ADVANCING THE CYBERINFRASTRUCTURE FOR SMART WATER METERING
AND WATER DEMAND MODELING

by

Nour A. Attallah

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

Approved:

____________________
Jeffery S. Horsburgh, Ph.D.
Major Professor

David Rosenberg, Ph.D.
Committee Member

____________________
Brian Crookston, Ph.D., PE
Committee Member

Alfonso Torres-Rua, Ph.D.
Committee Member

____________________
John Edwards, Ph.D.
Committee Member

D. Richard Cutler, Ph.D.
Interim Vice Provost
of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2022
ABSTRACT

ADVANCING THE CYBERINFRASTRUCTURE FOR SMART WATER METERING
AND WATER DEMAND MODELING

by

Nour A. Attallah

Utah State University, 2022

Major Professor: Dr. Jeffery S. Horsburgh
Department: Civil and Environmental Engineering

The research in this dissertation sought to advance the cyberinfrastructure available for current smart metering systems through: 1) development of an open source water end use disaggregation and classification algorithm that can break down total water use data measured at the household level into different end uses; 2) developing and testing a prototype datalogger that can not only collect high resolution water use data, but also serves as a field-based computational node capable of executing classification/disaggregation algorithms on the trace of high-resolution data collected at the data collection location using edge computing techniques; and 3) developing and testing an end use water demand model that can simulate and predict residential water use behavior at a city level using high resolution smart metering data. Given that residential water users consume as much as 56% of water delivered by a municipality or other water provider, efficient design and operation of water distribution system infrastructures requires knowledge of when and how water is being used by households (e.g., showers, toilets, faucets, dishwashers, outdoor irrigation, etc.). Although most water providers use analog water meters for monitoring water use on a periodic basis and for issuing and
checking the accuracy of water bills, these meters are typically read monthly to quarterly and do not provide detailed data that could be used for improving system design and operation. The recent advent of smart water metering technology allows for monitoring and recording water use at a high temporal scale (e.g., recording water use as frequently as every 1-5 seconds). While smart water meters can provide the type of high-resolution water use data needed to characterize the behavior of residential water users and the timing of their water use, the volume of data produced can be a significant roadblock without effective computer and data analytical infrastructure, collectively called “cyberinfrastructure,” needed to collect, manage, and extract usable information from the data. Results of this work advance existing understanding of residential water use and show how key information provided to water managers could help them decide whether rebate programs for replacing underperforming fixtures might yield conservation benefits. Results may also help in measuring the actual impact of such programs.

Outdoors, where water use can drastically exceed indoor uses, knowing the exact time, volume, and frequency of landscape irrigation can help homeowners and water managers identify inefficient use and make corrections where needed. In times of crisis, when restrictions are set in place, prompt access to this type of data could help water managers evaluate the effectiveness of their restrictions.

(210 pages)
PUBLIC ABSTRACT

ADVANCING THE CYBERINFRASTRUCTURE FOR SMART WATER METERING
AND WATER DEMAND MODELING

Nour A. Attallah

With rapid growth of urban populations and limited water resources, achieving an appropriate balance between water supply capacity and residential water demand poses a significant challenge to water supplying agencies. With the recent emergence of smart metering technology, where water use can be monitored and recorded at high resolution (e.g., observations of water use every 5 seconds), most existing research has been aimed at providing water managers with detailed information about the water use behavior of their consumers and the performance of water using fixtures. However, replacing existing meters with smart meters is expensive, and effectively using data produced by smart meters can be a roadblock for water utilities that lack sophisticated information technology expertise. The research in this dissertation presents low cost, open source cyberinfrastructure aimed at addressing these challenges. Components developed include an open source algorithm for identifying and classifying water end use events from smart meter data, a low cost datalogging and computational device that enables existing water meters to collect high resolution data and compute end use information, and a detailed water demand model that uses end use event information to simulate residential water use at a municipality level. Using this cyberinfrastructure, we conducted a case study application in the cities of Logan and Providence, Utah. We tested the applicability of the disaggregation algorithm in quantifying water end uses for different meter sizes and
types. We tested the datalogging computational device at a residential household and demonstrated collection, disaggregation, and transfer of high resolution flow data and classified events into a secure server. Finally, we demonstrated a water demand model that simulates the detailed water end uses of Logan’s residents using a combination of a set of representative water end use events and monthly billing data. Using the data we collected and the outputs from the model, we demonstrated opportunities for conserving water through improving the efficiency of water using fixtures and promoting behavior changes.
ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my supervisor Dr. Jeff Horsburgh for his invaluable guidance and patience throughout my PhD’s journey. His immense knowledge and life experience supported me in my academic research and daily life. I would also like to thank my committee members Dr. David Rosenberg, Dr. Brian Crookston, Dr. Alfonso Torres-Rua, and Dr. John Edwards for their technical contribution on my research. I thank my colleagues at the Utah Water Research Laboratory Camilo Bastidas, Amber Jones, Joseph Brewer, and AJ Beckwith. I have really enjoyed the opportunity to get to work closely with all of you in the past years. I honestly could not ask for better colleagues to share the office with. Thanks all for making it such a pleasure to be at the lab each day. Finally, I would like to express my gratitude and appreciation to my parents, my wife and my siblings. It would be impossible for me to complete my PhD study without their tremendous encouragement from the beginning of my journey.

I would like to express my gratitude to the Utah Water Research Laboratory and the U.S. National Science Foundation (Grant CBET 1552444 - Cyberinfrastructure for Intelligent Water Supply (CIWS): Shrinking Big Data for Sustainable Urban Water) for the financial support provided during my studies. I also acknowledge Logan and Providence Cities for their cooperation in the field data collection campaigns and for providing additional data.

Nour A. Attallah
CONTENTS

Abstract .......................................................................................................................... iii
Public Abstract .............................................................................................................. v
Acknowledgments .......................................................................................................... vii
Contents ......................................................................................................................... viii
List of Tables .................................................................................................................. xi
List of Figures ................................................................................................................ xiii

Chapter

1. Introduction .................................................................................................................. 1
   References ..................................................................................................................... 10

2. Splitting Overlapping Events And Classifying Water End Uses Using A Non-Intrusive, Self-Learning End Use Disaggregation Algorithm .............................................. 13
   Abstract ....................................................................................................................... 13
   2.1 Introduction ............................................................................................................ 14
   2.2 Background ............................................................................................................ 17
      2.2.1 Data Cleansing Techniques ........................................................................... 17
      2.2.2 End Use Disaggregation Techniques .............................................................. 18
      2.2.3. End Use Classification Techniques .............................................................. 20
   2.3. Methods .............................................................................................................. 22
      2.3.1. Case Study Design and Data Collection ....................................................... 22
      2.3.2. Data Cleansing ............................................................................................ 25
      2.3.3. Event Disaggregation ................................................................................. 26
      2.3.4. Event Classification ..................................................................................... 30
      2.3.5. Feature Scaling ............................................................................................. 30
      2.3.6. Feature Selection .......................................................................................... 31
      2.3.7. Semi-Supervised Classification ................................................................... 32
      2.3.8. Software Design and Implementation .......................................................... 36
   2.4. Results and Discussion ....................................................................................... 37
      2.4.1. End Use Classification Accuracy ................................................................. 38
      2.4.2. Overlapping Events ..................................................................................... 41
      2.4.3. Overall Water Use and Individual End Uses ................................................. 41
   2.5. Conclusions ......................................................................................................... 42
   Data Availability Statement ....................................................................................... 45
   Reproducable Results ............................................................................................... 46
   Acknowledgments ....................................................................................................... 46
3. Residential Water Meters As Edge Computing Nodes: Disaggregation End Uses And Creating Actionable Information At The Edge
   Abstract ........................................................................................................ 57
   3.1 Introduction ............................................................................................ 58
   3.2 System Description ................................................................................ 65
      3.2.1 Principle of Functioning .................................................................. 67
      3.2.2 Hardware .......................................................................................... 69
         3.2.2.1 Data Logging Components .................................................. 70
         3.2.2.2 Raspberry Pi ........................................................................... 70
         3.2.2.3. Bus Buffer .......................................................................... 71
         3.2.2.4. Power Control Circuit ........................................................ 72
         3.2.2.5. Real-Time Clock .................................................................. 73
         3.2.2.6. Manual Activation Button ................................................... 74
         3.2.2.7. Printed Circuit Board Design .............................................. 75
      3.2.3. Microcontroller Firmware ............................................................. 75
         3.2.3.1. System State Library ......................................................... 77
         3.2.3.2. Real-Time Clock Library ................................................. 77
         3.2.3.3. Store New Record Library .............................................. 77
         3.2.3.4. Communication Library .................................................. 80
      3.2.4. Software .......................................................................................... 80
   3.3. Calibration ............................................................................................. 85
   3.4. Case Study Application ......................................................................... 86
      3.4.1. Data Output .................................................................................... 88
      3.4.2. Battery Life ................................................................................... 89
      3.4.3. Accuracy ....................................................................................... 91
      3.4.4. Water Use ..................................................................................... 93
      3.4.5. Limitations and Errors .................................................................. 94
   3.5. Discussion and Conclusions ................................................................. 96
   Author Contributions .................................................................................. 100
   Funding ....................................................................................................... 100
   Institutional Review Board Statement ................................................... 100
   Informed Consent Statement ................................................................... 100
   Data Availability Statement ..................................................................... 101
   Acknowledgments ..................................................................................... 101
   Conflict of Interest ................................................................................... 101
   References ................................................................................................. 102
   Tables .......................................................................................................... 106
   Figures ........................................................................................................ 109

4. Simulating Community Water Use Behavior And Potential Water Conservation Using Detailed End Use Event Data ......................................................... 114
   Abstract ..................................................................................................... 114
4.1. Introduction
4.2. Methods and Materials
  4.2.1. Water End Uses (i)
  4.2.2. End Use-Level Water Demand Model Formulation
  4.2.3. Model Execution Procedure
  4.2.4. Simulating Water Conservation Strategies
  4.2.5. Case Study Application
  4.2.6. Model Validation and Comparison
  4.2.7. Water Conservation Actions
4.3. Results and Discussion
  4.3.1. Model Comparison and Applicability
  4.3.2. End Use Efficiency and Water Conservation Potential
4.4. Conclusions

Software Design and Implementation
Data Availability
Acknowledgments
References

Tables
Figures

5. Summary, Conclusions and Recommendations
References

Appendices
Appendix A. Overall Water Use and Characterizing Individual End Uses
Appendix B. Detailed Analysis of Indoor Water Use
Appendix C. Distribution of Duration, Volume, and Flowrate in all Simulated Households by End Use

Curriculum Vitae
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Event disaggregation process for different software tools.</td>
<td>51</td>
</tr>
<tr>
<td>2.2</td>
<td>Characteristics of the five households selected for water end use analysis.</td>
<td>51</td>
</tr>
<tr>
<td>2.3</td>
<td>Water end use event features.</td>
<td>52</td>
</tr>
<tr>
<td>2.4</td>
<td>Classification accuracy and F1-score measures for each end use type.</td>
<td>52</td>
</tr>
<tr>
<td>3.1</td>
<td>EEPROM data format.</td>
<td>106</td>
</tr>
<tr>
<td>3.2</td>
<td>Parts required and cost to build a CIWS computational node using off-the-shelf components.</td>
<td>106</td>
</tr>
<tr>
<td>3.3</td>
<td>Functions executed by the CIWS computational node Computational_Firmware.ino file and the main objective.</td>
<td>107</td>
</tr>
<tr>
<td>3.4</td>
<td>System State library functions.</td>
<td>107</td>
</tr>
<tr>
<td>3.5</td>
<td>RTC library functions.</td>
<td>108</td>
</tr>
<tr>
<td>3.6</td>
<td>Communication library functions.</td>
<td>108</td>
</tr>
<tr>
<td>3.7</td>
<td>Results from laboratory experiment 1 for the 1-inch and 5/8-inch Master Meters.</td>
<td>109</td>
</tr>
<tr>
<td>3.8</td>
<td>Results from laboratory experiment 2 for the 1-inch and 5/8-inch Master Meters.</td>
<td>109</td>
</tr>
<tr>
<td>4.1</td>
<td>Approaches used for different end use modeling studies.</td>
<td>156</td>
</tr>
<tr>
<td>4.2</td>
<td>Technical performance by end use type.</td>
<td>156</td>
</tr>
<tr>
<td>4.3</td>
<td>Data collected in the 2016 Residential End Uses of Water study.</td>
<td>157</td>
</tr>
<tr>
<td>4.4</td>
<td>Proposed water conservation actions and their associated characteristics in terms of water use.</td>
<td>157</td>
</tr>
<tr>
<td>4.5</td>
<td>Estimated D statistic and a p-value for each model.</td>
<td>158</td>
</tr>
<tr>
<td>4.6</td>
<td>Relative contribution of indoor water use type.</td>
<td>158</td>
</tr>
<tr>
<td>A.1</td>
<td>Indoor and outdoor water use per household.</td>
<td>180</td>
</tr>
</tbody>
</table>
Table A.2. The relative contribution of indoor water use in this study versus the 2016 Residential End Uses of Water study (DeOreo et al. 2016).........................180

Table A.3. Total number of water use events by household and end use..............180

Table B.1. Summary statistics for toilet use. .................................................................186

Table B.2. Summary statistics for showers.................................................................186

Table B.3. Summary statistics for clothes washer loads. .........................................186

Table B.4. Summary statistics for faucet events. .......................................................187

Table B.5. Summary statistics for bath filling events...............................................187
LIST OF FIGURES

Page

Figure 2.1. Water end use disaggregation methodology. ..............................................53

Figure 2.2. Example of data filtering output for a single water use event. Oscillations in the original data are caused by the combination of the data recording interval and pulse resolution of the water meter. ..........................................................53

Figure 2.3. Examples of single and overlapping events. Panel A shows multiple single use events. Panel B shows an irregular single use event. Panel C shows overlapping event. ..............................................................54

Figure 2.4. Illustration of the splitting procedure for overlapping events. .................54

Figure 2.5. Distributions of volume, duration, and flowrate features. Due to the high discrepancies in the frequency of events across the feature values, log scale y-axes were used to improve readability. .................................................................55

Figure 2.6. Boruta algorithm output. Box plots represent the distribution of importance scores over the 20 Boruta runs.................................................................55

Figure 2.7. Accuracy of different supervised classifiers. Error bars represent the standard deviation of the 10 accuracy score values. ......................................................56

Figure 2.8. Confusion matrix results for each end use type. ..............................................56

Figure 3.1. The CIWS computational node hardware architecture, including a Raspberry Pi computer, an Arduino datalogging shield, and a custom-designed Pi hat. 109

Figure 3.2. Power control circuit of the CIWS computational node. .................................110

Figure 3.3. Printed circuit board implementation of the CIWS computational node. This board is designed to mount directly to the pin header of the Raspberry Pi computer (shown underneath the PCB) and includes all of the hardware components on a single board.................................................................110

Figure 3.4. CIWS firmware architecture. ........................................................................111

Figure 3.5. CIWS computational node record storage architecture. ...............................112

Figure 3.6. CIWS computational node user interface help menu showing available functions..............................................................................................................112

Figure 3.7. Sample output from the CIWS computational node. Panel A shows the raw data. Panel B shows the extracted and classified events from the raw data........113
Figure 3.8. Solar-powered CIWS computational node prototype. ........................................113

Figure 4.1. Sampling process from the CDF of shower frequency. .................................159

Figure 4.2. Example CDF for bathtub filling events. .......................................................159

Figure 4.3. Panel a: flowrate CDFs for showers based on their technical performance, and Panel b: duration CDFs for showers based on their behavioral performance. ...............159

Figure 4.4. Panel a: sampling from flowrate CDF for shower events based on their technical performance, and Panel b: sampling from duration CDF for shower events based on their behavioral performance. ..........................................................160

Figure 4.5. Distribution of single-family households in Logan. .......................................160

Figure 4.6. Weight assigning to each 2016 REUWS dataset household. The average daily water use volume for each 2016 REUWS household is intersected with the PDF curve for the Logan dataset to obtain a probability density value for each 2016 REUWS household. These probability density values are used as the weights for 2016 REUWS households in the sampling procedure. ........................................161

Figure 4.7. PDF for average daily indoor water use of Logan's households versus the sample of households from the 2016 REUWS. .................................................................161

Figure 4.8. Cumulative distributions for observed average daily water use volume and simulated average daily water use volume of different simulation methods for all residential connections in Logan. .................................................................162

Figure 4.9. Distributions of average daily indoor water use for Logan City households between 2014-2018. .................................................................162

Figure 4.10. Ranges of potential water savings for technological and behavioral conservation actions. Some outliers were removed to enhance the readability of the figure. .................................................................163

Figure A.1. Average fraction of indoor water use volume by end use type for the five homes in the case study dataset .................................................................181

Figure A.2. Seasonal variation in the frequency of water use events by end use type ....181

Figure B.1. Distributions of toilet flushing duration, volume, and flowrate. .................187

Figure B.2. Distributions of shower event duration, volume, and flowrate. Some outliers with a duration of more than 30 minutes have been removed to improve the readability of the plot. .................................................................188
Figure B.3. Distributions of clothes washer event duration, volume, and flowrate. Some outliers with a duration of more than 12 minutes have been removed to improve the readability of the plot. .......................................................... 188

Figure B.4. Distributions of faucet event duration, volume, and flowrate. Some events with a duration of more than 2.5 minutes have been removed to improve the readability of the plot. .................................................................................. 189

Figure B.5. Distributions of bathtub filling event duration, volume, and flowrate........ 189

Figure C.1. Distributions of toilet flushing flowrate, duration, and volume across simulated households.......................................................... 190

Figure C.2. Distributions of shower event flowrate, duration, and volume across simulated households. Some outliers with a duration of more than 15 minutes have been removed to improve the readability of the plot............................................... 191

Figure C.3. Distributions of clothes washer event flowrate, duration, and volume across simulated households. Some outliers with a duration of more than 10 minutes have been removed to improve the readability of the plot.......................................................... 191

Figure C.4. Distributions of dishwasher event flowrate, duration, and volume across simulated households.......................................................... 191

Figure C.5. Distributions of bathtub event flowrate, duration, and volume across simulated households. Some outliers with a volume of more than 300 liters have been removed to improve the readability of the plot............................................... 192

Figure C.6. Distributions of faucet event flowrate, duration, and volume across simulated households. Some outliers with a duration of more than five minutes have been removed to improve the readability of the plot............................................... 192
CHAPTER 1
INTRODUCTION

As severe drought conditions have gripped the western U.S. in recent years, residential water users are being urged to reduce their water consumption, particularly for landscapes, even as lawns and shrubs wither in the dry heat. Understanding how much water a household is using, and for what purpose, can help water managers and residential consumers alike to identify ways to conserve precious water. In most large urban water systems in the U.S., the residential sector consumes the majority of total supplied fresh water with an average of about 56% of fresh water being utilized for residential uses (Dieter et al. 2018).

In the U.S., metering of residential water use is widespread (Boyle et al. 2013). However, the vast majority of residential meters rely on measurement technology created decades ago, many of the meters themselves are decades old, and most meters are only read by a water utility monthly to quarterly for billing purposes. Monthly data are too infrequent to characterize patterns in water use, leaving critical gaps in our understanding of water use behavior at the household and system level. This limits our ability to identify alternative water management strategies and opportunities for water conservation and increased efficiency. Some important questions that cannot be fully answered with monthly meter data include: 1) does the timing and volume of water use vary across socio-demographic groups and neighborhood types?; 2) how can detailed information about the timing of water demand be used by water providers to ensure water availability and efficiency now and in the future, plan for related energy demand, and increase
customer satisfaction?; and 3) how do water consumers change their behavior given detailed information about their water consumption?

With the advent of smart metering technology, several high frequency water monitoring studies were conducted worldwide from single household level to large cities (Anda et al. 2013; Boyle et al. 2013; Deoreo et al. 2016; Froehlich et al. 2011; Kowalski and Marshallsay 2005; Mayer et al. 2004, 1999a; Wong et al. 2010; Willis et al. 2013). These studies have demonstrated that high resolution data can enhance the ability to quantitatively describe the behavior of water users and to identify and characterize water use volumes from different end uses. High resolution data are typically collected on a water meter that measures the total water flow to a residence. Water end use characterization consists of breaking-down the total water use data registered at the household level into different end use categories such as toilets, showers, washing machines, faucets, etc. (Cominola et al. 2015). This is useful because identifying with a high level of certainty where, when, and how much water is used by a household can help water managers in understanding demand patterns, identifying opportunities for conservation, and in checking the effectiveness of proposed and implemented conservation actions. It can also help in identifying water users with significant capacity to conserve and in creating information that can be used to tell users how to consume less water. High resolution data may also help rebate programs target critical areas and determine success and commitment level.

In recent years, many water end use models and software tools that use high resolution metering data to identify end uses have been developed, including Trace Wizard (DeOreo et al. 1996), Identiflow (Kowalski and Marshallsay 2003), HydroSense (Froehlich et al. 2009), and Autoflow (Beal et al. 2011). These tools are resource and
time intensive, where a significant part of the data processing and analyzing require human intervention. They require centralized post-processing of collected data, and some of them also involve an intrusive period of data collection to train the algorithm with some required water end use signatures that can be used in disaggregating and characterizing end uses from the total water trace sequence. Perhaps a more significant limitation of existing work, though, is that for most existing studies that have involved working on end use disaggregation algorithms, neither the source code nor the data are available for testing, making existing work difficult or impossible to reproduce.

Previous applications of smart metering data and associated research studies have necessarily focused on the small number of cases where cities have upgraded to newer electronic meters or where individual dataloggers can be deployed to existing meters to collect high temporal resolution data (Omaghami et al. 2020; DeOreo et al. 2016; Makki et al. 2013; Beal et al. 2013; Srinivasan et al. 2011; Willis et al. 2010; Mead and Aravinthan 2009; Olmstead and Stavins 2009; Mayer et al. 2004). Broader uptake of smart water metering infrastructure has been slow, in part due to a limited focus on the back-end data mining and analytics functionality as well as front-end user orientation (Gurung et al. 2015). Several authors have recognized this lack of supporting cyberinfrastructure as a fundamental need for advancing the use of smart metering techniques (e.g., Sønderlund et al. 2016; Gurung et al. 2015; Harou et al. 2014; Mutchek and Williams 2014; Boyle et al. 2013; Fróes Lima and Portillo Navas 2012). Given the lack of options, in most existing studies high-resolution water use data were collected in the field and then transferred to a centralized location for post-processing to examine residential water use behavior, which has three significant limitations: (1) available
bandwidth of conventional telemetry systems may be inadequate for transferring the large volume of data produced to a centralized location for post-processing, requiring technicians to visit sites to manually download data, (2) the water providing utility may not have sophisticated information technology infrastructure available to them to enable data post-processing, and (3) the utility may also lack dedicated staff and technical expertise needed to employ end use disaggregation algorithms or other sophisticated analyses. A potential alternative to centralized information systems is to use a distributed approach, where data processing is performed at or near where the data are collected to extract and transmit only actionable data products to a centralized location. This distributed approach is aimed at reducing the data management and computational burden associated with tasks such as water end-use disaggregation. By mining and summarizing the big data produced by smart meters at the site of data collection, required transmission bandwidth can be minimized, and derived data products can be more readily created and used to inform and improve water system management.

