer Berlin Heidelberg, 2004, pp. 273–309.

[82] A. K. Jain, “Data Clustering: 50 Years Beyond k-Means,” *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[83] S. Das, A. Abraham, and A. Konar, “Automatic Clustering Using an Improved Differential Evolution Algorithm,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 1, pp. 218–237, 2008.

[84] F. Murtagh and P. Contreras, “Methods of Hierarchical Clustering,” *Computer*, vol. 38, no. 2, pp. 1–21, 2011.

[85] Y. Rubner, C. Tomasi, and L. J. Guibas, “The Earth Mover’s Distance as a Metric for Image Retrieval,” *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[86] W. M. Rand, “Objective Criteria for the Evaluation of Clustering Methods,” *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[87] E. B. Fowlkes and C. L. Mallows, “A Method for Comparing Two Hierarchical Clusterings,” *Journal of the American Statistical Association*, vol. 78, no. 383, p. 553, 1983.

[88] A. L. Gibbs and F. E. Su, “On Choosing and Bounding Probability Metrics,” *International*

Statistical Review, vol. 70, no. 3, pp. 419–435, 2002.

[89] L. Franek and X. Jiang, "Ensemble Clustering by Means of Clustering Embedding in Vector Spaces," *Pattern Recognition*, vol. 47, no. 2, pp. 833–842, 2014.

[90] MATLAB®, "convhulln," *MATLAB®*. [Online]. Available: https://www.mathworks.com/help/matlab/ref/convhulln.html. [Accessed: 01-Jan-2018].

[91] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Transactions on Mathematical Software*, vol. 22, no. 4, pp. 469–483, 1996.

[92] J. D'Errico, "inhull," *MATLAB® Central File Exchange*, 2012. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/10226-inhull. [Accessed: 01-Jan-2018].

[93] W. R. Adrion, M. A. Branstad, and J. Cherniavsky, "Validation, Verification, and Testing of Computer Software," *ACM Computing Surveys*, vol. 14, no. 2, pp. 159–192, 1982.

[94] R. Sargent, "Verification and Validation of Simulation Models," *Journal of Simulation*, vol. 7, no. 1, pp. 12–24, 2013.

[95] *IEEE Standard for System and Software Verification and Validation*. New York, NY: IEEE Computer Society, 2012.

[96] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Clustering Validation Measures," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 982–994, 2013.

[97] G. Ver Steeg, A. Galstyan, F. Sha, and S. DeDeo, "Demystifying Information-Theoretic Clustering," in *Proceedings of The 31st International Conference on Machine Learning (ICML)*, 2014, vol. 32.

[98] C. Li and G. Biswas, "Applying the Hidden Markov Model Methodology for Unsupervised Learning of Temporal Data," *International Journal of Knowledge Based Intelligent Engineering Systems*, vol. 6, pp. 152–160, 2002.

[99] G. Kunkel, *M36 Water Audits and Loss Control Programs*. Denver, CO: American Water Works Association, 2016.

[100] T. Britton, R. Stewart, and K. O'Halloran, "Smart metering: Enabler for Rapid and Effective Post Meter Leakage Identification and Water Loss Management," *Journal of Cleaner Production*, vol. 54, pp. 166–176, 2013.

[101] S.-C. Hsia, Y.-J. Chang, and S.-W. Hsu, "Remote Monitoring and Smart Sensing for Water Meter System and Leakage Detection," *IET Wireless Sensor Systems*, vol. 2, no. 4, pp. 402–408, 2012.

[102] J. Almandoz, E. Cabrera, F. Arregui, E. Cabrera Jr., and R. Cobacho, "Leakage Assessment Through Water Distribution Network Simulation," *Journal of Water Resources Planning and Management*, vol. 131, no. 6, pp. 458–466, 2005.

[103] R. Gomes, J. Sousa, and A. Sá Marques, "Influence of Future Water Demand Patterns on

the District Metered Areas Design and Benefits Yielded by Pressure Management," in *12th International Conference on Computing and Control for the Water Industry*, 2013.