The high temporal resolution datasets produced from different water use monitoring studies have revealed detailed, disaggregated water use behavior for a relatively large number of residential households (e.g., 400 households in the 2016 Residential End Uses of Water Study, DeOreo et al. 2016). Traditionally, urban residential water use and conservation potential modeling have used regressive relations based on historical trends of average household residential water use (LAWDP 2010). The approach is simple, but it does not reflect the heterogeneity of water use behavior, technical performance of end uses, and demographic factors across different households - thus limiting the utility of these model types. To begin to address this, researchers have
integrated stochastic optimization in their models. Work such as that conducted by Rosenberg (2009) in Amman, Jordan and that of Arbués et al. (2003) and Jenkins and Lund (2000) in California integrated decision trees in their models, where uncertainties (action cost, life span, etc.) were investigated using sensitivity analysis and Monte-Carlo simulation. With the increased availability of water end use studies, more mechanistic and detailed approaches to estimating and modeling residential water demand that account for behavior and technical performance of water using fixtures at the household level can now be adopted (Cahill et al. 2013). In this study, we investigated ways to scale water use estimates based on detailed end use information to the level of an entire municipality, demonstrating how valuable information could be created for water managers interested in how water demand at the city level may change with various population growth or water management scenarios.

The key to successfully initiating new advancements in the smart water metering discipline is that smart metering systems must remotely sense water flow at a resolution that improves current operational and customer decision making (e.g., at least hourly, and more frequent to quantify individual end uses) but must also include cyberinfrastructure for storing, managing, and mining collected data to produce useful information for a range of purposes relevant to both managers and consumers. Without these capabilities, the big data sets produced by smart meters will remain as roadblocks for water utility operators and will continue to limit uptake of smart metering.

The overall objective of this research was to advance smart water metering and supporting cyberinfrastructure for building the scientific data and knowledge base for sustainably managing urban water supplies. We worked to advance the
cyberinfrastructure available for building and managing next generation smart metering systems and their resultant data. The following objectives guided this research. Each of the objectives is addressed within one or more chapters of this dissertation.

Objective 1: Develop and test an open source water demand disaggregation algorithm.

Water end use disaggregation aims to separate household water consumption data collected from the main water meter into appliance/fixture-level consumption data. In recent years, the field has rapidly expanded due to increased interest in identifying opportunities to conserve water. Several water demand disaggregation algorithms have been developed to test the hypothesis that end uses of water can be effectively identified and quantified by collecting high resolution data from the single meter on the water supply line to an individual home. However, empirically comparing existing disaggregation algorithms is currently virtually impossible. This is due to the different data sets used, the lack of reference implementations of these algorithms, and the variety of accuracy metrics employed (Parson et al. 2014). Additionally, for nearly all of the published papers in this area, the source code or data are proprietary, making it impossible to reproduce the work described. The work under this objective was aimed at developing and testing a new, open source algorithm for water end use disaggregation and classification along with an open dataset for testing.

Objective 2: Investigate residential water meters as edge computing nodes: Disaggregating end uses and creating actionable information at the edge.

Up to now, the term “smart meter” has been used liberally to describe devices that are capable of measuring and recording water use data at high spatial and temporal
frequencies. However, the use of such meters has mostly been limited to data collection and leak detection with much of the “intelligence” of smart meters (e.g., the promise of automated procedures for extracting actionable information from high resolution data) going unrealized. Additionally, most residential meters in use today are analog, without high resolution data collection capabilities or additional data processing capabilities required for a meter to be “smart.” We investigated how existing, analog meters can be transformed into intelligent, computational nodes capable of “shrinking” the big data sets they produce into decision-relevant information that can be more easily transmitted - i.e., much smaller data volume requiring no post processing that can be immediately acted upon by both water managers and consumers. Under this objective, we advanced current smart metering applications by developing a device capable of turning existing analog meters into battery powered computational nodes that not only collect and store high-frequency flow data, but also calculate summary information (e.g., daily totals, timing of maxima, etc.) and execute any computational codes such as disaggregating metered flow into individual water end uses (e.g., summary totals for toilets, showers, dishwasher, etc.).

Objective 3. Develop and Test an Indoor End Use Water Demand Model Based on High Resolution Smart Metering Data.

Given the limitations of conventional water demand models that often assume averaged water use across different households and across individual water use events within a household we focused on a new approach that reflects the heterogeneity of water use behavior, technical performance of end use fixtures/appliances, and demographic factors across different simulated households. We integrated event level data derived
from smart water metering to allow for a more mechanistic and detailed approach to estimating household water demand and conservation potential. The work under this objective was focused on developing and testing a water end use demand model capable of scaling classified water end use events collected from a high temporal resolution water use monitoring study to the level of a whole municipality.

The outline of the rest of this dissertation is as follows. In Chapter 2, a new water end use disaggregation and classification tool that builds on existing end use disaggregation studies and addresses the unavailability of code and data used by prior studies is presented. The tool’s implementation and a case study application are presented. Chapter 2 mainly addresses Objective 1 but also contributed towards Objectives 2 and 3 by enabling the extraction of individual classified water end use events from residential households and the generation of the data and models used in the case studies presented in the other chapters.

Chapter 3 addresses Objective 2 by presenting the design, calibration, and field testing of a computational datalogger capable of collecting, analyzing, and transferring water end use data records collected from single family residential households to a centralized system where they can be used. Using available off-the-shelf electronic components, we designed and prototyped low cost and low-power devices capable of data collection, computation, and communication tasks required to “shrink” the Big Data generated by smart water meters into actionable information that can be used by water managers and consumers.

Chapter 4 addresses Objective 3 by presenting a case study in Logan City, Utah, that demonstrates how detailed water end use information from a sample of households
can be scaled to the level of all single-household residential connections within a whole municipality using sampling techniques. Chapter 4 builds on the developments reported in Chapters 2 and 3 to demonstrate one of the possible applications of the research products developed aimed at assisting investigations of residential water demand and water conservation potential.

The contributions of the work presented in this dissertation include a design and implementation of hardware and software tools that enable recording, analyzing, and transferring high temporal resolution data as well as case studies that demonstrate the suitability of these tools for addressing gaps in existing water end use disaggregation algorithms, centralized data management, and water end use models aimed at better understanding residential water use behavior. This research demonstrates how water end use measurement studies and a targeted residential modeling effort can provide detailed data (e.g., water end use events, volumes, and flows) for areas where water as a resource is scarce and where residential water systems may be most vulnerable. By characterizing how residential water is utilized inside households, these results provide information that may be useful for city engineers and planners in better understanding how and when water is used, in the development of best management practices, and in the design of improvements to residential water distribution infrastructure.
REFERENCES


CHAPTER 2

SPLITTING OVERLAPPING EVENTS AND CLASSIFYING WATER END USES USING A NON-INTRUSIVE, SELF-LEARNING END USE DISAGGREGATION ALGORITHM 1

Abstract

This paper demonstrates a new, open-source, non-intrusive water end use disaggregation and classification tool that can provide a detailed information of how water is used in residential settings. This type of information is significant for both water managers and homeowners in evaluating their water use and in identifying opportunities for conservation. The tool first applies a low-pass filter on unprocessed, high temporal resolution water use data collected on smart water meters to accentuate the shape of events along with their characteristics. Next, the tool applies a robust splitting technique we designed to disaggregate overlapping events formed by two or more simultaneous end uses. A random forest classifier is then used to develop an initial classification model using a manually labeled training dataset from a single home. The base classifier is then extended to homes for which no labeled events are available through a self-learning procedure that trains the model using end use events from that home. The tool was applied to five homes located in the Cities of Logan and Providence Utah, USA to demonstrate the generalizability of the tool. Results from homes with different meter types and sizes are presented to demonstrate the ability of the tool to disaggregate and

1 Nour A. Attallah, Jeffery S. Horsburgh, Camilo J. Bastidas Pacheco, An open-source semi-supervised water end use disaggregation and classification tool, Journal of Water Resources Planning and Management, 2022 (under review.)
classify high temporal resolution data into individual end use events. The tool was developed in Python and can be accessed via any current Python programming environment. The results of this paper are reproducible using openly available code and data, representing an accessible platform for advancing end use disaggregation tools.

2.1. Introduction

Water is a crucial resource on which humans depend for survival. Yet, increasing water scarcity, declining water quality, global climate change, and growing water stresses from the residential sector are building up new challenges to water managers to secure water supplies for current and future water demands. In most large urban water systems in the U.S., the residential sector consumes the majority of total supplied fresh water, using on average about 56% of piped fresh water (Contestabile 2018).

In the U.S., metering of residential water use is widespread (Boyle et al. 2013). However, most residential meters rely on measurement technology created decades ago, many of the meters themselves are decades old, and most are only read by a water utility monthly to quarterly for billing purposes. It is well known that monthly data are too infrequent to characterize patterns in water use, leaving gaps in our understanding of behavior at the household and system level (Cardell-Oliver 2013). This limits our ability to identify alternative management strategies and opportunities for water conservation and increased efficiency.

With the advent of smart metering technology, several water monitoring studies using high temporal resolution data (data with temporal resolution < 1 minute) were conducted from single household level to large cities (e.g., Anda et al. 2014; Boyle et al. 2013; Cominola et al. 2021; Sønderlund et al. 2016; DeOreo et al. 2016; Froehlich et al.
2011; Kowalski and Marshallsay 2005; Mayer et al. 2004, 1999; Wong et al. 2010; Willis et al. 2013). These studies demonstrated that high resolution data can enhance the ability to quantify water user behavior, to identify and characterize different end uses, and to formulate feedback to engage water consumers and foster water conservation. Given the difficulty associated with measuring flow at each point of use (e.g., individual faucets, toilets, etc.), high resolution data are typically collected on a water meter that measures the total water flow to a residence. These data are often referred to as a total water use “trace.” Water end use characterization consists of breaking-down the trace data at the household level into different end use categories (Cominola et al. 2015). Identifying with a high level of certainty where, when, and how much water is used by a household can help water managers in understanding demand patterns, identifying opportunities for conservation, and in checking the effectiveness of proposed and implemented conservation actions.

Several water end use models and software tools that use high resolution metering data to identify end uses have been developed, including Trace Wizard (DeOreo et al. 1996), Identiflow (Kowalski and Marshallsay 2003), HydroSense (Froehlich et al. 2009), and Autoflow (Beal et al. 2011). Limitations of these tools include: 1) they are resource and time intensive, where a significant part of the data processing and analysis require human intervention; 2) they involve an intrusive period of data collection to train the algorithm with required water end use signature data; 3) their accuracy in disaggregating overlapping events comprised of more than two simultaneous events is limited; and 4) their ability to handle oscillations caused by the data recording interval and pulse resolution of the meter is limited. To address these limitations, we developed a semi-
supervised, self-training, machine learning approach that does not require a large, fully labeled dataset of water end uses for training or prior knowledge of the number of end uses inside a home for classifying events. We also implemented noise filtering on the raw pulse data, leading to more clearly recognizable events, and a new splitting algorithm for simplifying complex, overlapping events into their single use components.

Despite the relatively large number of papers published about end use disaggregation tools and studies of water use behavior that have used them, opportunities for reproducing or replicating existing studies or building upon their results are limited because neither the data nor the code for existing algorithms are openly available and/or easily accessible (Di Mauro et al. 2020). In our inquiries with authors of these papers we found that the code was considered proprietary and could not be released, and the datasets were inaccessible due to privacy concerns. Yet, many of these papers call for additional research to verify and extend methods as well as for new applications of results, including Trace Wizard (DeOreo et al. 1996), Identiflow (Kowalski and Marshallsay 2003), HydroSense (Froehlich et al. 2009), and Autoflow (Beal et al. 2011). Thus, there is a clear need for open and reproducible approaches that enable other researchers to test, replicate, reuse, and build upon existing work. In response to these issues, this paper presents new, open source, and non-intrusive techniques for collecting and disaggregating high resolution water use data into component end uses, along with an openly available, anonymized dataset for testing this and potentially other water end use disaggregation algorithms. New results presented here not only address gaps in existing studies/algorithms, but are also openly available.
2.2. Background

Despite the variety of tools and techniques adopted by existing water end use studies, they all followed the same general four phases of: 1) data gathering; 2) data cleansing; 3) water end use disaggregation; and 4) classification (Fig. 2.1; Cominola et al. 2015; Pastor-Jabaloyes et al. 2018). Data cleansing is a preprocessing step that prepares raw, high frequency water use data for subsequent steps. Disaggregation extracts events from cleansed data, separates overlapping events (i.e., events made up of simultaneous end uses), and then identifies event features (e.g., volume, duration, etc.). Classification assigns events to an end use category based on their features. In the following sections, we describe each sub-process in more detail.

2.2.1. Data Cleansing Techniques

Water use data consist of time series of flow where the characteristics of the signal (i.e., periods of non-zero flow identified as water use events and the features of those events, including volume, duration, flowrate, etc.) reflect the type of end uses inside a household. Within the time series of flow data, periods of non-zero flow constitute water use events. Events must first be identified, and their features calculated, before they can be classified. This is not always simple given that some noise or signal distortion is always expected to be embedded in the raw trace data caused by the combination of data recording frequency and the volumetric pulse resolution at which data are collected (in many cases manifesting as a volume of water per electronic “pulse” generated by a water meter that can be counted). Noise can impede accurate data interpretation, including difficulty in identifying the start and end of events as well as accurate calculation of other event attributes/features.
Filtering can be used to remove undesired noise and make data ready for further analysis. Filtering can remove certain components of the signal (e.g., high frequency oscillations caused by the meter’s pulse resolution) while retaining other components (e.g., the overall shape of an event). Techniques for time series data filtering include empirical mode decompositions (Flandrin et al. 2004) and Monte Carlo techniques (Doucet et al. 2000). However, despite their wide use, their performance on water use data where the means, variance, and covariance change over time is poor (Nayak et al. 1999). As an alternative, Chen (2014) suggested isolating signal frequencies of interest by removing or keeping them either at the top (low-pass filter), the bottom (high-pass filter), or at both sides of the domain (band filter).

Given the variety in temporal and pulse resolutions recorded by smart water meters, filtering is an active area of investigation for cleansing high resolution water use data prior to end use analysis. For example, in this study we describe how a low-pass filtering technique can be applied to the raw trace data to maintain low frequency signals and adjust or remove high frequency oscillations caused by the pulse resolution of the meter. This worked well for the data recording frequency and water meter pulse resolutions we encountered in our case study, but we acknowledge other filtering/preprocessing techniques may work better for different data resolutions.

2.2.2. End Use Disaggregation Techniques

Single events are those where a single fixture is in use, while overlapping events occur when two or more fixtures are in use simultaneously. Trace disaggregation iterates on overlapping events until all subevents are single events. The process involves: 1) extracting events from the trace; 2) classifying events into either single or overlapping
events; and 3) breaking-down overlapping events until all resulting events are single events. An additional step involves calculating the features for all single events (e.g., start time, end time, duration, volume, average flow rate, etc.). Features play an important role in classifying events as either single or overlapping events. For example, Pastor-Jabaloyes et al. (2018) classified events as single or overlapping based on the number of vertices present in their filtered flow data, where a vertex is a point where the flow rate changes from one non-zero value to another within the same event. They presumed that events having only four vertices should be classified as single events. Events with more than four vertices were classified as overlapping.

Overlapping events have not been consistently handled by existing studies (Table 2.1). Most have concluded that overlapping events in single family houses account for a relatively small proportion of total events. Moreover, methods used by those who have tried to break down overlapping events were built upon assumptions (e.g., if the flow rate changes within an event, a new event is assumed to be introduced to the trace, and the time at which the flow rate change occurred is assumed to be the start time of the new event). Table 2.1 reports a summary of features incorporated in the disaggregation process, methods used in classifying events as single or overlapping, and methods used in breaking-down overlapping events into single events for the most used end use disaggregation software tools.

While the methods in Table 2.1 work for identifying single and in some cases disaggregating two overlapping end uses, to our knowledge no method has been able to accurately disaggregate overlapping events comprised of more than two overlapping end uses. Thus, in our case study, we describe a new method for disaggregating complex,
overlapping events formed by more than two simultaneous end uses by incorporating multiple physical features in the disaggregation process.

2.2.3. End Use Classification Techniques

One accurate approach to better understand household water use traces is through simultaneously monitoring each individual water end use inside the household with a water meter. However, developing such instrumentation is time-consuming, invasive, and expensive (Längkvist et al. 2014). An alternative is to use machine learning to classify events of different end use types derived from the raw trace data logged on the main water meter. Supervised techniques require a training dataset in which a set of events have been recorded and classified and from which a model is derived for classifying new events based on their features. Unsupervised techniques can cluster similar events regardless of the number of known events and without labeled training data, but it can be difficult to interpret the clusters produced. If labels for a small set of events are available, a class of algorithms called semi-supervised learning can be used. Semi-supervised techniques combine a small set of labeled data with a large set of unlabeled data to aid in the classification process. Here we define a “small dataset” as one that could be created by a single homeowner through manually labeling events (e.g., on the order of hundreds of real events). Supervised machine learning classifiers often require training datasets with on the order of tens of thousands of instances to train and test the algorithm, whereas with semi-supervised classifiers, a dataset with less than 1,000 instances can be used to train and test the algorithm (Chapelle et al. 2010).

In the context of end use disaggregation and classification, most studies have adopted supervised techniques such as decision trees (Kowalski and Marshallsay 2003;
DeOreo et al. 1996), Hidden Markov Model (HMM) (Nguyen et al. 2013), Bayesian probabilistic models (Froehlich et al. 2011), and Hybrid Signature-based Iterative Disaggregation (Cominola et al. 2017). Others have utilized unsupervised, partitional clustering such as K-means (Yang et al. 2018) and K-medoids (Pastor-Jabaloyes et al. 2018). However, the practical application of these techniques is limited because they either require predefining the number of clusters (i.e., the number of water end uses inside a household) in the case of the unsupervised techniques, or a large group of water end use events that have been accurately tagged with one or more labels that can be used to train a supervised model. Determining the number of water end uses requires manual surveys of residents, while generating labeled datasets requires manual logging of water use events. Both are labor intensive and difficult to achieve at any scale.

To address the limitations of previous studies, we sought to develop a new, fully automated machine learning technique for identifying and classifying end uses from water trace data that does not require a large, fully labeled dataset of water end uses for training or prior knowledge of the number of end uses inside a home for classifying events. We explored a semi-supervised approach, using a small set of labeled events from a single home to help in the classification of a much larger set of unlabeled events across multiple homes. The expense associated with collecting a large dataset of manually labeled events along with the fact that a small labeled dataset is not representative of the true variance of the data made development of a fully supervised model impractical. Semi-supervised learning makes it possible to combine the advantages of working with a small, labeled dataset to guide the learning process and a much larger, unlabeled dataset to increase the generalizability of the solution.
2.3. Methods

2.3.1. Case Study Design and Data Collection

The high temporal resolution data we used were collected at five homes drawn from a larger sample of 31 residential homes located within the Cities of Logan and Providence, Utah, USA. These cities made available to us their monthly water use data for residential customers. We ranked users based on their annual average water use computed from monthly records and divided them into classes of low (< 33 percentile), medium (33 – 66 percentile) and high (> 66 percentile) water users. From these classes, we randomly subsampled and invited potential participants to participate in our detailed data collection. Prospective participants were sent a letter in the mail inviting them to participate in this study. Of 200 letters sent, 11 participants responded positively and enrolled. Given the low response rate to mailed letters, an additional 20 participants were recruited and enrolled through word of mouth and targeted invitations. We achieved a sample size of 31 participants that broadly represent the spectrum of water users within Logan and Providence Cities.

For each participating household, we collected water use data using the home’s water meter for two weeks in the summer when outdoor water use was ongoing and another two weeks in the late fall/winter when there was no outdoor water use (four weeks total). We utilized low-cost electronic dataloggers with high temporal resolution data collection capabilities (Bastidas Pacheco et al. 2020a). These dataloggers were installed on the existing water meters at each home and recorded observations of water use with a 4 second data recording interval with pulse resolution that depended on the meter type and size.
For each participating household, we conducted a brief survey to identify the water end uses present in the home. We conducted regular visits to download the high frequency data and saved them as comma separated values (CSV) data files. Raw data were managed initially on a field laptop before being transferred to a secure file sharing system (for archival of original data files) and an operational database server (to support high performance queries and analysis) for secure and shared access among the research team. An anonymized version of the full high resolution data for all 31 homes that participated in our water metering study is available in the HydroShare data repository (Bastidas Pacheco et al. 2020b).

One difference among participants was meter type and size. For participants in Logan, Neptune T-10 meters were observed with pipe sizes of 0.625 or 1 inch. The meter sizes are described in inches with pulse resolutions in gallons to match manufacturer specifications for how these meters are sold in the U.S. T-10 meters produce a magnetic pulse detectable by our low-cost dataloggers every 0.03 L (0.008 gal) or 0.076 L (0.02 gal) of water consumed, respectively. In Providence, Master Meter bottom loading multi-jet meters were observed with pipe sizes of 0.625 or 1 inch. These meters generate a magnetic pulse every 0.076 L (0.02 gal) or 0.15 L (0.04 gal) of water consumed, respectively. Pulse resolutions of these meters were determined through laboratory testing at the Utah Water Research Laboratory (Bastidas Pacheco et al. 2020a).

Alongside the high temporal resolution water use data, we asked the resident of one participating household to manually label some water use events. Each time an end use was initiated inside the home, the type of end use and its start time were recorded. Matching the times of manually labeled events with the high resolution trace data enabled
us to better understand the shape of events for each water end use type, their characteristics, and, most importantly, the variables that contributed the most in distinguishing between events of different types. A total of 998 different single water end use events were manually labeled. We collected an initial dataset containing 538 events to serve as a training dataset, and we conducted an additional data collection period during which we collected 460 events to serve as a testing dataset. Besides the manually labeled events, we manually added irrigation events to the dataset since they were not labeled by the homeowner and they were easily distinguishable in the trace of water use.

For water end-use analysis, we selected a subset of five households with different meter sizes and types from the larger set of 31 sampled homes. Four of the households were selected because they had the highest number of residents and the highest average daily water use compared to other households with similar meter size and type. The fifth was selected as the household at which manually labeled events were recorded. The results reported in this paper are not meant to present a comprehensive analysis of water use behavior in all of the homes for which we collected data, but rather are focused on demonstrating the effectiveness of the disaggregation and classification tool we developed. All five households had outdoor water use and used sprinklers for irrigation. Table 2.2 summarizes the general characteristics of the selected households. Household IDs 1, 2, 3, 4, and 5 in Table 2.2 correspond to households 3, 11, 24, 27, and 19, respectively in the Bastidas Pacheco et al. (2020b) HydroShare dataset that contains data for all 31 sampled households. Identifiers have been changed here for convenience in referring to them in the text of this paper. To facilitate reproducibility of the results presented in this paper, the high resolution data for the 5 households listed in Table 2.2,
along with the manually labeled event data for Household 5 are available in a separate HydroShare resource (Attallah and Bastidas Pacheco 2021).

2.3.2. Data Cleansing

We cleansed the raw trace data for each residence using a low-pass filter (Eq. 1, Broesch 2008) to enhance the process of extracting events and their associated physical features (volume, duration, etc.). The filter modifies the raw trace data to accentuate event start and end times and the number of vertices of each event. Values output by the low-pass filter were rounded down to the nearest integer value. Fig. 2.2 shows an example of the output of this process. We chose a low-pass filter coupled with rounding because it retains the overall shape of events while removing or adjusting oscillations caused by the data recording interval and pulse resolution of the meter leading to more clearly recognizable events:

\[
P_j' = \begin{cases} 
\frac{\sum_{i=1}^{n} P_{j-i}' + \sum_{i=0}^{n} P_{j+i}'}{2n+1} & \text{for center data points} \\
\frac{\sum_{i=0}^{n} P_{j+i}'}{n+1} - 1 & \text{for edge data points}
\end{cases}
\]

where \(P_j'\) is the filtered data point at the \(j_{th}\) index, \(P_{j-i}'\) is the predecessor filtered data point value within the number of time periods \(n\), and \(P_{j+i}'\) is the original data point value at the \(j+1\) index within \(n\) periods of time. Edge points are the first and the last data point in the series, while center points are all other data points between the two edge data points. We used \(P_{j+i}'\) for the first data point, and \(P_{j-i}'\) for the last data point. For filter configuration, we used root mean square error (RMSE) to measure the difference between values predicted by the filter and observed values at different time periods \(n\). A value of 1 for \(n\) yielded the lowest value of RMSE, and thus was selected for the filter.
2.3.3. Event Disaggregation

The data cleansing process resulted in filtered data points that accentuate the shape of each event within the trace and make it easier to identify the beginning and end of each event. For example, in Fig. 2.2 the unfiltered trace data peak at 7 pulses near the beginning of the event, which might be mistaken as a different, short duration event superimposed on the longer event having values of 4 and 5 pulses. After filtering, the event is more identifiable as one event. From there, we identified all periods of non-zero flow in the unfiltered trace data as events. After that, we calculated several features for each event using either the filtered or the raw trace data (Table 2.3). The volume, duration, flowrate, and mode flowrate have been used in other water end use disaggregation studies and have been proven useful in separating and identifying events. To that list of features we added peak flowrate, peak flowrate frequency, mode flowrate frequency, root mean square (RMS), number of vertices, two irregularity measures, and a complexity measure because we noticed these additional features enhanced the disaggregation process.

RMS of each event was calculated using the formula suggested by Coppack (1990):

\[ RMS = \sqrt{\frac{1}{n} \sum_{i=1}^{n} FR_i^2} \]  

(2)

where RMS is the square root of the mean square of water use flowrate within an event, \( FR_i \) is the flow rate of filtered data at the \( i_{th} \) index, and \( n \) is the number of flow rate values within an event.

The disaggregation process then uses these features to classify events as single or overlapping and then to disaggregate overlapping events into single events. Before
Classifying events as single or overlapping, we first eliminated events with durations of one-time step and volume of 1 pulse. Given uncertainty around the nature of these events, we did not classify them. We observed single pulse, single time step faucet events in the set of events manually labeled by our study participant, but we also observed many other events of this type during time periods that were unlikely to be faucet events. Thus, where some other studies have assumed that these single pulse events are associated with leaks, we were unable to do so given that they occurred in the manually labeled data. Instead, we categorized these events as “unclassified” and note that unclassified events may result from leaks and/or single time step events that we were unable to classify further.

Classifying events as single or overlapping was carried out using a combination of features from Table 2.3, including the number of vertices, IR1, IR2, and CX, which are all calculated from the filtered data. These features reflect the shape, irregularity, and complexity of each event, and, hence, enable determination of whether an event is single or overlapping. Perfect, rectangular shaped events have a zero value for IR1 (RMS = mode flow rate). As the irregularity of an event increases, RMS and mode flow rate values become unequal and result in a non-zero value for IR1. A constant flow rate, single use event has a zero value for IR2 (peak flow rate = mode flow rate). When water use events overlap, the peak flow rate value and the mode flow rate value deviate from each other (IR2 becomes a non-zero value). We considered events to be irregular if both IR1 and IR2 were not equal to zero. Besides their irregular shapes, overlapping events are constituted by more than 4 vertices and have a CX value greater than one. After
calculating the complexity and the irregularity of each event originally identified in the trace, we identified overlapping and single use events using the following criteria:

\[
\text{Event} = \begin{cases} 
\text{Overlapping} & \text{if IR1} \neq 0 \text{ and IR2} \neq 0 \text{ and } CX > 1 \text{ and } V > 4 \\
\text{Single} & \text{all other events}
\end{cases}
\]  \hspace{1cm} (3)

the criteria suggest that for an event to be considered overlapping, it must meet the IR, CX, and V conditions all together. If one condition is not met, the event is considered a single use event. Fig. 2.3 depicts the process of classifying events as single use or overlapping events. Graphicly, most single use events (Fig. 2.3-A) are approximately rectangular and exhibit a constant flow rate throughout the entire event. IR1 and IR2 measures are zero for both events, and the CX measure is larger than one, hence both events are single use events. In Fig. 2.3-B the event exhibits a variable flow rate throughout the entire event. In this case, both IR1 and IR2 measures do not equal to zero, and the event is confined by more than 4 vertices, but the event violates the CX measure, where the frequency of the mode flow divided by the number of different vertices is less than 1. Thus, the event is classified as a single use event. Fig. 2.3-C shows an overlapping event that satisfies the IR1, IR2, CX, and V conditions all together, having more than 4 vertices, values of IR1 and IR2 that do not equal zero, and a CX value that is larger than 1.

Overlapping events identified in the first phase proceed to the second phase where they are split into their single event components using a sequential splitting procedure (Fig. 2.4). The first split is applied horizontally at the mode flow rate. The first identified sub-event is made up of any value that is less than or equal to the mode flow rate. After the first split, the remaining sub-events are considered to be new events whose features are estimated to test whether the event is a single use event. If yes, no more splitting is
needed, and the event proceeds to the clustering process. If not, the splitting procedure at
the mode flowrate continues in an iterative manner until all sub-events are single use
events.

In Fig. 2.4, the first split is applied on the mode flow rate value (3 LPM). Values
that are equal or less than 3 are added together to form event 1. After the first split, the
features of the newly-separated sub-events are recalculated and used to re-evaluate the
complexity and irregularity of each sub-event. The features of events 2 and 3 suggest that
they are both single events, hence no further splitting is performed on these two events.
However, event O1 is an overlapping event and can be further simplified. We applied the
second split on the mode flow rate of event O1 and re-evaluated the complexity and
irregularity of the resulting sub-events. This time, their features suggest that events 4 and
5 are single use events. The final split is applied to the last overlapping event in the series
that contains events 6 and 7. This process is run on the dataset for each house, and the
result of this iterative end use disaggregation process is a dataset of single use events
associated with their features described in Table 2.3 for each house. Creating an event
dataset for each house maintains consistency in end use types, water use behavior, and
the statistical distribution of water use events of each dataset. We acknowledge that there
may be a small number of overlapping events where the first event ends before the
second event ends. In these cases, the algorithm will split the second event into two
separate events. The separated events will still be classified correctly, but the count of
events will include an extra event for each of these instances.
2.3.4. Event Classification

Each single end use event output by the disaggregation process was assigned an end use category using a semi-supervised clustering and classification technique. We used clustering to identify outliers in the event dataset and classification to assign each non-outlier event to an end use category. This process was executed on the event datasets for each house, one house at a time.

2.3.5. Feature Scaling

The numeric values of the features listed in Table 2.3 are highly variable in units and range. Considering that most clustering algorithms use distance between data points (e.g., the Euclidean distance in a multidimensional space) as the measure of similarity, features with varying magnitudes will not be weighted equally in the distance calculations. To overcome this, we applied feature scaling prior to clustering. We explored the distributions for all features and observed that each of them exhibited a multimodal distribution (e.g., Fig. 2.5).

Since the feature distributions are not gaussian and we observed outlier data points in the distributions for some of the features, we used the RobustScaler function from the scikit-learn package for Python (https://scikit-learn.org) to scale the range and the magnitude of all features. The RobustScaler function scales the feature values by subtracting the median from each data point and then by dividing by the interquartile range (IQR) of the data. Mathematically, the RobustScaler can be expressed as:

\[
x_i' = \frac{x_i - Q_2(x)}{Q_3(x) - Q_1(x)}
\]  

(4)

where, \(x_i'\) is the scaled data point, \(x_i\) is the original data point, \(Q_1(x)\) is the original first quartile data point, \(Q_2(x)\) is the second quartile data point (median), and \(Q_3(x)\) is the
original third quartile data point. The output of this step in the process is a dataset of events for each house with scaled feature values.

2.3.6. Feature Selection

Including redundant or irrelevant features may inhibit clustering algorithm performance and can misguide results. To avoid this, we utilized a feature selection technique to subset a feature combination that produced the “best” clusters where the intra-cluster similarity was high and the inter-cluster similarity was low. There are several algorithms for selecting features, including Recursive Feature Elimination (RFE) (Guyon et al. 2002), Boruta (Kursa and Rudnicki 2010), and Genetic Algorithms (Whitley 1994). We used the Boruta algorithm because it was found to be more robust than other feature selection algorithms in a study that evaluated different feature selection methods (Degenhardt et al. 2017).

The Boruta algorithm uses a random forest-based classifier iteratively and attempts to capture all features that are relevant to the outcome variable (i.e., the predicted end use labels for each event). The algorithm determines which features to keep and which features to eliminate by first creating shadow features that are duplicates of the original features. After the shadow features are created, their values are shuffled to remove any potential correlation with the outcome variable. Next, the algorithm combines the original features with the shadow features into one dataset and runs the random forest classifier on the combined dataset multiple times (the default is 20). Boruta calculates an importance score for each feature in the combined dataset as the decrease in the classification accuracy of the outcome variable when a feature is excluded (mean decrease in accuracy of the outcome variable). The importance score for each original
feature in the dataset is then compared with a threshold value, defined as the highest importance score among the shadow features. If the feature’s importance score is higher than the threshold value, the feature scores a “hit.” If not, the feature scores a “no hit.” With each run, each feature is removed one at a time and its importance score is calculated. With the 20 run results, a feature that scores a “hit” in at least 95% of the total runs is deemed important. If the feature’s “hit” score is lower than 95% of the total runs, the feature is considered irrelevant and is eliminated from the dataset.

We used the implementation of the Boruta algorithm included in the scikit-learn contributed packages (Homola 2015; https://github.com/scikit-learn-contrib/boruta_py) on the manually labeled event dataset from Household 5. For the algorithm configuration, we defined the event label attribute as the outcome variable and the scaled features in Table 2.3 as the predictor variables. We used the default number of runs setting (20). The algorithm suggested that the mode flow rate, duration, volume, peak flow rate, average flow rate, and RMS features have the most impact in predicting the correct labels for events. Fig. 2.6 summarizes the output of the algorithm and the importance score for each feature. The color of the box in Fig. 2.6 indicates feature type. Green colored boxes are original features, while blue colored boxes are shadow features. Features with an importance score less than the maximum shadow feature score do not appear in the plot (e.g., CX). The output of the feature selection process is a dataset of events for each house with a reduced set of features that can be used in the classification process.

2.3.7. Semi-Supervised Classification

The final step in the classification process is clustering and label assigning, for which we used a “cluster-then-label” semi-supervised technique. This technique utilizes
unsupervised clustering to detect and eliminate outlier events that deviate from other events in the data space and then predicts the category label of each remaining unlabeled event in the developed clusters using semi-supervised classification. Similar approaches have been used by Tanha et al. (2017), Gan et al. (2013), and Weston et al. (2005). Due to the peculiar variation in the features of end use events of the same type caused by residents’ water use behavior (e.g., long versus short showers, faucet partially versus fully open), clusters of events may not have convex, isotropic shapes, which can be problematic for partition-based and hierarchical clustering techniques. Density-based clustering, on the other hand, identifies clusters as groups of data points of high density and is capable of identifying clusters of any shape. On account of this, we used the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm (Keim et al. 1996).

DBSCAN has two types of outputs: 1) outliers: scattered low density events that do not fall within an identified cluster, and 2) event clusters: definitive clusters made up of high-density core and border events that include only one type of end use. DBSCAN was run on the event dataset consisting of the subset of scaled event features selected by the Boruta algorithm to extract event clusters for each house. We observed that outliers were typically abnormal water use events (e.g., a very long shower or a dual toilet flush) that exhibit features that are different than anticipated behavior. Outliers identified by DBSCAN were added to the list of events that we did not classify, whereas events that were assigned to a specified cluster proceeded to the final step in the procedure, label assigning.
For assigning labels to unlabeled events, we tested several supervised classifiers in the scikit-learn package on the manually labeled training dataset, including quadratic discrimination analysis (QDA), multilayer perceptron (MLP), adaptive boost (ADA), random forest (RFC), gaussian naive Bayes (GNB), classification and decision tree (CART), Gaussian process (GPC), and support vector machine (SVM). A description of each of these classifiers can be found in the scikit-learn documentation (https://scikit-learn.org/stable/user_guide.html). Given the limited nature of our manually labeled sample of events, we used k-fold cross-validation analysis to test the accuracy of each classifier (Camacho and Ferrer 2012). The k-fold method splits the dataset into a number (k) of subsets (folds). Each data point is assigned to an individual fold and stays in that fold for the duration of the procedure. To begin the procedure, one fold is reserved for testing (called the hold-out), and the rest of the folds are combined and used as a training dataset for fitting a model. The accuracy of the fitted model is then evaluated on the data in the hold-out fold. The accuracy score of the model is calculated as the number of correctly predicted data points divided by the total number of data points within the hold-out fold. The accuracy score is then retained while the model is discarded. This process is repeated k times, each time using a different hold-out fold. Thus, each fold serves as the hold-out fold once and is used as part of the training dataset to fit a model k-1 times. For k-fold configuration, we used 10 folds with each fold holding 10% of the total data.

The average accuracy score of each tested classifier was calculated as the summation of accuracy scores for the 10 models developed using that classifier divided by the number of folds (10). By doing this, we were able to create many different models using each of the classifiers and test them on different subsets of the training data rather
than doing a single evaluation on one testing dataset. Consistently high accuracy scores across all folds indicates that the classifier performs well, and we can be confident that the trained algorithm will always produce a similar performance. We applied k-fold cross validation to all of the classifiers mentioned above and evaluated the average and standard deviation of their accuracy scores. We found that the random forest classifier produced the highest accuracy score and selected it for use, although it was only slightly better than the decision tree classifier (Fig. 2.7). Both classifiers performed consistently across all tested folds as indicated by their small standard deviation values.

We used the random forest classifier on the manually labeled training dataset to develop an initial model that was the same for all of the houses. We then implemented an iterative, self-training procedure to develop a final model for each house. The initial random forest classifier was used to predict the labels for all of a home’s unlabeled events. We used the similarity score metric function in the random forest classifier to quantify the quality of the predicted label for each event in the home’s dataset. The similarity score function uses the coefficient of determination ($R^2$) of the newly-labeled events to estimate the probability of each newly-labeled event being classified correctly when compared to the manually labeled events. On each iteration, a subset of the newly-labeled events with a similarity score of at least 90%, together with their predicted labels, were added to the labeled dataset. The random forest classifier was then re-trained on the larger set of labeled events to produce a new model for the home. This procedure was repeated, and the model was iteratively trained until all events with similarity scores of at least 90% with their predicted labels are added to the set of labeled events. The enhanced model based on the original and newly labeled events was then used to predict the labels
of the remaining unlabeled events. Each household required two to three iterations to reach a point where no new events met the 90% similarity score criterion. The pseudo-code of the algorithm can be formally described as follows:

Let \( L = \{ (x_i, y_i) \}_{i=1}^l \) be the input manually labeled training dataset where \( x \) is the set of labeled events, \( y \) is the label set, and \( l \) is the number of labeled events. Let 
\[ U = \{ x_j \}_{j=l+1}^{l+u} \]
be the set of \( u \) unlabeled events for a home.

1. Apply DBSCAN on \( U \) to generate an initial set of clusters
2. Using the initial DBSCAN clusters, identify outlier data points and remove them from \( U \)
3. Train a predictive random forest classifier \( f \) using \( L \) as a training dataset
4. Predict the label of all unlabeled events in \( U \)
5. Calculate the similarity score for the newly labeled events in \( U \)
6. Add \( \{ (x, f(x)) \mid x \in S \} \) to \( L \), where \( S \) is the subset of newly labeled events in \( U \) with similarity score > 90%
7. Remove \( S \) from \( U \)
8. Retrain the classifier \( f \) using the enhanced training dataset \( L \)
9. Repeat steps 4-8 until there is no \( (x, f(x)) \) with similarity score > 90% in \( U \), or until \( U = \emptyset \)

The output of this process is a random forest classifier model for a home that can be applied to the set of unclassified events for that home to predict their event type labels.

We used the manually labeled testing dataset for testing the accuracy of the final model developed for Household 5.

2.3.8. Software Design and Implementation

The tool was designed and developed using the Python programming language as a single script that can be executed using any Python programming environment. The tool was developed using the SciPy, Pandas, NumPy, and scikit-learn packages for Python.

The input to the tool is a comma-separated values (CSV) file that contains high resolution meter data collected every 4 seconds for an individual home and a pulse resolution conversion factor that corresponds to the size of the meter (Bastidas et al. 2020b). The input file has three fields: Time, Record, and Pulses. The Time field contains the
timestamps at which individual observations of water use were recorded. The Record field is a sequential numbering attribute that uniquely identifies rows in the dataset. The Pulses field contains the number of magnetic pulses recorded by the datalogger within a 4-second interval, where each pulse corresponds to a known volume of water use. The first three rows of the file are reserved for a metadata header including the site number, datalogger ID, and meter size. Data start on the fourth row. The length of the file is not restricted.

The tool reads the data file and loads it into a date/time-indexed Pandas data frame, after which the filtering process on raw water use data is applied. The disaggregation process outputs four different CSV files. The first file contains single water use events, the second contains unclassified water use events, the third contains overlapping water use events, and the fourth contains the disaggregated single water use sub-events derived from overlapping events. All output files from the disaggregation process include values for the features listed in Table 2.3. The tool then proceeds with the classification process. The classification process outputs one CSV file that contains classified and labeled water use events along with the features of each event, including duration, volume, flow rate, peak flow rate, and mode flow rate.

2.4. Results and Discussion

After data cleansing and removal of single pulse events, the tool identified 16,420 unprocessed water use events retrieved from the five households (excluding unclassified events). After breaking down overlapping events into their single use event components, the total number of events increased to 18,491. The average processing time for all filtering, disaggregation, and classification operations per single day of data for one
household was approximately 50 seconds. Unclassified events (including single pulse events) accounted for only 3% of the total water use volume across the five houses; however, these events appeared frequently in the trace data and accounted for more than 40% of all recorded events in some households. Given the uncertainty around the nature of unclassified events and their relatively small overall volume, we excluded them from further analysis.

2.4.1. End Use Classification Accuracy

We used the testing dataset consisting of manually labeled events from Household 5 to quantify the classification accuracy of the developed model at Household 5. For consistency with other end use classification studies, we calculated model accuracy as the fraction of events whose end use category was correctly predicted by the model when compared to the manual labels (Table 2.4). The overall accuracy of the classification process was 98.2%, with the highest accuracy observed (100%) for clothes washer and shower events. These results represent a significant improvement compared to the 88.4 faucet accuracy reported by Autoflow v3.1 (Yang et al. 2018). We also used the F-1 score metric to quantify the model accuracy (Table 2.4). The weighted average F-1 score for all end use categories was 98%, where weights were assigned based on the number of events in each end use category. The disparity between the classification accuracy for bathtub events (67%) versus the F-1 score metric (80%) indicates that the model may not be finding all bathtub events, but that those events classified by the model as bathtub events are likely to be correct. We also illustrate classification results using a confusion matrix (Fig. 2.8), which is a summary of predicted labels for water end use events compared to actual labels. The diagonal elements represent the number of events for which the
predicted label is equal to the true label, while off diagonal elements represent misclassified events. As shown in Fig. 2.8, all clothes washer and shower events were classified correctly. The classification accuracy for bathtub events was lower than the other event types. Of the bathtub events in the testing dataset that were incorrectly labeled, 80% of them (4 out of 5) were classified as shower events. We attribute this to similarity between the characteristics of bathtub and shower events and the small number of bathtub events in the manually labeled training dataset.

For sites where no manually labeled events were available, it was not possible to directly evaluate the accuracy of the classifier. Instead, we applied a manual verification procedure consisting of examining the characteristics of extracted events and their raw trace data for each site. While the characteristics of water fixtures may vary from one household to another, the data show that their overall characteristics are relatively close (e.g., a toilet flush in one house might have an average volume around 6 LPF versus 7 LPF in a different house). Events extracted from all homes were compiled into one file. Then, for each home, events within each end use type were sorted according to their features in a step-by-step procedure (e.g., sorting toilet events for a home by their volumes in descending order and then ascending order). We then investigated differences between events at the endpoints and events in the middle of the distribution of each end use type for each home. We assumed that mislabeled events are more likely to occur at the endpoints of the feature distributions of each end use type. The number of events we examined for each end use type varied depending on how similar events at the endpoints of the distribution for each end use type compared to the rest of events. We examined between 20 to 50 events per end use type per house. Besides the examination of extracted
classified events, we also examined the raw trace data for the same events to verify similarities with other events having the same label. As a last step in the verification procedure, we re-labeled misclassified water events according to where they were most likely to belong, based on the analyst’s decision, considering all the elements described above. Without considering unclassified events, on average, changes were made to less than 2% of the labels assigned by the algorithm at each site.

The information collected at each home during enrollment helped us confirm the verification process and catch mislabeled events (e.g., the algorithm labeling bathtub events that were similar to clothes washer events in homes where bathtubs were not present or used). We retrained these homes using a training dataset that did not include bathtub events. The developed method seeks to classify water end use events without the need for labelled events at each home. Thus, the manual evaluations we did were aimed at ensuring the quality of our analyses.

While the accuracies we observed for household 5 were similar to those reported by Yang et al. (2018), it should be noted that the accuracy values are not exactly comparable. We estimated the classification accuracy of the tool using a relatively small set of events that were manually labeled by a study participant. Thus, we were confident in the event labels. The Autoflow classification accuracy was estimated based on a much larger set of events that were labeled by an analyst using the TraceWizard software (Yang et al. 2018). While we cannot verify the actual accuracy of the labels assigned by the analyst, the Autoflow software was able to match those labels with a high level of accuracy. Using the set of manually labeled events, we were able to identify the features of dishwasher events. However, we found that their features were indistinguishable from
faucet events, so we were not able to separate dishwasher and faucet events. Thus, dishwasher events were grouped with faucet events (see Appendix A.6 for more details).