[104] T. Britton, R. Stewart, and K. O'Halloran, "Smart Metering: Providing the Foundation for Post Meter Leakage Management," *Journal of Cleaner Production*, vol. 54, no. 2013, pp. 166–176, 2009.

[105] R. T. Clemen, "Linear Constraints and the Efficiency of Combined Forecasts," *Journal of Forecasting*, vol. 5, no. 3, pp. 31–38, 1986.

[106] A. H. Murphy and R. L. Winkler, "Probability Forecasting in Meterology," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 489–500, 1984.

[107] R. L. Winkler, "Combining Probability Distributions from Dependent Information Sources," *Management Science*, vol. 27, no. 4, pp. 479–488, 1981.

[108] W. M. Stallings and G. M. Gillmore, "A Note on 'Accuracy' and 'Precision,'" *Journal of Educational Measurement*, vol. 8, no. 2, pp. 127–129, 1971.

[109] R. Brown, "Personal Communication." 2017.

[110] G. Kunkel, S. Bowns, F. S. Brainard, B. Brainard, K. Brothers, T. Brown, L. Counts, T. Galitza, D. Gilles, P. Godwin, T. Holder, W. Hutcheson, T. Jakubowski, P. Johnson, D. Jordan, D. Kirkland, C. Leauber, R. Liemberger, J. Lipari, D. Liston, J. Liston, D. Mathews, T. McGee, R. McKenzie, R. Meston, R. Ruge, J. Hock, M. Simpson, J. Thornton, M. Shepherd, and A. Vickers, "Committee report: Applying Worldwide BMPs in Water Loss Control," *Journal / American Water Works Association*, vol. 95, no. 8, pp. 65–79, 2003.

[111] B. Jorgensen, M. Graymore, and K. O'Toole, "Household Water Use Behavior: An Integrated Model," *Journal of Environmental Management*, vol. 91, no. 1, pp. 227–236, 2009.

[112] R. M. Willis, R. A. Stewart, K. Panuwatwanich, P. R. Williams, and A. L. Hollingsworth, "Quantifying the Influence of Environmental and Water Conservation Attitudes on Household End Use Water Consumption," *Journal of Environmental Management*, vol. 92, no. 8, pp. 1996–2009, Aug. 2011.

[113] M. Prieto, M. Murado, J. Bartlett, W. Magette, and T. P. Curran, "Mathematical Model as a Standard Procedure to Analyze Small and Large Water Distribution Networks," *Journal of Cleaner Production*, vol. 106, pp. 541–554, 2014.

[114] R. Chitta and M. N. Murty, "Two-Level k-Means Clustering Algorithm for k–t Relationship Establishment and Linear-Time Classification," *Pattern Recognition*, vol. 43, no. 3, pp. 796–804, 2010.

[115] L. Zheng, T. Li, and C. Ding, "Hierarchical Ensemble Clustering," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2010, vol. 1, pp. 1199–1204.

[116] J. Xi and W. Han, "Application of High-Order Cumulant in the Phase-Space Reconstruction of Multivariate Chaotic Series," in *Proceedings of 2010 International*

Conference on Intelligent Control and Information Processing, ICICIP 2010, 2010, pp. 49–53.

[117] K. Yeung and W. Ruzzo, "Details of the Adjusted Rand index and Clustering algorithms Supplement to the paper 'An empirical study on Principal Component Analysis for clustering gene expression data,'" *Bioinformatics*, pp. 1–6, 2001.

[118] D. M. Johnson, C. Xiong, J. Gao, and J. J. Corso, "Comprehensive Cross-Hierarchy Cluster Agreement Evaluation," *Late-Breaking Developments in the Field of Artificial Intelligence - Papers Presented at the 27th AAAI Conference on Artificial Intelligence, Technical Report*, vol. WS-13-17, pp. 56–58, 2013.

[119] D. M. Johnson, "From Hierarchies to Metrics: Learning Nonlinear Models of Semantic Association," State University of New York, 2017.

[120] J. C. Dunn, "Well-Separated Clusters and Optimal Fuzzy Partitions," *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.