2.4.2. Overlapping Events

A similar verification procedure was applied to overlapping events, where we manually verified the accuracy of our method in identifying and separating overlapping events by visually inspecting events collected at different sites. A total of 515 of the original 16,420 events we extracted from the water trace data for five households covering a monitoring period of four weeks per household were identified as overlapping events. The overlapping events were disaggregated into 2,071 single events, bringing the total number of single events to 18,491. We visually inspected 20% of all overlapping events identified by the algorithm and verified that all of them met the algorithm’s criteria for overlapping events and that the splits were applied correctly. Thus, we are confident that the algorithm is correctly identifying overlapping events and that the splitting procedure works as designed.

2.4.3. Overall Water Use and Individual End Uses

For completeness and for comparison with other studies, we examined the indoor and outdoor water use for each of the studied households. We also characterized individual end uses for each home to quantify the distribution of volume and frequency of use across the different end uses along with potential seasonal variation. These analyses are provided as supplemental materials in Appendix A for overall water use and Appendix B for detailed end use analysis.
2.5. Conclusions

We presented a new, open source, semi-supervised water end use disaggregation and classification tool that can break down the total water use observed at the household level into different end uses. The tool uses non-intrusive monitoring data collected at high temporal resolution from a residential home’s water meter along with machine learning techniques to disaggregate water use into discrete end use events. This work was driven by the fact that, for most other studies that have worked on end use disaggregation algorithms, neither the source code nor the data are available for testing or further advancement. It is our hope that the code and anonymized data we have openly shared can be a platform for advancing the availability and functionality of open tools for water end use disaggregation studies.

Unlike other end use disaggregation techniques, we used a semi-supervised classification approach to overcome the challenges associated with classifying events. While our approach required an initial set of manually labeled events, which can be expensive and difficult to collect, we employed this relatively small number of labeled events to show how a semi-supervised model can be developed and used for classifying events from any residential home. The data we collected and the developed water end use disaggregation tool are now available for potential use by others who may want to use similar end use classification approaches. Additionally, where some other studies validated their results by comparing to events classified in post-processing by a data technician (which may or may not be correct depending on the dataset and experience level of the technician), our approach used actual events manually labeled by a study participant. The openly-available dataset of manually labeled events paired with the
corresponding high resolution water use data from the meter that we produced could be expanded by other investigators in other geographical areas to produce a much larger set of labeled data to produce a large corpus of data for training machine learning models as has been done in other fields of study (e.g., the ImageNet dataset, Deng et al. 2009). Our intermediate results (e.g., the disaggregated events and their features) could also be repurposed for testing other clustering or classification techniques.

In our case study application, the number of events participating in overlapping events accounted for approximately 11% of all recorded events and contributed to about 28% of total water use volume, demonstrating the importance of handling them correctly in the disaggregation process. Our disaggregation approach extends what has been done in other studies and enabled us to separate the overlapping events into single events prior to classification. Executing the disaggregation and classification algorithms on data from different households with different meter sizes and types and different water use characteristics showed that the algorithm can be used across the meter types and sizes we tested, which should mean that it can be used across a wide range of residential meter types and sizes, although further testing with new datasets would be needed to confirm this. While we manually verified that the algorithm correctly identified overlapping events according to our criteria and separated them according to the rules we set, a new study with data collection focused specifically on recording and labeling overlapping events could provide benchmark datasets for further testing the accuracy of algorithms for disaggregating overlapping events.

The tool provides significant benefits for water consumers and water utilities. For consumers, it can provide information about how and when they are consuming water.
Our detailed results illustrate both behavioral differences in water use across households and technical differences based on the performance of the water fixtures within the homes, which we found were not always meeting plumbing standards (Appendix A and B). A compilation of household-level water end use data could assist water utilities in identifying opportunities for incentive programs to encourage water conservation and monitoring effectiveness of those programs. For researchers, the open nature of the data collection hardware and the methods described in this paper for end use disaggregation present a new opportunity for advancing beyond the limitations imposed by lack of available data and the proprietary nature of existing software. The code we have provided for analyzing the disaggregated water use data can serve as a base for further work.

The work described in this paper builds on other end use disaggregation studies and, like those other studies, demonstrates how water end use studies can provide detailed data to inform water resource management in areas where water is scarce. By characterizing how residential water is utilized inside households, these results will provide information that may be useful for city engineers and local water managers as they operate existing infrastructure, formulate plans to increase the efficiency of current water supply and distribution infrastructure, as well as in planning for future improvements. Indeed, understanding water use at the end use level is essential for gaining insight into how, when, and why water is being used. This information is, in turn, critical for water managers in identifying opportunities for conservation, assessing the impact of conservation programs, forecasting demand, and determining how water use patterns may change over time in response to population growth, demographic shifts, and
improvements in technology. Supplying this type of information to water users can also be a tool for impacting water use behavior and managing demand.

Detailed information about water use is needed by Utah, other states in the U.S., and other similar areas throughout the world to better project future water needs. High resolution data from metered households can provide valuable information on daily and seasonal consumption patterns, especially when coupled with both structural (e.g., lot size and landscaping characteristics, appliance and fixture age, etc.) and socio-demographic information (e.g., age, family size, income level, ethnicity, etc.) about the household. Future water use projections can then be made based on the demographics of projected growth and not just on the projected number of people. In addition to more accurate demand forecasting, these data provide a potential opportunity for water utilities to reduce operational costs now and in the future through efficiency gains and deferral of upgrades.

Data Availability Statement

Some or all data, models, or code generated or used during the study are available in a repository online in accordance with funder data retention policies.

The CIWS Disaggregator software is open-source, released under the Creative Commons Attribution CC BY license, and available in the HydroShare repository (Attallah and Bastidas Pacheco 2021). Documentation of hardware and software requirements, Jupyter notebooks with examples of workflows of each part of the code, and instructions for running the code are provided in the HydroShare resource. The high resolution water use dataset containing the data for all 31 houses we sampled is available in HydroShare (Bastidas Pacheco et al. 2020b). The manually labeled event dataset and
the processed event data resulting from the case study analyses in this paper are also available in HydroShare (Attallah and Bastidas Pacheco 2021). While we anticipate that other geographical areas may have different meter configurations, we anticipate that the disaggregation and classification procedure described in this paper can be applied to high resolution metering data collected anywhere. Specific instructions for implementing the code on our dataset (or other similar datasets) are provided in HydroShare (Attallah and Bastidas Pacheco 2021).

Reproducible Results

Amber Spackman Jones (Utah State University, Utah) downloaded the CIWS-Disaggregator code and input dataset from HydroShare (Attallah and Bastidas Pacheco 2021). She installed the code using instructions available in the HydroShare resource and ran the CIWS-Disaggregator using the input data set provided in the HydroShare resource to reproduced results in Figures A1 and A2, and Tables A1 to A3 in the Appendix A section, and Figures B1 to B5 and Tables B1 to B5 in the Appendix B section.

Acknowledgments

This research was funded by the United States National Science Foundation under grant number 1552444. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional financial support was provided by the Utah Water Research Laboratory at Utah State University. We acknowledge Logan and Providence Cities for their cooperation in the realization of the field data collection campaigns. The authors also gratefully acknowledge the homeowners who participated in our data collection efforts.
REFERENCES


Nguyen, K. A., Stewart, R. A., Zhang, H. (2013). An intelligent pattern recognition model to automate the categorisation of residential water end-use events,


Utah Division of Water Resources (2005). Municipal and Industrial Water Use in Utah, https://static1.squarespace.com/static/530e3f55e4b088e83bb95f08/t/562145e8e4bb95f08/t/562145e8e4bb95f08/1445021160802/DNR_WaterUse.pdf.


Tables

Table 2.1. Event disaggregation process for different software tools. If a tool does not use any feature in the disaggregation, does not attempt to identify whether the event is single or overlapping, or does not attempt to breakdown overlapping events, N/A is reported.

<table>
<thead>
<tr>
<th>Software Tool</th>
<th>Features Incorporated</th>
<th>Overlapping Event Identification Method</th>
<th>Overlapping Event Separation Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace Wizard</td>
<td>Start time, duration, volume, maximum flow rate, and mode flow rate</td>
<td>Decision Tree algorithm</td>
<td>N/A</td>
</tr>
<tr>
<td>Identiflow</td>
<td>Duration, volume, average flow and maximum flow</td>
<td>Decision Tree algorithm</td>
<td>Decision Tree algorithm, and volume threshold value</td>
</tr>
<tr>
<td>Autoflow</td>
<td>Volume, duration, maximum flow-rate, mode flow rate, frequency of mode flow-rate, magnitude of initial flow-rate rise, magnitude of flow-rate drop at the end of event, gradient of initial flow-rate rise, and gradient of flow-rate drop at the end of the event</td>
<td>A hybrid combination of Hidden Markov Model (HMM), Artificial Neural Networks (ANN), and the Dynamic Time Warping (DTW) algorithms</td>
<td>A hybrid combination of Hidden Markov Model (HMM) and Artificial Neural Networks (ANN) algorithms</td>
</tr>
<tr>
<td>HydroSense</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Soft Computing Technique</td>
<td>Total volume, average flow rate, and the number of vertices</td>
<td>Number of vertices</td>
<td>Gradient flow rate change and average flow rate threshold value</td>
</tr>
</tbody>
</table>

Table 2.2. Characteristics of the five households selected for water end use analysis.

<table>
<thead>
<tr>
<th>Household ID</th>
<th>Meter size (inch)</th>
<th>Meter type</th>
<th>Meter resolution (L/Pulse)</th>
<th>Number of residents</th>
<th>Number of bathrooms</th>
<th>Irrigation system</th>
<th>Legal property size (m²)</th>
<th>Building size (m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5/8</td>
<td>Neptune T-10</td>
<td>0.033</td>
<td>4</td>
<td>3.5a</td>
<td>Manual</td>
<td>1,133</td>
<td>138</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Neptune T-10</td>
<td>0.126</td>
<td>4</td>
<td>3.5</td>
<td>Automatic</td>
<td>890</td>
<td>136</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Master Meter BL</td>
<td>0.157</td>
<td>6</td>
<td>3.5</td>
<td>Automatic</td>
<td>1,052</td>
<td>144</td>
</tr>
<tr>
<td>4</td>
<td>5/8</td>
<td>Master Meter BL</td>
<td>0.096</td>
<td>7</td>
<td>3.5</td>
<td>Automatic</td>
<td>3,116</td>
<td>300</td>
</tr>
<tr>
<td>5b</td>
<td>1</td>
<td>Master Meter BL</td>
<td>0.157</td>
<td>5</td>
<td>3</td>
<td>Automatic</td>
<td>1,133</td>
<td>128</td>
</tr>
</tbody>
</table>

a A half bathroom contains only a toilet and sink, but no bathtub or shower.
b Manually labeled events were collected for Household 5.
Table 2.3. Water end use event features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Definition</th>
<th>Data used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume (L)</td>
<td>The summation of water use volume between the beginning and end of a set of consecutive non-zero values</td>
<td>Raw</td>
</tr>
<tr>
<td>Start time</td>
<td>The time at which the water use volume changes from zero to any positive value</td>
<td>Raw</td>
</tr>
<tr>
<td>End time</td>
<td>The time at which the water use volume transitions back to zero</td>
<td>Raw</td>
</tr>
<tr>
<td>Duration (min)</td>
<td>The elapsed time of the event calculated as the difference between start and end times</td>
<td>Raw</td>
</tr>
<tr>
<td>Flow rate (LPM)</td>
<td>The volume of an event divided by its duration</td>
<td>Raw</td>
</tr>
<tr>
<td>Peak flow rate (LPM) – original</td>
<td>The maximum rate of water flow for any time step within the event</td>
<td>Raw</td>
</tr>
<tr>
<td>Peak flow rate (LPM) – filtered</td>
<td>The maximum rate of water flow for any time step within the event</td>
<td>Filtered</td>
</tr>
<tr>
<td>Peak flow rate frequency – original</td>
<td>The number of occurrences of the maximum flow rate within an event</td>
<td>Raw</td>
</tr>
<tr>
<td>Peak flow rate frequency – filtered</td>
<td>The number of occurrences of the maximum flow rate within an event</td>
<td>Filtered</td>
</tr>
<tr>
<td>Mode flow rate (LPM)</td>
<td>The flow rate value that appears most often within an event</td>
<td>Filtered</td>
</tr>
<tr>
<td>Mode flow rate count</td>
<td>The number of occurrences of the mode flow rate within an event</td>
<td>Filtered</td>
</tr>
<tr>
<td>Root mean square (RMS)</td>
<td>The square root of the mean square of flow rate (the arithmetic mean of the squares of the flow rates within the event)</td>
<td>Filtered</td>
</tr>
<tr>
<td>Number of vertices (V)</td>
<td>The number of points where the flow rate changes from one non-zero value to another non-zero value within the same event</td>
<td>Filtered</td>
</tr>
<tr>
<td>Irregularity measure I (IR1) (LPM)</td>
<td>The difference between the mode flow rate and the RMS value</td>
<td>Filtered</td>
</tr>
<tr>
<td>Irregularity measure II (IR2) (LPM)</td>
<td>The difference between peak flow rate and mode flow rate value</td>
<td>Filtered</td>
</tr>
<tr>
<td>Complexity measure (CX)</td>
<td>The count of the mode flow rate divided by the number of vertices</td>
<td>Filtered</td>
</tr>
</tbody>
</table>

Table 2.4. Classification accuracy and F1-score measures for each end use type. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

<table>
<thead>
<tr>
<th>End use type</th>
<th>Number of events</th>
<th>Classification accuracy (%)</th>
<th>F1-scores (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathtub</td>
<td>15</td>
<td>66.7</td>
<td>80</td>
</tr>
<tr>
<td>Clothes washer</td>
<td>24</td>
<td>100</td>
<td>98</td>
</tr>
<tr>
<td>Faucet and Dishwasher</td>
<td>190</td>
<td>98.9</td>
<td>99</td>
</tr>
<tr>
<td>Shower</td>
<td>60</td>
<td>100</td>
<td>96</td>
</tr>
<tr>
<td>Toilet</td>
<td>171</td>
<td>99.4</td>
<td>99</td>
</tr>
<tr>
<td>All categories</td>
<td>460</td>
<td>98.2</td>
<td>98</td>
</tr>
</tbody>
</table>
Figures

Figure 2.1. Water end use disaggregation methodology.

Figure 2.2. Example of data filtering output for a single water use event. Oscillations in the original data are caused by the combination of the data recording interval and pulse resolution of the water meter.
Figure 2.3. Examples of single and overlapping events. Panel A shows multiple single use events. Panel B shows an irregular single use event. Panel C shows overlapping events.

Figure 2.4. Illustration of the splitting procedure for overlapping events.
Figure 2.5. Distributions of volume, duration, and flowrate features. Due to the high discrepancies in the frequency of events across the feature values, log scale y-axes were used to improve readability.

Figure 2.6. Boruta algorithm output. Box plots represent the distribution of importance scores over the 20 Boruta runs.
Figure 2.7. Accuracy of different supervised classifiers. Error bars represent the standard deviation of the 10 accuracy score values.

Figure 2.8. Confusion matrix results for each end use type.
CHAPTER 3
RESIDENTIAL WATER METERS AS EDGE COMPUTING NODES:
DISAGGREGATING END USES AND CREATING ACTIONABLE INFORMATION AT THE EDGE¹

Abstract

We present a new, open source, computationally capable datalogger for collecting and analyzing high temporal resolution residential water use data. Using this device, analyses like execution of water end use disaggregation algorithms or other data analytics can be performed directly at an analog residential water meter without disrupting their operation, effectively transforming existing water meters into smart, edge computing devices. Computation of water use summaries and classified water end use events directly on the meter minimizes data transmission requirements, reduces requirements for centralized data storage and processing, and reduces latency between data collection and generation of decision-relevant information. The datalogger couples an Arduino microcontroller board for data acquisition with a Raspberry Pi computer that serves as a computational resource. The computational node was developed and calibrated at the Utah Water Research Laboratory (UWRL) and was deployed for testing on the water meter for a single family residential home in Providence City, Utah, USA. Results from field deployments are presented to demonstrate the data collection accuracy, computational functionality, power requirements, communication capabilities, and

applicability of the system. The computational node’s hardware design and software are open source, available for potential reuse, and can be adapted to specific research needs.

3.1. Introduction

Commercial “smart” or “intelligent” metering systems promise remote recording of water use at high temporal resolution with the potential for creating decision-relevant information (e.g., autonomously created reports and summary data products) for both water providers and consumers. However, commercially available smart meters have not yet been widely adopted in the U.S. for several reasons. First, replacing existing, analog meters with smart meters is expensive, labor-intensive, and disruptive. Second, smart meters produce “Big Data” with high volume and velocity [1]. Extracting decision-relevant information from the large volume of data produced by smart meters involves several unsolved challenges, including a critical shortage of professionals capable of working with “Big Data” to extract full value from smart metering systems. This is an instance where “Big Data” needs to be “shrunk” (i.e., summarized and mined) into information relevant to both water suppliers and consumers. The cyberinfrastructure, algorithms, and technologies to do this have not yet been well developed. In fact, many commercial smart meters collect data with high frequency but are not really “smart” in that they do not yet have the supporting cyberinfrastructure needed for fully utilizing the high frequency data they produce. Furthermore, some commercial smart metering systems limit data recording frequency to hourly intervals to reduce data volume and avoid data storage and communication bandwidth challenges. Hourly intervals are not frequent enough to identify individual water end uses (e.g., showers, toilets, faucets, etc.),
limiting the utility of the data for use in understanding water use behavior and targeting efficiency measures.

Previous applications of smart metering data and associated research studies have necessarily focused on the small number of cases where cities have upgraded to newer electronic meters or where individual dataloggers can be deployed to existing meters to collect high temporal resolution data. Most of these have been conducted in Australia [2,3,4,5,6] and the United States [7,8,9,10,11,12,13,14,15]. In these studies, high-resolution water use data were collected in the field and then transferred to a centralized location for post-processing to examine residential water use behavior.

In the aforementioned non-intrusive smart metering studies [2,3,4,5,6,7,8,9,10,11,12,13,14,15], several water end use disaggregation and classification algorithms were adopted to break down the total water use registered on the household’s main meter into different water end use categories. Regardless of the algorithm adopted, the primary focus of all of these studies was to better understand the water use behavior of residential users. For example, using disaggregated and classified water end use events, references [2,6] developed a theoretical integrated water use model for understanding household water consumption, references [3,5,12,13,14] evaluated the effectiveness of implemented water demand management and rebate programs, reference [4] reconciled differences between perceived and actual residential end use water consumption, reference [7] presented an open source, low cost monitoring system for the collection of high-resolution water data on residential water meters, reference [8] estimated the probability of water fixtures being used during peak hours, references [9,15] provided a detailed understanding of how water is being used inside residential
settings, and reference [10] estimated the price elasticity of water demand with water end use data.

Transferring data produced by smart water meters to a centralized location for post-processing, particularly for meters or dataloggers collecting data at a high enough temporal resolution to support end use disaggregation, has three significant limitations: (1) available bandwidth of conventional telemetry systems may be inadequate for transferring the large volume of data produced to a centralized location for post-processing, requiring technicians to visit sites to manually download data, (2) the water providing utility may not have sophisticated information technology infrastructure available to them to enable data post-processing, and (3) the utility may also lack dedicated staff and technical expertise needed to employ end use disaggregation algorithms or other sophisticated analyses.

A potential alternative to centralized information systems is to use a distributed approach, where data processing is performed at or near where the data are collected to extract and transmit only actionable data products to a centralized location. This distributed, or edge, computing approach is aimed at reducing the data management and computational burden associated with tasks such as water end-use disaggregation. By mining and summarizing the Big Data produced by smart meters at the site of data collection, required transmission bandwidth can be minimized, and derived data products can be more readily created and used to inform and improve water system management [16].

Edge computing is a distributed computing paradigm focused on bringing computing as close to the source of data as possible. Edge computing promises a range of
benefits for smart Internet of Things (IoT) applications and use cases across a variety of industries. Some of the most obvious benefits of edge computing include its ability to increase network performance by processing data closer to where it is collected and reducing or eliminating the physical distance over which data must travel. Because computational tasks are performed close to where data are collected, results can be transferred to other devices with less computing power (e.g., a smart, in-home display or a centralized database system used by a public utility). This may reduce latency—defined here as the time between when the data are collected and when actionable information extracted from the data is available for use. It may also reduce the need for centralized computational resources for processing data [17].

From a security perspective, the rapid spread of edge computing devices increases the overall attack surface for networks. However, because edge computing distributes data processing and storage across a wide range of devices, it is difficult for any disruption to take down the entire network, which can provide a more secure and reliable architecture for many use cases. In addition to the speed and security advantages, edge computing offers additional advantages through scalability. Because computation is done at the edge of the network, adding additional edge devices expands computing capability without imposing large data storage and computational burdens on a centralized infrastructure and without drastically increasing network bandwidth requirements for data transfer. Additionally, since processed data products arrive at a centralized location in a usable format, requirements on water utility staff are minimized [18].

In the context of edge computing, sensor motes, or nodes, are small, affordable, low-power computer boards or microcontrollers with a radio for wireless communication
Sensor motes are capable of collecting data, processing it, packaging it, and transferring it to a remote location [20]. In the current market, there are many commercially available sensor mote platforms, with some of them providing significant computational capabilities. However, most available platforms are general purpose and require significant work to adapt them for specific applications (i.e., collecting and processing data from a residential water meter).

For educational, research, and prototyping work, the Raspberry Pi platform [21], which consists of a Linux-based, single-board computer, has proven to be low cost (~$35), reliable, and adaptable for many different applications. Several groups have investigated the computational capabilities of the Raspberry Pi. For example, reference [22] developed an implementation of Multi-label classification and Random Kitchen Sink data mining algorithms on a Raspberry Pi computer using Mathematica. Implementation of Random Kitchen Sink algorithm on the Raspberry Pi computer using Mathematica improved the accuracy of Multi-label classification, reduced the code required for implementing data mining algorithms, and improved the memory usage when using large datasets. Reference [23] developed a 64-node computational cluster using a Raspberry Pi computer. Compared to conventional data-center based clusters, the computational cluster developed in [23] is low cost and low-power, portable due to its small size and weight, and has its own ambient cooling system. However, the Raspberry Pi computer’s usefulness as an edge computing device is not only because of its computational capabilities but also its ability to interface with a variety of sensors for data collection and communication peripherals for transmitting data. With available off-the-shelf electronic
components, it is now much easier to design and prototype low cost and low-power devices capable of data collection, computation, and communication tasks.

In this paper, we describe the design and testing of an open source datalogger and computational node that uses the data collection and computational capabilities of modern, single-board microcontrollers and computers to turn existing, analog, and residential water meters (the vast majority of meters in use today) into battery-powered, edge computational nodes that not only collect and store high-frequency flow data but also execute algorithms for disaggregating metered flow into individual water end uses (e.g., summary totals for toilets, showers, clothes washer, etc.). The computational node is also capable of transmitting raw data and/or actionable data products to a centralized location for further analysis, interpretation, and use.

We developed the computational node as part of a larger effort aimed at developing Cyberinfrastructure for Intelligent Water Supply (CIWS). The design of the computational node builds upon our earlier work in building a simpler datalogger capable of recording data at a user-configurable temporal resolution as high as every 1 s (CIWS datalogger, [7]). Using the CIWS datalogger, data must be manually downloaded in the field and then post processed to generate useful information. For example, water end use events can be extracted from the raw trace data and classified into water end use categories using algorithms such as the CIWS disaggregator algorithm developed by [24]. To the data logging capabilities of the CIWS datalogger, the CIWS computational node described here adds the ability to run code designed to post process the data locally on the node as well as the capability to transmit raw data and/or processed data products over the Internet to a remote server. This required: (1) substantial new work on an entirely new
hardware design that couples an Arduino-based data collection device with a Raspberry Pi computer to enable both data collection and edge computing capabilities; (2) addition of communication capabilities to enable transmission/telemetry of collected and/or processed data; (3) an innovative power control circuit design to enable low-power operation of tandem data collection and computational/communication devices; (4) an entirely new printed circuit board (PCB) design for manufacturing the computational node device; (5) an entirely new software design for the computational component that enables data recording, execution of arbitrary data processing code, and transmission of recorded data and/or processed results; and (6) a new case study using a water end use disaggregation and classification algorithm executed on the node and field test the demonstrates successful deployment to the field. In our case study application, we demonstrate how water end uses can be identified and classified by the node, but we designed the data processing capabilities of the computational node to be generic and support any data processing code that may be needed.

In the following sections, we describe the CIWS computational node, its operating principles, hardware design, software, and user interface (Section 3.2). In Section 3.3, we describe the methods we used for testing our prototypes within a laboratory setting using multiple water meters from different manufacturers. We then describe the results of a field deployment campaign used to assess the capabilities of the computational node under typical field operating conditions (Section 3.4). The final section presents discussion and conclusions. The Data Availability section provides a link where readers can find: (1) hardware designs for the computational node along with instructions for building a prototype device using off-the-shelf components, including
performing all of the hardware modifications; (2) a PCB design with all information
required to manufacture them commercially; (3) firmware code along with more detailed
documentation about the organization and functioning of the firmware; and (4) data and
scripts to reproduce calculations presented in the Case Study Application section of this
paper.

3.2. System Description

The CIWS computational node was designed to collect, process, and transfer high
temporal resolution water use data on existing residential analog water meters and to
meet the following requirements: (1) operation on top of existing, analog meters without
affecting the function of the meter (i.e., recording data for a water utility’s monthly
billing purposes); (2) autonomous operation for at least two weeks without supplemental
power, including data collection, processing, and transfer; (3) simplicity of deployment
and use with an easily operable user interface; (4) generalized support for computations
to be performed (e.g., execution of any data processing code); and (5) output data and
computed results in accessible, platform-independent formats without requiring visits to
deployment sites for manual data downloading. The hardware and software of the CIWS
computational node are open source and can be customized to fit specific research needs,
which means that the CIWS node is an open and customizable platform for collecting and
processing high temporal resolution water use data at the edge.

The CIWS computational node adopts a double processor architecture to achieve
both low power consumption and computational capabilities. The first processor is a low-
power Arduino microcontroller that continuously collects high temporal resolution water
use data and temporarily stores it within an electrically erasable programmable read-only
memory (EEPROM) chip. The second processor is a Raspberry Pi single-board computer that reads the data from the EEPROM chip, writes the data to its own file system, and then executes any code that has been designed to process the raw data (e.g., identifying individual end uses, classifying them, and then transferring raw, event, and/or other summary data to a remote server). Arduino is an open-source hardware and software platform with an AVR single chip microcontroller [25]. Raspberry Pi is a small, single-board computer that uses an open-source distribution of the Linux operating system [21]. As it is always collecting data, the Arduino platform is always powered. Given its higher power consumption, the Raspberry Pi computer is only powered during periods when computations are to take place.

We developed two prototypes of the computational node. We iterated on the first prototype using off-the-shelf components, including an Arduino Pro microcontroller and a Raspberry Pi computer, to perfect our design. The second prototype consists of a PCB that integrates all of the components into a single “hat” that can be interfaced directly with a Raspberry Pi computer’s pin header. The PCB hat includes only those components needed by the computational node to function and was intended to make the computational node easy to manufacture. For example, we included the ATmega328P microcontroller [26] used by the Arduino Pro in the PCB design without the unnecessary peripherals. Given this, we use the term “microcontroller” in the sections that follow to refer to either the Arduino Pro in our off-the-shelf prototype or the ATmega328P in our PCB prototype.
3.2.1. Principle of Functioning

The CIWS computational node uses the same methods for measuring flow through magnetically-driven water meters as the CIWS datalogger. Here, we provide a brief description for completeness, but more specific details are provided by [7]. Both the CIWS datalogger and CIWS computational node were designed to measure and record water flow through magnetically-driven, residential water meters. Many meters of this type use a nutating disc, rotating impeller, or other similar element to measure water flow using the positive displacement principle. When water flows through a meter’s fixed volume measurement element, the nutating disc or rotating impeller is actuated. A nutation of the disc or rotation of the impeller corresponds to a fixed volume of water. The rate of nutation or revolution is proportional to the flow rate, and the count of nutations or revolutions is recorded using a magnetically-driven register. A magnet inside the meter’s register is paired with a spinning magnet attached to the measurement element inside the meter’s sealed housing. These paired magnets rotate together, and the revolutions of the magnet are counted by the meter’s register to determine flow rate and volume. The CIWS computational node uses a magnetometer sensor mounted to the outside of the meter’s housing to measure water flow by counting and recording the number of times the magnet inside the meter rotates.

The computational node differs from the CIWS datalogger in how it records the raw pulse data. Instead of immediately writing pulse count data to an SD card connected to the microcontroller, the microcontroller stores data within an EEPROM chip. As the microcontroller is always powered and collecting data, it is responsible for switching power to the Raspberry Pi computer, which triggers the Raspberry Pi computer to boot,
read data from the EEPROM chip, and then run any computational or data transmission code.

The software of the CIWS computational node is comprised of five main modules: LoggerShell_CLI.py, logger.c, LoggerAutoRun.py, piHandler.py, and arduinoHandler.py (see Section 3.2.4 for more details on the software running on the computational node). LoggerShell_CLI.py is the command line interface for all of the datalogging functionality. logger.c is the module designed for communication with the attached AVR-based datalogger. LoggerAutoRun.py is the autonomous functionality of the node executed every time the Raspberry Pi computer is powered on. piHandler.py is the computational module of the node responsible for handling data processing, data transfer, and data storage functions used by the Raspberry Pi computer. arduinoHandler.py is a wrapper module for logger.c responsible for handling functions used to communicate with the Arduino.

User-created computational code, written in Python, can be imported into the piHandler.py module. All computational code is passed the name of the file that contains the raw data. The computational code reads the data, performs computations, and writes its output to a new file and returns the name of this new file. When all computations are complete, the raw data file and all computation files are, depending on the settings set by the user, stored on the Raspberry Pi computer and sent to a remote server. To send data to a server, the user needs to fill in the upload_url, upload_token_url, and client_passcode strings in the piHandler.py module. When these parameters are filled out, the device is capable of sending data using the Python requests HTTP module.
3.2.2. Hardware

The CIWS computational node’s main components include a microcontroller, an EEPROM flash storage chip, a magnetometer sensor, a Raspberry Pi computer, a bus buffer, a power control circuit, a real-time clock, and a manual activation button (Figure 3.1). In brief, the microcontroller collects pulse data and stores it on the EEPROM chip. The Raspberry Pi computer, when powered by the microcontroller, reads the pulse data from the EEPROM, processes the data, and stores the data and any computational results in files on its Micro SD card. The microcontroller and Raspberry Pi computer communicate with each other using a universal asynchronous receiver-transmitter (UART) serial connection rather than over the serial peripheral interface (SPI) bus, which is only used for reading and writing data to the EEPROM in this design. The CIWS computational node’s main components are described in detail in the following sections.

We implemented our off-the-shelf prototype using an Arduino datalogging shield and a custom shield designed to interface with a Raspberry Pi computer. This dual system enables low power data logging by the microcontroller and periodic, resource-heavy computations on the Raspberry Pi computer. During a majority of the device’s on-time, the Raspberry Pi computer is powered off. The microcontroller is always powered on, but we made several modifications to reduce power consumption. Only timers and peripherals necessary for logging data using the microcontroller are maintained. Modifications included disabling and enabling peripherals such as UART, SPI, and two-wire interface (TWI), as needed. The Raspberry Pi computer is directly connected to the EEPROM chip, but has a buffer in line to ensure there is no interference by the Raspberry Pi computer while the microcontroller is writing data. The Raspberry Pi computer and
microcontroller communicate with each other using UART and two general purpose input/output (GPIO) pins.

3.2.2.1. Data Logging Components

The CIWS computational node’s data logging components include an Arduino microcontroller, an EEPROM flash storage chip, and a magnetometer sensor. The Arduino microcontroller implemented in the design is the ATmega328p 8-bit AVR core microcontroller [26]. The microcontroller is primarily responsible for collecting raw data from the magnetometer, detecting pulses in the data, storing pulse counts to the EEPROM, controlling power to the Raspberry Pi computer, controlling the Raspberry Pi computer’s access to the EEPROM, communicating with the Raspberry Pi computer, and generating timestamps for each recorded data value using information from the real-time clock (RTC). The EEPROM flash storage is a 25LC1024 SPI EEPROM chip [27] that can store 128 kB of data from the microcontroller. The data format stored in the chip is listed in Table 3.1.

The magnetometer used in our prototype is an LIS3MDL magnetometer by ST Microelectronics [28]. The LIS3MDL we use is mounted to a board and sold by Pololu [29]. The LIS3MDL magnetometer has several configurable sample rates. The sample rate used in our design is 560 Hz. When the LIS3MDL signals that the data are ready, the data are read by the microcontroller using the I2C serial bus and are processed and stored on the EEPROM chip. For full details on the data logging components, see [7].

3.2.2.2. Raspberry Pi

The Raspberry Pi computer used in the system is the third-generation Model B version [30]. The Model 3B is based on a 1.2 GHz Broadcom BCM2837, ARM Cortex-
A53 processor. We used the default Raspbian operating system to run the code we
developed on the Raspberry Pi computer. We chose a Raspberry Pi computer for this
application because it provides a fully functional operating system that has all the
features of a computer, including a processor, random access memory (RAM), ability to
run sophisticated computer code, communications capabilities, and a file system for
managing files. The Raspberry Pi computer is responsible for retrieving data recorded by
the microcontroller and then performing any data processing, which includes writing the
data to the Raspberry Pi computer’s file system and any computations required by the
user. The Raspberry Pi computer is also responsible for transmitting data over the
Internet to a remote server. In addition to this technical functionality, the Raspberry Pi
computer implements a user interface for the datalogger (see Section 3.2.4). Via the user
interface, the user settings are communicated to the microcontroller, and information
from the microcontroller is communicated to the Raspberry Pi computer and then to the
user.

3.2.2.3. Bus Buffer

Bus buffers are used to provide a sufficient drive capability to pass signals and
enable communication between several devices over the same bus. Since the
computational node has two master devices (the Raspberry Pi computer and the
microcontroller) trying to communicate with the same servant device (the EEPROM), we
used the bus buffer to make sure data are not corrupted when both the Raspberry Pi
computer and the microcontroller try to communicate with the EEPROM. The bus buffer
was implemented using a 74HC125N buffer chip [31]. We used the bus buffer to connect
the SPI bus on the Raspberry Pi computer, the microcontroller’s SPI bus, and the
EEPROM chip together. The buffer is controlled by the microcontroller and controls data transmission from the EEPROM to the Raspberry Pi computer. When the buffer is activated by the microcontroller, the Raspberry Pi computer is connected to the EEPROM chip. When the buffer is deactivated by the microcontroller, the Raspberry Pi computer is disconnected from the EEPROM chip. This permits the SPI bus to be used while the Raspberry Pi computer is off and is necessary because driving the I/O pins of the Raspberry Pi computer while it is powered off can cause damage to the computer.

3.2.2.4. Power Control Circuit

The power control circuit switches power on and off to the Raspberry Pi computer and is controlled by the microcontroller. Its design required mediating across the different power levels of the power supply (5 V or greater), the Raspberry Pi computer (5 V), and the microcontroller (3.3 V). In the current design, the Raspberry Pi computer is powered on once per day at midnight. When the Raspberry Pi computer is powered on, data are copied from the EEPROM to the Raspberry Pi computer’s memory, which are then processed, and the results are transferred to a remote server. When data transfer is finished, the Raspberry Pi computer turns itself off, and the microcontroller cuts power to the Raspberry Pi computer. The power control circuit we used to enable power switching between the Arduino microcontroller and Raspberry Pi computer is shown in Figure 3.2. This diagram is part of a larger design schematic for the computational node that is available in the project’s GitHub repository (see the Data Availability section).

In Figure 3.2, R10 and R11 are the Arduino Pro’s resistors, Q1 and Q2 are transistors, C15 and C16 are capacitors, U6 is a power regulator, Vin is the input voltage, Vout is the output voltage, and GND is the ground reference point. The power control
circuit can be powered by any battery with a voltage equal or larger than 5 V. The voltage regulator U6 is the component responsible for converting the battery voltage down to 5 V for the Raspberry Pi computer. The two capacitors C15 and C16 smooth out the power supply voltage so that the Raspberry Pi computer does not experience sharp changes in voltage away from normal operating levels at 5 V. The transistor Q1 sits between the battery (>5 V) and U6 and acts as a switch. When it is turned on, U6 will output 5 V. When Q1 is off, U6 will have no input voltage, and will, therefore, give no output voltage. To turn Q1 on, the ‘gate’ pin (labeled ‘1’) must be set to ground. To turn Q1 off, the ‘gate’ pin must be set to the supply voltage. The microcontroller can set the pin to ground with no issues; however, the microcontroller by itself can only set a pin to 3.3 V because its supply voltage is 3.3 V. To remedy this problem, a second transistor (Q2) is used. Q2 is a different kind of transistor and works differently than Q1. To turn on Q2, the electric current must flow into it through pin 2, the ‘base’ pin. To turn off Q2, no current can flow. This process is controlled by the microcontroller. When Q2 is ‘off’, the gate pin on Q1 is connected to the supply voltage, which ensures that Q1 is ‘off’ and cannot conduct. This then turns off the voltage regulator U6, which cuts the power supply to the Raspberry Pi computer. When Q2 is ‘on’, the gate pin of Q1 is no longer connected to the supply voltage. Instead, it is connected to ground. This allows Q1 to conduct and turn on the voltage regulator U6, which supplies power to the Raspberry Pi computer.

3.2.2.5. Real-Time Clock

In order to accurately control the sampling interval and to record timestamps associated with each raw data value, the system uses an RTC. We chose to use a PCF8523 RTC manufactured by NXP Semiconductor [32]. This RTC is already present
on the Arduino datalogging shield and was, thus, a simple prototyping choice for the RTC. The PCF8523 also communicates with the microcontroller using I2C and shares the I2C bus with the LIS3MDL magnetometer. The RTC is used by the microcontroller for sample interval timing and timestamp generation. The RTC signals the microcontroller whenever a specified data recording interval has passed, and the microcontroller then counts up the pulses detected by the magnetometer sensor and stores that value in the EEPROM. The date/time is also read from the RTC when this interval has passed, from which the microcontroller creates a timestamp.

3.2.2.6. Manual Activation Button

The manual activation button is used to start up the Raspberry Pi computer manually. This component is required to enable a user to interact with the Raspberry Pi computer on demand rather than waiting for it to be turned on automatically by the microcontroller. When the manual activation button is pressed, the Raspberry Pi computer powers on and waits for the user to log in via a terminal. The user can then log in to the system to view files, start/stop a logging session, view water flow data, and perform other tasks available through the user interface (see Section 3.2.4).

Table 3.2 lists all components, sources, and approximate costs per unit at the time of this writing to build a CIWS computational node using off-the-shelf components. The cost to build a node is approximately $199.47 with pricing that varies depending on the number of components purchased. Some parts, including cables and connectors, are only available in quantities larger than what is needed for a single node. The costs presented in Table 3.2 were estimated after purchasing the materials needed to build three nodes.
Specific part numbers and a URL link to each vendor are available in the project’s GitHub repository.

3.2.2.7. Printed Circuit Board Design

In an attempt to reduce the time and effort required to manufacture the CIWS computational node using off-the-shelf components, we translated the design for our prototype computational node into a PCB design. The PCB design includes all of the hardware components required to produce a hardware “hat” that interfaces directly with the pin header on a Raspberry Pi 3B computer. We sent the PCB design files to the PCBWay PCB manufacturing company [33] for production and assembly and ordered and tested a small run of five devices to verify their functioning. We successfully tested the PCB devices in a laboratory setting using the testing procedure described in Section 3.3. The total cost for manufacturing and assembling a PCB device (Figure 3.3) was $61.90 USD, which included manufacturing of the PCB and placing of all of the components to create a finished product. The manufacturing cost can be reduced with bulk orders for a larger number of devices. The information required to manufacture the Computational node PCB, including schematics showing connections between all of the parts, Gerber design files with configuration parameters, aperture definitions, coordinate information for the location of parts, and a list of the materials required, is publicly available in the project’s GitHub repository.

3.2.3. Microcontroller Firmware

Similar to the CIWS datalogger, the firmware for the CIWS computational node is organized using the C-like Arduino programming approach and was developed within the Arduino Interactive Development Environment (IDE) [34]. For each of the libraries
that were developed for the CIWS computational node (System State, Store New Record, Real-Time Clock, and System State), the source code and detailed documentation of the functions developed within each library are available in the project’s GitHub repository. In the sections that follow, we describe the libraries developed specifically for the CIWS computational node and their main functions. Along with the newly-developed libraries, additional libraries developed originally for the CIWS datalogger [7] were used, including the detectPeaks, magnetometer, and powerSleep libraries. For completeness, these libraries are included in the GitHub repository for the CIWS computational node, but the reader is directed to [7] for details regarding the functionality each contains.

The main firmware file that operates and controls the CIWS computational node, “Computational_Firmware.ino”, calls all of the libraries mentioned above. It contains four functions: (1) setup(), (2) loop(), (3) INT0_ISR(), and (4) INT1_ISR(). A flow chart describing the firmware is shown in Figure 3.4. As with most microcontroller programs, the setup() function is called once when the device is powered, and the loop() function is called repeatedly until the microcontroller is reset. The functions INT0_ISR() and INT1_ISR() are interrupt service routines that are executed when an event in hardware occurs. These functions manage the retrieval of new data from the magnetometer sensor and the RTC. As the CIWS computational node uses the same sensor measurement and observation timing as the original CIWS datalogger, both functions were adopted as-is from the CIWS datalogger firmware [7]. Table 3.3 lists the main objective of the four functions that comprise the firmware of the CIWS computational node and are included in the Computational_Firmware.ino file.
3.2.3.1. System State Library

The system state library defines two C/C++ structures: State and SignalState. State keeps track of several important values (Table 3.4). When the State structure is initialized, the pulse count is set to zero, the record number is set to one, and the Boolean flags are initialized to false. The SignalState structure keeps track of values used for processing the magnetometer signal, including the input signal from the magnetometer, direct current (DC) removal filter pole, output signal from DC removal filter pole, and software-based Schmitt trigger.

3.2.3.2. Real-Time Clock Library

The RTC library defines a list of hexadecimal addresses and a date/time structure for the RTC’s registers in the microcontroller, which holds the current year, month, day, hour, minute, and second. Table 3.5 lists the functions defined in the RTC library and their main objective.

3.2.3.3. Store New Record Library

The storeNewRecord library defines three functions responsible for storing data in the EEPROM chip, including writeDataSize(), writeDateAndTime(), and storeNewRecord(). The function writeDataSize() is used to write the current number of records to the EEPROM chip. This function is called when the user presses the activation button, or once a day at midnight, so that the number of records on the EEPROM chip is written where the Raspberry Pi computer can find it and so that the Raspberry Pi computer knows exactly how many bytes to read from the EEPROM chip. The function writeDateAndTime() is used to write a timestamp to the EEPROM chip. This function is called when the first data record is written to the EEPROM chip. The timestamp is
written to the address range 0x003–0x008 of the EEPROM chip. This function is called by storeNewRecord(). The function storeNewRecord() is used to store a data record to the EEPROM chip.

The default storage for new data records is the EEPROM chip. However, in some instances, data may be temporarily stored in an array allocated in the microcontroller’s volatile, static random-access memory (SRAM). As long as the EEPROM chip is not being used by the Raspberry Pi computer, all data records bypass the array and are stored directly in the EEPROM. If the EEPROM chip is being used by the Raspberry Pi computer, then the data record is stored in the array. As soon as the Raspberry Pi computer releases the EEPROM back to the microcontroller, the data in the array are appended to the end of existing data records section of the EEPROM chip. Once this process is complete, the array index resets to zero.

Unlike the CIWS datalogger project, an individual data record consists of only one byte, which is the number of pulses detected within the time interval of the record. The Raspberry Pi computer fills in the record number and timestamp when it reads the data from the EEPROM. In addition to storing the current record to the EEPROM chip, the storeNewRecord() function also checks for records in the array waiting to be written to the EEPROM. If there are records and the EEPROM is free, they are written to the EEPROM first, followed by the current record. If the EEPROM is being used by the Raspberry Pi computer, then the current record is instead written to the secondary data buffer. If the Raspberry Pi computer has not freed up the EEPROM when this function is called and the secondary data array is full, the program sets a flag in the system state structure, RPiFalseON, which indicates to the program that the Raspberry Pi computer
has, for some reason, failed to release the EEPROM chip. The Raspberry Pi computer will then be shut down by cutting the power, and data in the secondary buffer will be appended to what is currently in the EEPROM. Unfortunately, this solution creates the potential for data loss if the Raspberry Pi computer freezes before releasing the EEPROM, but is required as a last resort to ensure that the computational node can continue operating in the event that the Raspberry Pi computer encounters a fatal problem. Figure 3.5 illustrates the record storage architecture and the data format for the EEPROM.

The number of records is stored in the first three bytes of the EEPROM (addresses 0, 1, and 2). The byte at address zero is the most significant byte or high byte. The byte at address one is the middle byte, and the byte at address two is the least significant byte. This number corresponds to the number of data records after the timestamp data in the EEPROM. The next six bytes of the EEPROM hold the starting timestamp. This timestamp is the timestamp associated with the first data record in the EEPROM and is the only timestamp stored in the EEPROM. Because the data records are stored based on timing from the RTC, the timestamps for each record can be determined from the original timestamp by the Raspberry Pi computer. The timestamp is composed of six bytes, with one byte per field. The fields are year, month, day, hour, minute, and second. The EEPROM is read and written to using an SPI bus, which is a common peripheral on many microcontrollers, including the ATmega328p we used. A brief description of the transaction with the EEPROM chip is provided here. Further information can be found in the 25LC1024 datasheet (Microchip document DS22064D) [35].
Every time the microcontroller writes to the EEPROM, it must enact two SPI transactions: (1) send a write-enable instruction and (2) send the data to be written. Sending the write-enable instruction consists of sending the single byte 0x06. Writing this byte enables the EEPROM to accept a write instruction. This byte must be sent as a separate SPI transaction before any data writing can be attempted. There is no response from the EEPROM when the write-enable instruction byte is sent. After the write-enable instruction byte has been sent to the EEPROM, the EEPROM is ready to accept the data to be written. The maximum number of data bytes that can be written in a single transaction is 256, as long as all 256 bytes reside on the same page (memory block) in the EEPROM chip.

3.2.3.4. Communication Library

The Communication library defines several functions used for communicating between the EEPROM chip and the Raspberry Pi computer. The functions provide an interface to SPI bus transactions, UART transactions, powering the Raspberry Pi computer on and off, and updating system information based on data in a “report”. Table 3.6 lists the functions defined in the Communication library and their main objective.

3.2.4. Software

The microcontroller powers the Raspberry Pi computer when the user presses a physical activation button, or once a day at midnight. The Raspberry Pi computer must then autonomously read, process, and store the EEPROM data with no intervention from the user. The Raspberry Pi computer must also power itself back off again. All of this is done by the autonomous functionality script of the node (LoggerAutoRun.py). The script was implemented in Python and is run at startup from an rc.local command. rc.local is a
file in the Linux directory whose commands are run at startup. In the node’s design, the LoggerAutoRun.py is executed every time the Raspberry Pi computer is powered on and boots. If the power-on event occurs at the scheduled time at midnight, the script assumes that the power-on was automatic and the Raspberry Pi computer is shut down after it reads data from the EEPROM and performs any computations needed. Otherwise, the script assumes that the user powered on the Raspberry Pi computer via the activation button, and the Raspberry Pi computer is not shut down.

The LoggerAutoRun.py script calls a set of functions in a module called arduinoHandler.py, which was developed to enable interactions between the Raspberry Pi computer and the microcontroller. For example, upon being powered, the Raspberry Pi computer automatically runs the LoggerAutoRun.py script, which calls the setPowerGood() function from the arduinoHandler.py module. The setPowerGood() function sets the Raspberry Pi computer’s general purpose input output (GPIO) 25 pin high, which alerts the microcontroller that the Raspberry Pi computer has successfully powered on. The LoggerAutoRun.py script then calls the writeEEPROMToFile() function from the same arduinoHandler.py module to create a file to hold the data stored in the EEPROM by the microcontroller, calls a SPI transaction to read the data from the EEPROM chip, and copies the data to the data array in the Raspberry Pi computer’s memory.

Copying data from the EEPROM is initiated by calling the reportSwap() function from the arduinoHandler.py module, which sends a START byte to the microcontroller. Sending a new START byte will always initiate a new data copy, even if one is in progress. Data transaction from the EEPROM to the Raspberry Pi computer’s memory is
done once per day to conserve power and create data files containing one day of
uninterrupted data, which is convenient for some data processing steps (e.g., daily
summaries). Data are temporarily saved on the EEPROM until it is full. When the
EEPROM memory is full, newly recorded data points are written and saved over the
oldest data points.

While reading from the EEPROM, the Raspberry Pi computer sets its GPIO 24
pin high using the setRomBusy() function, which alerts the microcontroller that the
Raspberry Pi computer is reading the EEPROM. After copying all data from the
EEPROM to the Raspberry Pi computer, the Raspberry Pi computer calls the
writeEEPROMToFile() function to translate it into a comma separated values (CSV) file
and optionally save it within the file system on the Raspberry Pi computer’s Micro SD
card memory. Both setRomBusy() and writeEEPROMToFile() are called from the
arduinoHandler.py module.

Once the data copy from the EEPROM is complete, the Raspberry Pi computer is
free to run any computational code that the user requires. We describe a case study
application for end use disaggregation and classification in the section that follows, but
we designed the firmware for the node to enable execution of any computational code.
Computational code files are handled using the dataAnalysis() function, which loops
through all computational code files in the piHandler.py module, executes them, and
returns the outputs of each computational code file as CSV files.

For each raw data file and processed data file returned by the
writeEEPROMToFile() and dataAnalysis() functions, respectively, the Raspberry Pi
computer either saves them in the software data directory in the Raspberry Pi computer’s
file system, or transfers them to a remote server based on the data transfer configuration set by the user. If the device configuration is set to save the data on the Raspberry Pi computer, they are saved in the savedData directory of the Raspberry Pi computer. If the device configuration is set to transfer the data to a remote server, the send() function is initiated. The send() function is implemented within a Python module called PiHandler.py.

All data transfers from the node to a remote server are handled by sending files using HTTP POST requests to a data posting service (DPS) hosted on the remote server (for more details of the server software, see [36]). The standard Python requests library was used to implement the creation of HTTP POST requests to transfer the data from the node to the web server. To enable sending data to a server, the user needs to set the upload_url, upload_token_url, and client_passcode strings in the piHandler.py module. The upload_url field contains the database server hostname or IP address. The client_passcode is the password for a user with permission to write data to the remote server. The upload_token_url is an authentication key used to authenticate upload requests. When the Raspberry Pi computer sends an HTTP POST request containing data files (raw data and/or processed data) to the server, the requests are received and handled by the DPS. The DPS authenticates HTTP POST requests using the token supplied by the user (upload_token_url, for more details see [36]).

A flag set by the user in the initial configuration of the device is used to determine which files will be transferred to a remote server. The user can modify the data transfer flag to decide which files to transfer to the remote server through the user interface shell using the set-transmission command. When a flag value of “1” is set, only unprocessed
raw data files are sent to the server. When a flag value of “2” is set, only files resulting from computations are sent to the server. When a flag value of “3” is set, both raw data files and files resulting from computations are sent to the server. Data transfer uses an all-or-nothing protocol; meaning when data transfer is initiated, all data consistent with the flag setting are transferred. Data transfer can be accomplished using the Raspberry Pi computer’s integrated WiFi or via other attached radios (e.g., cellular).

Once all data computations and data transfer are finished, the Raspberry Pi computer sets both GPIO 24 and GPIO 25 pins to low by calling the setPowerOff() and setRomFree() functions from the logger.c module. Setting the GPIO 24 pin low sends a signal to the microcontroller that the Raspberry Pi computer has finished reading the EEPROM. Setting the GPIO 25 pin low sends a signal to the microcontroller to power the Raspberry Pi computer off. The microcontroller will cut power to the Raspberry Pi computer roughly 12 s after the signal is received.

We developed an interactive, command line shell interface (LoggerShell_CLI.py) in Python as a user interface to all of the functionalities of the computational node, including the Raspberry Pi computer and microcontroller. The user interface allows users to execute basic functions needed to configure and operate the computational node, along with managing and retrieving processed and unprocessed data files. This includes configuring the device to work with different water meter brands and sizes. The user interface operates on the Raspberry Pi computer and can be accessed through any serial terminal emulator wirelessly via WiFi, or through a direct Ethernet connection. To access the user interface via the interactive shell on the serial terminal, the power button on the datalogging shield is clicked first, which will power the Raspberry Pi computer and cause
it to boot. Once powered on, a serial connection can be made to the Raspberry Pi computer. By default, the LoggerShell_CLI.py Python script automatically displays the help menu, and the set of commands listed in Figure 3.6 will be accessible.

3.3. Calibration

Laboratory experiments were conducted to ensure that the computational node can accurately measure, process, and transfer water use data at different water flow rate conditions and for different meter types and sizes. Two different water use scenarios were tested in laboratory experiments to examine the data collection accuracy of the device. In both experiments, we pumped water through test meters at multiple flow rates ranging from 0 to 75 LPM for 30 min. In the first experiment, we used a flow controlling valve to pump uninterrupted water flow through the meters while increasing the flowrate every 10 min. This enabled us to test the accuracy of data collection over the range of flowrates expected for residential water meters. The second experiment was similar to the first, but we used the flow controlling valve to interrupt the flow between each flowrate increase. This enabled us to ensure that the computational node accurately measures flow across multiple, discrete events as expected within residential homes. We manually read the manufacturer’s register for each meter before and after each run. We calculated the volume of water used in each run as the difference in manual register readings. We calculated the volume of water logged by the computational node on each meter as the number of pulses recorded by the node multiplied by the pulse resolution of each meter. We then compared the volume registered by the meter’s register with the volume registered by the computational node to ensure that the computational node accurately recorded water flow.
In both experiments, a CIWS computational node was installed on top of a 1-inch Bottom Loading (BL) Master Meter and another on a 5/8-inch Master Meter of the same model. Meter sizes are reported in inches consistent with how these meters are sold in the U.S. The water use volume registered by the meter’s register and the CIWS computational node on the 1-inch meter were 756.8 L and 756.39 L, respectively, for the first experiment and 618.08 L and 625.12 L, respectively, for the second experiment. A maximum percent error of 1.35% was observed across both experiments. For the 5/8-inch meter, the water use volume registered by the meter and the CIWS computational node were 757.35 L and 762.31 L, respectively, for the first experiment and 621.60 L and 633.24 L, respectively, for the second experiment. A maximum percent error of 1.87% was observed across both experiments. Table 3.7 and Table 3.8 show the volumes read by each meter’s register, the volumes captured by the corresponding CIWS computational node, and the percent error for each step for the meters tested in the laboratory.

3.4. Case Study Application

As a case study for demonstrating the node’s data collection and computational functionality, we developed an application for automatically identifying and classifying end use events from raw water trace data recorded by the node on a residential water meter. The node was configured to record raw water use data with a four second recording interval. We then used the CIWS disaggregator algorithm designed by [24] to process the raw water use data to produce classified events. We used the Raspberry Pi computer’s terminal and the package installer for Python (pip) to install all of the dependencies and Python libraries required by the CIWS disaggregator algorithm on the
Raspberry Pi computer. We then placed the CIWS disaggregator algorithm in the software working directory of the Raspberry Pi computer. Prior to deployment in the field, we tested and verified the computational and data transfer capabilities of the node using test data files from a prior study [24]. We manually placed the test dataset in the data working directory of the Raspberry Pi computer, manually executed the CIWS disaggregator algorithm by calling the dataAnalysis() function, and then manually executed the data transfer by calling the send() function.

Once we verified that the node was working correctly in a laboratory setting, we then ran a field deployment of the node. In the field, we installed the node on the water meter for a home in the city of Providence, Utah, USA between 17 January 2021 and 11 February 2021 to evaluate its performance under field conditions. We installed the node with a new, fully charged, 12 V, 10 Ahr battery and ran data collection, event disaggregation, and data transfer to a remote server until the battery failed. This deployment enabled us to test the power consumption of the device and estimate the length of autonomous operation we could expect without an external power supply.

During the field deployment, we configured the Raspberry Pi computer to connect to the homeowner’s WiFi network using its integrated WiFi capability. We also set the device to execute the CIWS disaggregator algorithm, store both raw data and classified event files on the Micro SD card on the Raspberry Pi computer, and transfer both sets of files to a secured server at midnight every day. In the case of a WiFi network failure, we designed the device to store the files within the Raspberry Pi computer’s file system on the local Micro SD card. When the WiFi connection is restored, all files stored on the Raspberry Pi computer’s Micro SD card are transferred to the server.
The remote server to which files were transferred consisted of an instance of the CIWS cyberinfrastructure described by [36], which includes a data posting service that was specifically designed to authenticate and accept data files posted to the server via HTTP POST requests from devices like the computational node. The remote server was implemented on an Ubuntu Linux virtual machine hosted within Utah State University’s Enterprise Data Center. While it is beyond the scope of this paper to describe the details of the larger cyberinfrastructure needed to manage the data created by a network of operational nodes, readers are referred to [36], where we performed scalability testing to investigate the performance of the CIWS cyberinfrastructure and showed that even a modestly provisioned server could robustly handle HTTP POST requests from hundreds of active nodes submitting data at the same time.

3.4.1. Data Output

For our case study deployments, the CIWS computational node output two comma-separated values (CSV) files per day: one for the unprocessed, high-resolution water use data and another for the classified water end use events output by the CIWS disaggregator algorithm. The first three rows in the raw data file are reserved for a standard metadata header that includes a site number at which the node is deployed, a unique identifier for the node, and the meter pulse resolution where the device is installed, i.e., the volume of water corresponding to each magnetic “pulse” recorded by the meter, see [7]. These values were set using the node’s user interface. The fourth row serves as a header for the data and has three fields: Time, Record, and Pulses. The Time field contains the date and time values at which individual observations of water use were recorded. The Record field is a sequential numerical ID used to keep track of the number
of observations logged. The Pulses field is an integer number that corresponds to the number of pulses registered in a time interval.

The classified water end use events file output by the CIWS disaggregator algorithm has eight fields: StartTime, EndTime, Duration, OriginalVolume, OriginalFlowRate, Peak_Value, Mode_Value, and Label. The StartTime field represents the start of an event and contains the date and time at which the water use volume transitions from zero to any positive value. The EndTime field represents the end of an event and contains the date and time at which the water use volume transitions back to zero. The Duration field is the elapsed time of the event in seconds. OriginalVolume is the summation of water use volumes of an event between its start and end times in gallons. OriginalFlowRate is the volume of an event divided by its duration and is recorded in gallons/minute. Peak_Value is the maximum rate of water flow within the event for any time step (gallons/minute). Mode_Value is the rate of water flow that appears most often within an event (gallons/minute). The Label field contains the water end use type of each event in the dataset output by the CIWS disaggregator algorithm. Figure 3.7 shows an example of two CSV files obtained from the CIWS computational node. Only a subset of records is presented.

3.4.2. Battery Life

During the first field deployment, we measured the voltage of the node’s battery before deployment and then monitored the battery’s discharge over the course of the field deployment until it was fully drained and the node failed. The discharge time was 26 days, indicating that the node could reasonably be used to collect, process, and transfer
four-second temporal resolution and classified events data for over three weeks with no external power before the 10 Ahr battery has to be replaced.

Where longer field deployments are needed, a higher capacity battery or a charge regulator connected to a solar panel can be used to enhance the lifespan of the device (Figure 3.8). In addition to the field deployment, we conducted quantitative power testing in the laboratory using a 4.4 Wh battery and a power regulator. We operated the node and monitored the battery voltage on a daily basis using a multimeter. The device’s power consumption was calculated by estimating the device’s full cycle power draw. The Raspberry Pi computer turns on once a day, so its power cycle consists of 24 h. When the Raspberry Pi computer is off, the device uses 0.04 W of power with a 12 V power supply. While the Raspberry Pi computer is running, 1.4 W are consumed at 12 V. The average power used by the device during a single cycle is dependent on how long the Raspberry Pi computer takes to perform its computations. For a device whose computations take 5 min to complete, the average power used by the device is 0.045 W. For a device whose computations take 10 min, it consumes 0.05 W on average.

In further testing to simulate potential field conditions with a small form-factor battery, we used Adafruit’s BQ24074 regulator with a 3.7 V, 4.4 Wh battery and observed 10 days of continual operation before the battery failed. The computational node regulates the input voltage to 5 V and 3.3 V for the Raspberry Pi computer and microcontroller, respectively. To boost the battery’s voltage from 3.7 V to a voltage that could be regulated down to 5 V we used an external Pololu U3V12F9 step-up voltage regulator to boost the voltage up to 9 V. Given the power requirements and considering the losses of the BQ24074 and U3V12F9, 10 days is a reasonable life expectancy for a
3.7 V, 4.4 Wh battery. A solar panel can be used to prolong the life of the device. The BQ24074 already supports connecting a solar panel. However, the practicality of adding a solar panel is highly dependent on the quality and quantity of sunlight available in the region. Put simply, a larger battery lengthens the lifetime of the device while a larger solar panel allows the device to recuperate quicker.

3.4.3. Accuracy

Evaluation of computational node performance in the field was based on multiple accuracy metrics, including volume accuracy, accuracy of event identification and classification, and accuracy of data transfer. To assess volume accuracy, we manually recorded the water use volume on the water meter’s register twice during the field deployment: once during the initial installation of the computational node and a second time two weeks after the installation. We calculated the total water use volume during the two weeks as the difference between the two manual readings. We then aggregated the pulses registered by the computational node for the same period of time and multiplied the total number of pulses by the pulse resolution of the meter to estimate the water use volume registered by the node. The volumes estimated from manual meter readings and from the computational node were 9402 L and 9459 L, respectively, with an error of 0.6%. This error is similar to the levels of error we observed within our laboratory tests. It is also less than the error threshold value of 5% used in our previous study [7], where we presumed that any error value within 5% was acceptable for the purpose of our study.

Evaluating the field accuracy of event detection and classification required collection of an additional dataset. During the field deployment period, we asked the home’s residents to manually label some water use events. For a subset of events in the
home, the residents recorded the water end use type and the event start time. A total of
333 different water end use events were manually labeled during the field deployment as
follows: 127 faucet events, 124 toilet flushes, 38 showers, 19 clothes washer events, 14
dishwasher events, and 11 bathtub events. The accuracy of the classified events was
quantified as the fraction of water use events whose end use category was correctly
predicted by the node when compared to the labels manually assigned by the home’s
residents. The resulting classification accuracy of water end use events ranged from
100% for toilet, faucet, and clothes washer events to 64% for bathtub-filling events,
which is consistent with the performance of the CIWS disaggregator algorithm reported
in our earlier work, where classification was performed on a centralized computer after
manually downloading the raw data from devices in the field [24]. The overall accuracy
of the classification performed by the node was 98.4%. Given these results, we are
confident that the CIWS disaggregator algorithm operated correctly on the computational
node and classified events with the same level of accuracy achieved through manual
downloading and post-processing.

During the period of our field deployment, we experienced no issues with data
transfer from the computational node to a remote server located on Utah State
University’s campus. The accuracy of data transfer was 100% and was quantified as the
percentage of data transfer attempts that were successful. Successful attempts were
verified by logging into the server each day and verifying that both raw data and
classified event data files were uploaded successfully. We also verified that the
transferred files were sent correctly by comparing the files on the server to the original
files stored on the Raspberry Pi computer’s Micro SD card to ensure that they were the same.

3.4.4. Water Use

Using the data collection and computational capabilities of the CIWS computational node, we were able to identify and classify 1480 water use events retrieved from one residential household over the 26 days of our field deployment test, averaging 57 events per day and 14.2 events per capita-day. The average daily indoor water use of the studied household was 625.8 Lpd, and the average per capita indoor water use was 156.5 Lpcd. Compared to the per capita indoor water use for the state of Utah of 227.1 Lpcd estimated by the Utah Division of Water Resources [37], the studied household fell well below the estimate.

For the studied household, showers accounted for the largest volume of water use during the field deployment, followed by toilet flushing, clothes washer events, faucets, and bathtubs. Showers accounted for an average of approximately 34.6% of total indoor water use, toilets accounted for an average of 33.4%, clothes washer events contributed an average of 17%, faucet and dishwasher events contributed an average of 10.6%, and bathtub-filling events contributed an average of 4.3%.

With regard to utilization rate, we used the classified events to calculate the frequency of use for each end use type in the study home. Faucets were the most frequently utilized end use fixture at approximately 31.7 uses per day. Toilet flushing was the second most frequently utilized fixture at approximately 20.2 uses per day. Bathtub filling was the least frequently utilized indoor water end use, accounting for only 0.3 uses per day.
Using the capabilities of the computational node, we were able to produce directly on the node the information needed to perform a detailed water end use analysis. While we chose to transmit both the raw and disaggregated event data for testing purposes during our case study, the disaggregated event data calculated by the computational node is identical to what would be produced if the data were collected on the device, manually downloaded, and post-processed using a centralized computer. Thus, if the final use of the data is to examine classified end use events, the raw data need not be transferred to the server.

3.4.5. Limitations and Errors

An obvious limitation of the CIWS computational node observed during field deployment is related to communication. In Providence, UT, water meters are installed underground to prevent freezing during winter, and the depth of meter pits can exceed 0.5 m. The meter pit depth where the device was installed was approximately 1.5 m. WiFi signals are prone to attenuation by the meter pit casing and the soil surrounding the meter pit. To overcome this issue, and to ensure a reliable WiFi signal strength received by the device, we placed the magnetometer sensor on the meter underground, but placed the computational node in a weatherproof enclosure above ground.

While we used the homeowner’s WiFi network for our field deployment testing, open WiFi networks are not ubiquitous, which may limit their use in larger deployments. Given this, we tested the data transmission capabilities of a Raspberry Pi computer connected to a Hologram Nova cellular data modem [38] and verified that the same data files we produced and transferred over WiFi in the field could be successfully sent to the same server via HTTP POST requests over Hologram’s cellular data network. Repeated
tests in the laboratory showed that transfer of a single day of data (a 550 KB raw data file and an 8 KB disaggregated event data file) required approximately 0.25 s over WiFi versus 25.31 s over cellular. Cellular transmission is slower and would likely consume more power because the Raspberry Pi computer would be powered for longer; however, the cost of transmitting data over cellular is based on data size and not transmission time.

Emplacing the computational node’s weatherproof enclosure above ground made it physically exposed to be opened, compromised, or stolen. While the Raspberry Pi computer has basic endpoint security measures, such as a username and password required for login, it currently lacks data encryption both on the device and during data transfer, which may make it vulnerable to unauthorized access. This is likely not a strong concern because water use data are not highly sensitive. Even so, edge computing reduces the amount of data that are at risk at any one time because each computational node contains data for only a single home. A breach in the network would expose data from one node, whereas a breach in a centralized system may expose much more data.

Another potential limitation of the node is related to data importance. While we used the computational node to transfer both unprocessed and processed data to the server to demonstrate its capabilities, in the quest to minimize data transfer bandwidth and reduce latency, it is more practical to only transfer processed data that contains useful, decision-relevant information for water providers and users. In the event that the raw, unprocessed data are never transported to a centralized system and may never be saved on the computational node, important information that may be present in that data could be overlooked and discarded. There may be useful applications of the raw data (beyond end use disaggregation) that are unrealized because the raw data are not
transmitted and stored long term. This is the specific reason why our design enables local saving and transmission of both raw and processed data. Local saving of raw data can be turned off to save space on the Raspberry Pi computer’s Micro SD card. Transmission of raw data can be turned off to minimize communications bandwidth and required centralized storage. Both can be temporarily turned on to enable monitoring and diagnostics of performance. Localized processing of data on the node may even reduce the amount of data to be transmitted to something that may be feasible over networks with much lower bandwidth than WiFi or cellular (e.g., LoRaWAN) [39], although we did not explore this option.

3.5. Discussion and Conclusions

The work presented here builds upon existing smart water metering and end use disaggregation studies to develop and demonstrate new, open source, and reproducible data collection, end use disaggregation, and classification methods that can be executed on existing water meters using relatively inexpensive hardware. In the context of our case study application, the record of classified events produced by the computational node is the same as what would be produced by logging data in the field followed by manual downloading and centralized post-processing of the high-resolution data. The type of data products that can be produced by the computational node have already been shown to support a wide range of analysis and modeling applications typically undertaken by interdisciplinary research and integrated water management teams.

The hardware designs and the firmware code used to prototype the computational node are available and open source. The device we designed can either be built using off-the-shelf components or it can be manufactured by a PCB manufacturer, providing
flexibility for potential users who may not have electronics prototyping expertise. The computational node was successfully deployed and tested in a laboratory setting under optimal conditions (e.g., constant temperature with a dedicated power supply) and in the field under variable temperature and power configurations to demonstrate successful sensing, data logging, and computational capabilities on existing analog water meters. This means that for approximately $199.47, the computational node can be viably used with existing, magnetically-driven residential water meters to: (1) collect data at a very high temporal frequency and up to the pulse volume resolution of the meter, (2) extract and classify water end use events or other computational tasks directly on the node, and (3) transfer the unprocessed raw data and/or the classified water end use events to a centralized server without affecting the performance of the existing meter.

We anticipate that the device that we have prototyped could be used by researchers in data collection and processing to address questions about residential water use and user behavior, by water managers to collect and analyze data from residential settings for operational use, by homeowners to monitor their water use, and by water meter manufacturers to upgrade the designs of existing smart water meters. We believe that the hardware design of the computational node is generalizable across these potential users and their use cases, but this may require additional software to present the event data produced in a context understandable by the user. For example, researchers and water managers could likely parse and analyze event data using general purpose data visualization and analysis software, but homeowners may need additional levels of data aggregation or summary and a custom user interface (e.g., a smartphone app) designed to present the event data in an easily interpretable way.
While our case study application focused on end use disaggregation and classification, the computational capabilities of the node are generic. Similarly, the potential uses of the data are not constrained and could certainly include optimization programs. Any analysis or computational code required by the user can be executed by the Raspberry Pi computer. Processing data at the edge of the network close to where data are generated instead of centrally enables delivery of intelligent, near real-time responsiveness, while drastically reducing the amount of data that must be transferred. Because computational tasks such as end-use disaggregation are performed directly at the meter site, the results can be transferred to other devices with less (or almost no) computing power. This minimizes the required visits to sites for retrieving data and reduces the amount of processing power required to provide local (e.g., an in-home display or smartphone app) or centralized (e.g., aggregated data for a water utility) data services because the useful information has already been computed by the time it gets to the system on which it is used. Because computing is done before results are sent, water utilities do not have to wait (or pay) for data to transfer over a network or for it to be computed centrally and sent back, thus minimizing potential network latency. It also minimizes the need for central storage and management of large volumes of raw data. Although many services related to water use data do not need to be done in real-time, applications such as leak detection could be delivered in near real time to water providers and consumers. The current tradeoff for these capabilities is that the device’s owner must install and maintain the device, including its battery.

The results of computations (e.g., daily water use summaries and disaggregated end uses) are much smaller than the raw, high-resolution data and can be much more
easily communicated over a network with a much lower bandwidth. This would be considerably cheaper on a cellular data network and more scalable on radio networks such as LoRaWAN, which can be relatively inexpensive but has lower available bandwidth for data transfer. In our case study application, one data collection site collected approximately 550 KB of data per day of raw data at the four second recording interval. This would be about 201 MB per year per site. In a small city with 5000 residential connections, this would add up to about 1 TB of data sent over a telemetry network per year if every meter were equipped with high-resolution data collection. The disaggregated event data files averaged less than 8 KB per day for the same site, which means that savings of multiple orders of magnitude in data size could be realized if only processed events are transmitted.

The computational node we designed for collecting, processing, and transferring high temporal resolution data advances available smart water metering and supporting cyberinfrastructure for building the scientific data and knowledge base for sustainably managing urban water supplies. We anticipate that our design and the concepts that we have demonstrated will be useful in building and managing next-generation smart metering systems and their resultant data.

The classified water end use events can equip water utilities and water users with a detailed information on the variation in water use for each end use type, including the total number of events, number of events per day, number of events per capita per day, event volume, event duration, and event flowrate for each of the end uses.
Author Contributions

All authors contributed to the conceptualization of the work presented, to the selection of the methodology used, and in testing and evaluating prototypes of the computational node presented. N.A.A. wrote the initial draft of the paper. J.S.H., A.S.B.J. and R.J.T. contributed to review and editing. R.J.T. and A.S.B.J. led hardware prototyping and firmware development with contributions from N.A.A. and J.S.H. N.A.A. collected the field data and curated it in HydroShare. J.S.H. provided project supervision and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the United States National Science Foundation under grant number 1552444. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Institutional Review Board Statement

The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Institutional Review Board of Utah State University (protocol code 9595 approved on 24 June 2020).

Informed Consent Statement

Informed consent was obtained from all subjects involved in the study.
Data Availability Statement

All of the hardware modifications, parts, PCB design, firmware code, and supplemental materials required for producing the computational node described in this paper are available in the GitHub repository for the project at https://github.com/UCHIC/CIWS-WM-Node (accessed on 15 June 2021). The repository contains separate folders for Hardware and Firmware. All of the firmware libraries (.h and .cpp files) and supplemental firmware documentation are available in the Firmware folder. The Hardware folder contains additional images of the logger, hardware design, layout, PCB design, and instructions to perform the hardware modifications described in this article. All of the data collected during the field deployment associated with the results reported in Section 3.4.3 and Section 3.4.4 are open source and publicly available in the HydroShare repository [40].

Acknowledgments

The authors would like to acknowledge the support of the Utah Water Research Laboratory at the Utah State University for facilitating the laboratory work and providing technical support. We would like to acknowledge Providence City for their cooperation and support in the realization of the field campaign. The authors would also like to acknowledge support from Camilo J. Bastidas Pacheco in the field deployment campaign. We also want to acknowledge and thank the owner of the residential home that participated in the data collection campaign.

Conflicts of Interest

The authors declare no conflict of interest.
REFERENCES


### Tables

**Table 3.1. EEPROM data format.**

<table>
<thead>
<tr>
<th>Number of bytes</th>
<th>Record type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-2</td>
<td>Number of data records</td>
</tr>
<tr>
<td>3-8</td>
<td>Starting timestamp</td>
</tr>
<tr>
<td>9-N</td>
<td>Data bytes</td>
</tr>
</tbody>
</table>

**Table 3.1. Parts required and cost to build a CIWS computational node using off-the-shelf components.**

<table>
<thead>
<tr>
<th>Part</th>
<th>Cost ($)</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>12V 10Ah Duracell Battery</td>
<td>39.99</td>
<td>Batteries + Bulbs</td>
</tr>
<tr>
<td>Raspberry Pi 3B</td>
<td>35.00</td>
<td>Adafruit</td>
</tr>
<tr>
<td>Pelican 1150 Waterproof Case (with foam)</td>
<td>31.96</td>
<td>Amazon</td>
</tr>
<tr>
<td>3.3V, 8MHz Arduino Pro (ATmega328p Board)</td>
<td>15.95</td>
<td>Sparkfun</td>
</tr>
<tr>
<td>Datalogging Shield</td>
<td>15.95</td>
<td>Adafruit</td>
</tr>
<tr>
<td>Micro SD Card with Adapter</td>
<td>9.95</td>
<td>Mouser</td>
</tr>
<tr>
<td>5-Conductor Cable</td>
<td>9.11</td>
<td>Mouser</td>
</tr>
<tr>
<td>TSR_1-2450 Converter</td>
<td>5.48</td>
<td>Digikey</td>
</tr>
<tr>
<td>LIS3MDL Magnetometer + Breakout Board</td>
<td>4.95</td>
<td>Pololu</td>
</tr>
<tr>
<td>1725656 Terminal Block</td>
<td>1.66</td>
<td>Digikey</td>
</tr>
<tr>
<td>1725685 Terminal Block</td>
<td>4.10</td>
<td>Digikey</td>
</tr>
<tr>
<td>25LC1024-E/SM Connectors</td>
<td>3.09</td>
<td>Digikey</td>
</tr>
<tr>
<td>Anderson Powerpole Connectors</td>
<td>2.60</td>
<td>Amazon</td>
</tr>
<tr>
<td>Battery Lead Connectors</td>
<td>2.75</td>
<td>Grainger</td>
</tr>
<tr>
<td>Box Kit</td>
<td>2.10</td>
<td>Mouser</td>
</tr>
<tr>
<td>1920-1076-ND Cable Glands</td>
<td>1.75</td>
<td>Digikey</td>
</tr>
<tr>
<td>FQP27P06 MOSFET</td>
<td>1.78</td>
<td>Digikey</td>
</tr>
<tr>
<td>Strap Set: Gear Strapz (+5 Clasps)</td>
<td>1.52</td>
<td>Amazon</td>
</tr>
<tr>
<td>100NF 50V 0805 Capacitor</td>
<td>1.50</td>
<td>Mouser</td>
</tr>
<tr>
<td>1uF Ceramic Capacitors</td>
<td>0.92</td>
<td>Mouser</td>
</tr>
<tr>
<td>Stripboard</td>
<td>1.43</td>
<td>Mouser</td>
</tr>
<tr>
<td>10k Ohm Resistor</td>
<td>0.20</td>
<td>Digikey</td>
</tr>
<tr>
<td>4.7k Ohm Resistor</td>
<td>0.10</td>
<td>Digikey</td>
</tr>
<tr>
<td>In-Line Fues Holder</td>
<td>4.21</td>
<td>Mouser</td>
</tr>
<tr>
<td>2N3904 Transistor</td>
<td>0.44</td>
<td>Mouser</td>
</tr>
<tr>
<td>Spacers</td>
<td>0.20</td>
<td>Mouser</td>
</tr>
<tr>
<td>Fuse</td>
<td>0.17</td>
<td>Mouser</td>
</tr>
<tr>
<td>Button</td>
<td>0.16</td>
<td>Mouser</td>
</tr>
<tr>
<td>Screws</td>
<td>0.17</td>
<td>Mouser</td>
</tr>
<tr>
<td>Nuts</td>
<td>0.12</td>
<td>Mouser</td>
</tr>
<tr>
<td>Serial Extender Housing Pack</td>
<td>0.16</td>
<td>Pololu</td>
</tr>
<tr>
<td><strong>Total Cost</strong></td>
<td><strong>199.47</strong></td>
<td></td>
</tr>
</tbody>
</table>
Table 3.3. Functions executed by the CIWS computational node Computational_Firmware.ino file and the main objective.

<table>
<thead>
<tr>
<th>Function</th>
<th>Main objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>setup()</td>
<td>Executes the following tasks:</td>
</tr>
<tr>
<td></td>
<td>- Initializes the system state data structure.</td>
</tr>
<tr>
<td></td>
<td>- Initializes GPIO pins.</td>
</tr>
<tr>
<td></td>
<td>- Initializes the magnetometer sensor.</td>
</tr>
<tr>
<td></td>
<td>- Initializes the real-time clock.</td>
</tr>
<tr>
<td></td>
<td>- Initializes the AVR SPI module.</td>
</tr>
<tr>
<td></td>
<td>- Sets up the magnetometer and real-time clock interrupt handlers.</td>
</tr>
<tr>
<td></td>
<td>- Initializes Raspberry Pi report data.</td>
</tr>
<tr>
<td></td>
<td>- Stops using the clock for all unused peripherals to reduce power consumption.</td>
</tr>
<tr>
<td>loop()</td>
<td>The datalogger firmware’s main loop function that performs the following actions:</td>
</tr>
<tr>
<td></td>
<td>- Check if the Raspberry Pi activation button is pressed.</td>
</tr>
<tr>
<td></td>
<td>- Copy report data with the Raspberry Pi.</td>
</tr>
<tr>
<td></td>
<td>- Negotiate the SPI bus with the Raspberry Pi.</td>
</tr>
<tr>
<td></td>
<td>- Check if a data recording interval has elapsed.</td>
</tr>
<tr>
<td></td>
<td>- Update timestamp.</td>
</tr>
<tr>
<td></td>
<td>- Check if magnetometer data is ready.</td>
</tr>
<tr>
<td></td>
<td>- Process incoming data to count peaks.</td>
</tr>
<tr>
<td>INT0_ISR()</td>
<td>Checks if there is any new data ready to report.</td>
</tr>
<tr>
<td>INT1_ISR()</td>
<td>Checks if the data recording interval has elapsed.</td>
</tr>
</tbody>
</table>

Table 3.4. System State library functions.

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte</td>
<td>pulseCount()</td>
<td>The number of pulses in the current sample period</td>
</tr>
<tr>
<td>Byte</td>
<td>lastCount()</td>
<td>The number of pulses in the previous sample period</td>
</tr>
<tr>
<td>Byte</td>
<td>interval()</td>
<td>The time interval between data records</td>
</tr>
<tr>
<td>Integer</td>
<td>totalCount()</td>
<td>The number of pulses since logging started</td>
</tr>
<tr>
<td>Long</td>
<td>recordNum()</td>
<td>The record number of the current sample period</td>
</tr>
<tr>
<td>Long</td>
<td>romAddr()</td>
<td>Pointer to the current address in EEPROM</td>
</tr>
<tr>
<td>Bool flag</td>
<td>logging()</td>
<td>True if the device is logging, false if it is not</td>
</tr>
<tr>
<td>Bool flag</td>
<td>flag4()</td>
<td>True if a data recording interval has passed, false if not</td>
</tr>
<tr>
<td>Bool flag</td>
<td>readMag()</td>
<td>True if magnetometer data is ready, false if it is not</td>
</tr>
<tr>
<td>Bool flag</td>
<td>newReport()</td>
<td>True if a transaction with the Raspberry Pi is complete, false if it is not</td>
</tr>
<tr>
<td>Bool flag</td>
<td>RPiON()</td>
<td>True if power is supplied to Raspberry Pi, false if it is not</td>
</tr>
<tr>
<td>Bool flag</td>
<td>powerGood()</td>
<td>True if the Raspberry Pi signals after power-on, false if not</td>
</tr>
<tr>
<td>Bool flag</td>
<td>romFree()</td>
<td>True if the Raspberry Pi signals it is finished with EEPROM, false if not</td>
</tr>
</tbody>
</table>
Bool flag  RPiFalseON()  True if the Raspberry Pi is unresponsive on power-on, false if it is not

Table 3.5. RTC library functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtcTransfer()</td>
<td>Responsible for transferring raw data collected on the microcontroller to the RTC and takes an eight-bit register number, a read/write flag, and an eight-bit value to write. This function utilizes the Arduino IDE’s Wire library for I2C communication with the RTC.</td>
</tr>
<tr>
<td>loadDateTime()</td>
<td>Reads all of the RTC’s date and time registers, and stores the resulting data in a date/time structure. This function is called each time a data recording interval has passed.</td>
</tr>
<tr>
<td>copyDateTime()</td>
<td>Reads data in a date/time structure and stores the data in a second date/time structure. This function is called when the Raspberry Pi is activated. The copied timestamp is the start time for the next batch of data in the EEPROM.</td>
</tr>
<tr>
<td>setClockPeriod()</td>
<td>Adjusts the RTC interrupt clock period, thus adjusting the time interval between data records.</td>
</tr>
</tbody>
</table>

Table 3.6. Communication library functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>updateReport()</td>
<td>Used to update system information and configuration based on data in a report from the Raspberry Pi. These reports are passed between the Raspberry Pi and the microcontroller one byte at a time.</td>
</tr>
<tr>
<td>powerRPiON()</td>
<td>Used to power on the Raspberry Pi by setting pin PC2 (microcontroller analog pin 2) high. This action triggers the power switching circuit, which connects the battery to a 5-volt regulator, which then powers the Raspberry Pi.</td>
</tr>
<tr>
<td>powerRPiOFF()</td>
<td>Used to power off the Raspberry Pi by setting pin PC2 low. This action turns off the power switching circuit, essentially disconnecting the Raspberry Pi’s regulator from the battery.</td>
</tr>
<tr>
<td>UART_Init()</td>
<td>Initializes the microcontroller’s UART module at a baud rate of 9600 bps (bits per second). The UART is only used to communicate with the Raspberry Pi.</td>
</tr>
<tr>
<td>UART_Transmit()</td>
<td>Takes an input byte and writes it to the UART data register, UDR0. The microcontroller automatically takes the data in UDR0 and transmits it on the UART Tx pin.</td>
</tr>
<tr>
<td>UART_Receive()</td>
<td>Reads the UART data register. Reading the data register loads the byte received on the UART Rx pin.</td>
</tr>
<tr>
<td>UART_End()</td>
<td>Disables the UART module.</td>
</tr>
<tr>
<td>spiInit()</td>
<td>Initializes the microcontroller’s SPI module, which is used for writing data to the EEPROM chip. The module is reactivated whenever a transaction is about to take place.</td>
</tr>
<tr>
<td>spiOff()</td>
<td>Deactivates the microcontroller’s SPI module.</td>
</tr>
<tr>
<td>spiSelectSlave()</td>
<td>Sets the SPI chip select pin low, signaling to the EEPROM that an SPI transaction is about to take place. This function also activates the SPI module.</td>
</tr>
<tr>
<td>spiReleaseSlave()</td>
<td>Sets the SPI chip select pin high, signaling to the EEPROM that the transaction has been completed. This function also deactivates the SPI module.</td>
</tr>
<tr>
<td>spiTranceive()</td>
<td>Iterates over the array of input bytes and writes each byte to the SPI data register (SPDR). The function waits while each byte is transmitted, then reads the SPDR.</td>
</tr>
</tbody>
</table>
Table 3.7. Results from laboratory experiment 1 for the 1-inch and 5/8-inch Master Meters.

<table>
<thead>
<tr>
<th>Time</th>
<th>Flowrate (LPM)</th>
<th>1inch Meter Water Use Volumes (L)</th>
<th>5/8inch Meter Water Use Volumes (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Meter</td>
<td>Computational Node</td>
</tr>
<tr>
<td>9:00</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9:10</td>
<td>17.6</td>
<td>191.36</td>
<td>191.36</td>
</tr>
<tr>
<td>9:20</td>
<td>30.8</td>
<td>289.12</td>
<td>289.44</td>
</tr>
<tr>
<td>9:30</td>
<td>54.4</td>
<td>518.4</td>
<td>519.04</td>
</tr>
<tr>
<td>9:40</td>
<td>71.2</td>
<td>756.8</td>
<td>759.39</td>
</tr>
</tbody>
</table>

Table 3.8. Results from laboratory experiment 2 for the 1-inch and 5/8-inch Master Meters.

<table>
<thead>
<tr>
<th>Time</th>
<th>Flowrate (LPM)</th>
<th>1inch Meter Water Use Volumes (L)</th>
<th>5/8inch Meter Water Use Volumes (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Meter</td>
<td>Computational Node</td>
</tr>
<tr>
<td>9:50</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>10:00</td>
<td>13.80</td>
<td>147.20</td>
<td>147.20</td>
</tr>
<tr>
<td>10:10</td>
<td>35.2</td>
<td>351.20</td>
<td>351.52</td>
</tr>
<tr>
<td>10:20</td>
<td>55.2</td>
<td>472.64</td>
<td>473.28</td>
</tr>
<tr>
<td>10:30</td>
<td>64.8</td>
<td>618.08</td>
<td>625.12</td>
</tr>
</tbody>
</table>

Figures

Figure 3.1. The CIWS computational node hardware architecture, including a Raspberry Pi computer, an Arduino datalogging shield, and a custom-designed Pi hat.
Figure 3.2. Power control circuit of the CIWS computational node.

Figure 3.3. Printed circuit board implementation of the CIWS computational node. This board is designed to mount directly to the pin header of the Raspberry Pi computer (shown underneath the PCB) and includes all of the hardware components on a single board.
Figure 3.4. CIWS firmware architecture.
Figure 3.5. CIWS computational node record storage architecture.

```
……………………………… LoggerShell …………………………………
|
| Welcome to the LoggerShell Command-Line Interface |
| Type help for a list of commands |
|…………………………………………………………………………………………|
 |
> help
> LoggerShell is a shell interface for all of the
> functionality between the Raspberry Pi and the
> datalogging microcontroller. The following is a list
> of commands:
> |
> > date-time: // Displays the current date/time
> > set-date-time // Updates the current date/time
> > set-timer-resolution // Sets the time interval between samples
> > get-timer-resolution // Displays the time interval between samples
> > exit // Exit from LoggerShell
> > exit-poweroff // Exit from LoggerShell and power off (Microcontroller will continue to log data)
> > start-logging // Begin logging data
> > stop-logging // Stop logging data
> > set-id // Set a new datalogger ID number
> > get-id // Read and display the datalogger ID number
> > set-site // Set a new datalogger site number
> > get-site // Read and display the datalogger site number
> > set-meter-resolution // Set a meter resolution for the datalogger
> > get-meter-resolution // Read and display the meter resolution for the datalogger
> > set-transmission // Change data transmission setting
> > get-transmission // Read and display transmission setting
> > set-storage // Change data storage setting
> > get-storage // Read and display storage setting
> > get-battery-voltage // Display battery voltage
> > internet-status // Checks if node is connected to the internet
> > water-flow // Display water flow data for the previous sample
> >
```

Figure 3.6. CIWS computational node user interface help menu showing available functions.
Figure 3.7. Sample output from the CIWS computational node. Panel A shows the raw data. Panel B shows the extracted and classified events from the raw data.

Figure 3.8. Solar-powered CIWS computational node prototype.
CHAPTER 4

SIMULATING COMMUNITY WATER USE BEHAVIOR AND POTENTIAL

WATER CONSERVATION USING DETAILED END USE EVENT DATA

Abstract

We present a model of indoor residential water use that estimates water demand and conservation potential by end use for a target community by simulating indoor water end use events at a household level. The model uses end use event data from a set of representative residential households to simulate a larger community and advances existing end use models by: 1) accounting for an expanded set of indoor water end uses; 2) considering the variability in flowrates, durations, and volumes for end use events over different days of the week; and 3) providing a generalized approach for simulating indoor water usage and potential conservation at the city level. The model simulates residential water use behavior in individual households by randomly sampling water end use events for different end use types for each day of the week and then aggregating the sampled water end use events per day to estimate the daily water use per household. We used the model to evaluate a set of technological and behavioral conservation actions to quantify the conservation potential in each simulated household as well as aggregated to the city level. We evaluated the performance of the model in predicting the observed average daily water use of households in Logan City and compared against other common water demand models to demonstrate the reliability of the developed model. The results of this paper are reproducible using openly available code and data, representing an accessible platform for advancing water demand modeling using detailed water end use data.
4.1. Introduction

With rapid growth of urban populations and limited resources, improving the short and long term planning and management of urban water supply has created a persistent need to develop and adopt alternative management schemes (Gaudin 2006). In the last decade, several water demand forecasting and simulation strategies have been proposed to promote water conservation and water demand management (Koutiva and Makropoulos 2019). Residential water demand modeling aims to simulate the water demand behavior of households and how it is influenced by management strategies and external factors (e.g., environmental, social, etc.). Since the 1960s, many residential water demand modeling-oriented studies have been published, where monthly water use data have been frequently used for management programs. For example, in 2008, Aurora Water tracked and analyzed residential monthly water use records for the city of Aurora, Colorado, USA for a period of one year and investigated the impacts of different demand management programs enacted for different months (e.g., price, water restrictions, and rebate programs) (Kenney et al. 2008). Despite their dissimilar contexts and techniques, residential water demand modeling studies have mostly shared the same procedure in simulating water demand that first determines a set of independent variables to be used in the model for predicting water use (e.g., number of residents, age of the property, etc.) and second determines an estimation method or model formulation (Worthington and Hoffman 2008).

The major determinants of water use included in most existing demand modeling studies have been the number of residents in a household, the existence of swimming pools, precipitation rates, price of water, and the outdoor lot size (Wentz et al. 2013;
Kenney et al. 2008; Haley et al. 2007; Arbués et al. 2003; Dalhuisen et al. 2003; Gaudin 2006; Mayer et al. 1999; Espey et al. 1997). Regression models have been the prime estimation method adopted in several studies to simulate and predict residential water use, including ordinary least squares regression (Agthe and Billings 1980; Carver and Boland 1980; Schefter and David 2006), two-stage least squares (Chicoine et al. 1986; Renwick et al. 2019; Stevens et al. 1992), three-stage least squares (Chicoine et al. 1986), instrumental variable approach (Higgs and Worthington 2001; Martinez-Espiñeira 2002; Renwick et al. 2019), maximum likelihood approach (Hajispyrou et al. 2002), generalized least squares approach (Gaudin et al. 2006; Höglund 1999), and generalized method of moments approach (Garcia and Reynaud 2004; Nauges and Thomas 2003).

With respect to spatial scale, residential water demand models have been developed at district levels (Mamade et al. 2014), household levels (Kontokosta and Jain 2015), and water end use levels (Cahill et al. 2013). At district levels, water demand models have used a spatial scale consisting of a group of residential households in one or more cities. Such a spatial scale is typically relevant for infrastructure planning and long term water demand forecasting (di Mauro et al. 2020). At the household level, water demand models have been primarily used to estimate peak water demand and timing with output estimates for a single household (di Mauro et al. 2020). At the end use scale, water demand models have been used to better understand residential water use behavior, the consumption rate of each water end use inside household units, and to develop targeted water end use conservation actions. Given the variability of models at different spatial and temporal scales, the required input data and model output also varies. The temporal scale of district-level water demand modeling varies from hourly to monthly and annual
intervals (di Mauro et al. 2020), whereas the temporal scale of household and end use models can vary from minutes to one day. The majority of household scale models use data inputs collected with a time resolution of 15 minutes to one day. End use scale models use data inputs gathered at seconds to one minute resolutions.

Achieving an appropriate balance between water supply capacity expansion and water conservation requires more mechanistic and detailed modeling approaches that allow water managers to control for demographic, behavioral, and social variation in water use across households (Jorgensen et al. 2009). This can be vital for utilities where water is scarce and developing more water supply is expensive or even impossible. In addition, given new standards and technologies in water end uses and demographic and behavioral heterogeneities of water consumers, growth in water demand is unlikely to be homogenous. Thus, detailed water modeling and targeted conservation actions may be necessary planning tools for water supplying agencies.

Over the last two decades, models have started to include behavioral factors (e.g., shower duration) (Matos et al. 2013; Romano and Kapelan 2014; Talebpour et al. 2014) and geospatial factors (e.g., climate) (Maeda et al. 2011; Praskievicz and Chang 2009; Kuski et al. 2020). The emergence of smart metering technology and the high temporal and spatial resolutions of recent water end use monitoring studies has enhanced the development of residential water use models that account for economic, behavioral, and geospatial factors (Cominola et al. 2015; Makki et al. 2015). Some of these more advanced models integrate end use data to simulate the water demands of individual water end uses such as faucets, showers, toilets, etc. and then aggregate end uses to estimate consumption at the household level (Cominola et al. 2018). Coupling such an event level water demand model with
demographic surveys about households and their residents including number of residents, age distribution, age of household, and characteristics of water-using fixtures can lead to more realistic simulations of water demand patterns that compare well to those that have been observed.

For example, Blokker et al. (2009) developed a water demand model to predict water use from end use measurements. Statistical data from a survey conducted across 46 households in the city of Amsterdam, Netherlands, including census data and the average age in each household, were incorporated into the model along with water use data obtained from different end uses. The frequency of water use for each event type was simulated using a Poisson distribution, water use volume for each individual event of different end use types was assumed to be constant, and the flowrate of water use for each event was simulated as a lognormal distribution. Williamson et al. (2011) modified the model developed by Blokker et al. (2009) to develop an enhanced water end use model. Modifications included changing fixed volumes of water use for different end uses to probability distributions, which allowed them to account for the water use variability present in each end use type. However, Williamson et al. (2011) used water end use data collected from only 20 residential households in South East Queensland, Australia and generated probability distribution curves for sampling using end use data and statistical data from a survey conducted across those households. End use probability curves were used to sample water consumption, while statistical probability curves were used to sample demographic variation of residents, including the number of residents of a simulated household and technical performance of its water end uses.
Table 4.1 lists characteristics of several approaches for modeling residential end uses, including: indoor end uses incorporated, whether the model can simulate conservation potential, software used, whether the model uses an open source software license, whether the model accounts for daily variation in water use, and whether the model is generalizable to other communities. Despite the recent improvements in residential water use modeling established by these models, some important variables have been left out or not adequately integrated into the models. This includes not accounting for all different types of water end uses, assuming constant flow rate and/or constant water use volume for all end uses of the same type, and not having a realistic probability for occurrence of water use events over different days of the week. In 2011, a team of researchers conducted a review study of the existing residential urban water end use models and concluded that the ability of existing models to simulate water end use demands especially at a city scale is limited (House-Peters and Chang 2011).

In this paper, we present an end use water demand model that addresses these gaps in prior modeling efforts reported in Table 4.1 and that is aimed at improving understanding of residential water use behavior and promoting water management and conservation strategies for water utilities. The model described in this paper simulates a more complete set of indoor water end uses than other models and uses realistic probability of occurrence for all events and their associated features (frequency, volume, duration, and flowrate) instead of assuming average values for these features. The model accounts for heterogeneity of water use behavior amongst different residential households by using an event dataset drawn from a representative set of households. The model is also open source for further testing and reuse.
The simulation process results in estimates of water use for a group of households that reflect realistic variability in water end use technologies and residents with diverse water use behavior. We utilized this detailed technical and behavioral information to investigate a set of water conservation actions and quantified their associated water saving potential. Technological practices included those designed to reduce water irrespective of the residents’ behavior (e.g., retrofitting an inefficient showerhead). Behavioral practices focused on changing residents’ habits irrespective of the technology being used (e.g., fewer showers). Water use savings associated with these actions was calculated as the difference in water use before and after conservation actions were implemented. This study was focused on answering the following research questions: a) How can improving the representation of water end uses at a detailed level within a water demand model improve our ability to predict residential water use and the effects of conservation actions? b) What is the water saving potential for individual homes as well as aggregated to a city level associated with different technological and behavioral conservation actions designed to reduce current indoor water use?

The case study presented demonstrates how detailed water end use records from an existing study can be used to simulate the water use behavior of residential households for which there is no detailed water end use data available. In the case study application, we used the simulation results to analyze the variability of water use in terms of timing and distribution of end uses, efficiency of end uses, and water conservation potential of residential households in the city of Logan, Utah, USA. We demonstrate how the model is generally applicable and can be modified to simulate the detailed water end uses of other cities. Applying the model requires availability of monthly water use records for the
simulated households and the existence of a sample of households from a detailed water end use dataset such that the water use behavior of the sample households is representative of the water use behavior of the simulated households.

4.2. Methods and Materials

4.2.1. Water End Uses

We identified seven indoor water end uses to be incorporated in the water end use demand model, including faucet, toilet, shower, bathtub, clothes washer, dishwasher, and unclassified (events not associated to any end use type, i.e., leaks). In the U.S., these are the main water end uses expected in single-family residential households. To evaluate the efficiency of these end uses, we used specifications from the current federal standard defined by the U.S. Energy Policy Act of 1992 (DOE 1992), the Environmental Protection Agency’s (EPA) Energy Star Program (EPA 2021a), and the U.S. EPA WaterSense efficient fixtures (EPA 2021b). The Energy Policy Act of 1992, which became a law in 1994, mandates a maximum water use volume or flowrate for different end use fixtures manufactured and installed in the U.S. after 1994 and was designed to encourage manufacturing of high performing, water efficient fixtures. Based on these specifications, we divided faucet, toilet, and shower events into three categories: inefficient events, typical events, and efficient events. Inefficient events are those that have water use volumes or flowrates higher than the maximum water use volume or flowrate mandated by the U.S. Energy Policy Act of 1992. Typical events are those that have water use volumes or flowrates less than the maximum mandated standard by the U.S. Energy Policy Act of 1992 and higher than the EPA WaterSense program specifications. Efficient events are those that have water use volumes or flowrates less
than or equal to the EPA WaterSense program specifications. For clothes washer and dishwasher events, we used the specifications defined by the EPA EnergyStar Program to classify events as efficient, typical, or inefficient (Table 4.2). We assessed the efficiency of bathtub filling events using the size of a bathtub. Standard bathtubs can hold up to 300 liters of water. Smaller bathtubs can hold up to 150 liters of water. However, since bathtub filling events do not use the full capacity of the bathtub, we assumed that a bathtub filling event will use approximately two thirds of its capacity. Based on that, we identified efficient bathtub filling events as those that use less than or equal to 100 liters of water, typical events as those that use between 100 and 200 liters of water, and inefficient events as those that use more than 200 liters of water. Table 4.2 summarizes the technical performance of each end use type according to the Federal Standard and EPA specifications.

4.2.2. End Use-Level Water Demand Model Formulation

An end use water demand model can be formulated based on the premise that total water use for a household is the sum of all of the end uses of water. Given that, the total water use volume for an individual simulated household for a given day can be calculated as:

\[ V_{T,D} = \left( \sum_{i=1}^{n} B_i V_{i,D} \right) \]

where \( V_{T,D} \) is the total water use volume for a household (liter) on day of the week \( D \), \( B_i \) is a coefficient indicating the absence (0) or presence (1) of an end use \( i \), \( V_{i,D} \) is the volume of water used by end use type \( i \) during day of the week \( D \) (liter), and \( n \) is the number of end uses within the simulated household. The volume of water consumed
during the day by unclassified events that cannot be prescribed to a particular end use (e.g., leaks) (liter), is modeled as a separate end use.

Water end use technical performance, number of residents in a household, water use behaviors, variation in occupancy of the household on different days of the week, and demographic factors that vary across households and individual water use events within a household all affect the total volume of water used by each end use during a day \( (V_{i,D}) \). In order to account for this variability, the volume of each individual end use event is simulated in the model. To do so, we accounted for: 1) the number of individual water use events from each end use type that occurs during a day of the week \( D \), or frequency \( (f_{i,D}) \), and 2) the volume of each end use event \( j \) of type \( i \) \( (v_{i,j}) \) (Eqn. 2):

\[
V_{T,D} = \left( \sum_{i=1}^{n} B_{i} \sum_{j=1}^{f_{i,D}} v_{i,j} \right)
\]

(2)

this enables the model to estimate the total daily water use for each of the different end uses while accounting for variation in volumes of each individual water use event across each of the different end uses. Instead of assuming average volume and frequency estimates for events of each end use type, the model simulates the frequency of event occurrence for each household and day along with the volume of each individual end use event using a Monte Carlo sampling approach. We chose a Monte Carlo sampling approach rather than assuming average volume and frequency because we have observed that volume and frequency are not consistent across homes or days of the week and we were interested in the conservation potential associated with different event types, which depends on variability in event volumes. Another approach that could be used consists of choosing individual homes with detailed end use data and using the events for those
homes without manipulation. However, given the relatively small number of homes with detailed end use data, this would result in reusing the same events over and over which may not be representative of the distribution of events from the much larger set of homes to be simulated. A Monte Carlo simulation approach provides a wider variety of water use events to sample from and results in a smoother, more realistic distribution of events for simulated homes, providing an opportunity to simulate a wider variety of water use behaviors reflective of a broader group of residential water users.

Event frequency values for each day (Monday - Sunday) and a volume for each simulated event are drawn from cumulative distribution functions (CDFs) for frequency and volume derived from detailed event data for a subset of the households from a detailed water use monitoring study. By doing so, the model is able to simulate variability in water use across different households and across the different end use types.

To satisfy the input requirements of the model, detailed, disaggregated water end use event data obtained from smart metering studies are needed. Detailed water end use event data consist of individual water use events for a household and additional information about each event, including the date, start time, volume, duration, and flowrate. To simulate residential homes within a city using the model formulation above, a representative number of households with their detailed water end use data can be scaled up to the level of all residential homes within a city. However, most cities lack detailed water end use datasets. Existing end use studies have necessarily focused on a small group of households within a municipality boundary, and there have been few large scale studies to date (Boyle et al. 2013). Furthermore, existing studies may include bias associated with their spatial distribution, with most of them having been conducted in
Queensland, Australia and scattered cities across the U.S. (e.g., Jorgensen et al. 2009; Makki et al. 2015; Willis et al. 2013). These limitations have restricted existing end use demand models to places where detailed end uses of water studies were conducted.

For cities where no detailed water end use datasets are available, an alternative is to draw a sample of households with their detailed water end use data from one of the existing end use studies such that the water use behavior of the drawn sample is representative of the water use behavior of the households to be modeled (see Section 4.2.5 for how we did this for our case study). Similarity in water use can be quantified using data that are widely available for different cities (e.g., monthly billing data). The monthly water use data for households to be simulated can be used to calculate the overall water use probability distribution for those households, where the distribution shows the probabilities of occurrence of different monthly water use volumes for all households within the city. Then, monthly water use volumes for households with detailed water end use data are used to draw a representative sample of households from existing end use studies.

The input to the model is a comma-separated values (CSV) file that contains the water end use event data for the representative sample of households and the number of households to be simulated. The output of the model is a CSV file that contains the simulated water end use events for the number of residential households in the input (e.g., all single family residential homes in a modeled city). In the following sections, we first describe the Monte Carlo sampling procedure used in executing the model. We then describe in more detail how the model inputs were developed for our case study application in Logan, including how the representative sample of households with
detailed water end use data was selected and how CDFs input to the Monte Carlo sampling procedure were constructed. Following that, we describe how we validated the “existing conditions” model simulation results and then implemented the ability to simulate water conservation strategies.

4.2.3. Model Execution Procedure

The model initiates the sampling procedure from CDFs for event frequency so that water use behavioral factors for a simulated household are related. For example, a simulated household with a high toilet flush frequency is expected to have a high faucet use frequency. We used the cumulative distribution function that characterizes frequency of water use to rank households as having low (< 33rd percentile), medium (33rd - 66th percentile), or high (>66th percentile) frequency of water use, depending on their percentile ranking of number of events per day. We then devised a Monte Carlo sampling procedure to ensure that the frequency of end use events of the same type within the same simulated household for different days of the week were drawn from the same group of frequencies. For example, if the frequency of the first end use type is sampled from the low frequency group (< 33rd percentile), the frequency values for all simulated days for that end use type for the same household are sampled from the low frequency group. The model assumes that once a high, medium, or low frequency has been set for an end use type for a simulated household, that end use type for the simulated household remains in that category to preserve the same frequency behavior for end uses of the same type throughout different days of the week.

In order to account for the variation of technical performance of end uses across different households, we constrained the sampling process for events of the same type to
choose only events within the same level of technical performance (i.e., inefficient, typical, or efficient). For example, if the first event of one type is sampled from efficient events, all subsequent events of the same type are sampled from the same group of efficient events. This was implemented in our sampling procedure by sub-setting flowrates and/or volumes from different end use types into different groups based on their technical performance. We then devised a Monte Carlo sampling procedure to ensure that the simulated events of the same type share the same technical performance, but still capture observed variability across water use events. By doing this, we ensured a realistic water use behavior for each simulated household.

Using these Monte Carlo methods, we sampled from the distributions of event frequencies and event volumes/flowrates for each day of the week to generate a simulated set of events that when summed provide a water use estimate for each simulated household over a one-week period. For sampling purposes, we used the CDF for each input (event frequency and event volume/flowrate for different days of the week). The CDF’s x-axis encloses the range of possible values of an input, while its y-axis holds the non-exceedance probability values, which vary from 0 to 1. After generating CDFs for frequency and individual water use event volumes/flowrates for each day of the week using the event data input to the model, Equation 2 was evaluated as follows for each individual simulated residence:

- Select day of the week \( D \) for the simulated household
- For each end use type \( i \):
  - Determine the frequency of end use event type \( i \) (e.g., shower) for the simulated household for the selected day of the week by randomly
sampling from the CDF of frequency values for that day – generate a random number between 0 and 1 representing a non-exceedance probability and select the corresponding frequency from the x-axis (Figure 4.1). In the example below for the shower end use, the randomly generated non-exceedance probability value of 0.4 indicates that the simulated household is drawn from the medium water use frequency group. For other days of the week, narrow the randomly selected non-exceedance probability value for the same end use type to be within the range of medium water use frequency group (0.33-0.66).

- If the frequency is zero, the simulated household does not possess an end use \( i \). Set the value of \( B_i \) to 0 and the end use volume \( v_{i,j} \) to zero. If \( B_i = 1 \), proceed to the next step. The shape of the generated CDF curve of frequencies is influenced by the frequency values in the original data. For example, if 50% of all households in the representative sample do not have bathtub filling events, the CDF curve of frequencies will have a steeper slope segment at the beginning of the curve (Figure 4.2) indicating that many of the frequency values used to generate the distribution have a value of zero. This implies that the likelihood of a sampled household having a \( B_i \) value that equals to zero (no bathtub filling events) is high assuming that the sampling is random and unbiased.

- Based on end use type, generate three CDFs of water volumes or flowrates, one for efficient events, one for typical events, and one for inefficient events. For faucet and shower end uses, flow rates are used to
reflect their technical performance, while user behavior is captured in the duration for each event. For other end use types, including toilet, bathtub, clothes washer, and dishwasher end uses, only volume is considered since it is more relevant than the flow rate in terms of technical performance. An example of the CDFs used for shower event sampling is shown in Figure 4.3 with both shower event flowrate and duration CDFs used for sampling.

- For each event \( j \) in the set of end use events of type \( i \) defined by frequency \( f_{i,D} \):
  - For the first event of toilet, bathtub, clothes washer, and dishwasher \((j = 1)\), randomly pick a CDF curve of volumes from the inefficient, typical, and efficient distributions generated in the previous step for events of type \( i \). Determine an end use volume, \( v_{i,1} \), by randomly picking a volume from the selected CDF of volumes – generate a random number between 0 and 1 representing a non-exceedance probability and select the corresponding volume from the x-axis.
  - For the first event of faucet and shower end uses \((j = 1)\), Determine an end use flowrate, \( FR_{i,1} \), by randomly picking a flowrate from the selected CDF of flowrates – generate a random number between 0 and 1 representing a non-exceedance probability and select the corresponding flowrate from the x-axis. Determine an end use duration \( D_{i,1} \) by randomly picking a duration from the CDF of durations – generate a random number between 0 and 1
representing a non-exceedance probability and select the corresponding duration from the x-axis. Calculate the water use volume of the first event of faucet and shower end uses by multiplying its flowrate by its duration. In the example below, the first shower event was picked from the CDF for efficient showerheads, and its duration was randomly picked from the CDF of shower durations (Figure 4.4).

- For succeeding events ($j > 1$) of types toilet, bathtub, clothes washer, and dishwasher, determine an end use volume by randomly sampling from the CDF for event volume after narrowing the sampling range to a set of event volumes that matches the technical performance of the first selected event – generate a random number between 0 and 1 representing a non-exceedance probability and select the corresponding volume from the x-axis. For succeeding events ($j > 1$) of types faucet and shower, determine an end use flowrate by randomly sampling from the flowrate CDFs after narrowing the sampling range to a set of event flowrate values that match the technical performance of the first selected event. By doing this, we ensure a consistent technical performance of water use events of the same type in the same household for different days of the week. Determine an end use duration by randomly sampling from the CDF of shower durations. Calculate the water use volume of each event by multiplying the
flow rate of each event by its duration. In this example, succeeding shower events are sampled from the typical event CDF (Figure 4.4, Panel a) while their duration can be any value within the CDF curve of durations (Figure 4.4, Panel b).

- Add the volume of the current event $j$ to a total volume tally for event type $i$ for the current day $D$.
- For water use events that are not prescribed to a particular end use type, generate a single CDF of event volumes. Determine an event volume, $v_{i,1}$, by randomly picking a volume from the generated CDF of volumes – generate a random number between 0 and 1 representing a non-exceedance probability and select the corresponding volume from the x-axis. For succeeding events ($j > 1$) determine an end use volume by randomly sampling from the CDF of volumes. By not constraining the sampling procedure of water use events that are not prescribed to a particular end use type (e.g., leaks), we ensure realistic behavior of these events given that they have been observed to vary drastically from one day to another within the same home.
  - Add the total volume tally for events of type $i$ to the total daily volume tally for the current day $D$.
- Repeat the steps described above for each day of the week for each simulated residence until the number of residences in the input has been simulated.
4.2.4 Simulating Water Conservation Strategies

The total volume of water savings in liters is calculated in the model as the difference between water use before and after conservation actions are applied (e.g., installing a low-flow showerhead for a certain household will reduce overall water use by reducing water used by showers). The savings associated with conservation actions depend on the initial state of a household. For example, a household that already has efficient shower heads will not realize water savings by installing low-flow showerheads. The expected amount of water saved by making end uses more efficient can be calculated as:

\[ V_{S,D,i} = \left( \sum_{i=1}^{n} B_i \sum_{j=1}^{f_{i,D}} (v_{i,j} - v'_{i,j}) \right) \]

where \( V_{S,D,i} \) (liter) is the water savings from retrofitting end use \( i \) and/or changing water use behavior for the household on day of the week \( D \), \( B_i \) is a coefficient indicating the absence (0) or presence (1) of an end use \( i \), \( v_{i,j} \) is the volume of water used by end use type \( i \) for an individual event \( j \) during day of the week \( D \) (liter) before retrofitting, \( n \) is the number of end uses within the simulated household, and \( v'_{i,j} \) is the volume of water used by end use type \( i \) for an individual event \( j \) during day of the week \( D \) (liter) after retrofitting.

This expression enables the model to investigate technological and behavioral conservation actions at the household level. While households that already have efficient fixtures will not save water for conservation actions that involve retrofitting fixtures, conservation actions that involve behavioral change of water use can still be considered for those households. The amount of water savings is assumed by the model to be a
simple superposition of the effectiveness of each independent action. For example, if a household chooses to reduce shower lengths and reduce clothes washer use frequencies, the total effectiveness of those actions together is modeled as the sum of the effectiveness of each of those actions when implemented independently. The total water savings from adopting multiple conservation actions is then estimated as the sum of water savings associated with each implemented action, which can be denoted as:

\[ V_{S,D,T} = \sum_{i=1}^{n} V_{S,D,i} \]  

where \( V_{S,D,T} \) is the total water savings (liter) for day of the week \( D \), and \( V_{S,D,i} \) is as described above.

4.2.5 Case Study Application

The water end use model we developed can be applied to simulate any set of residential households where the following conditions are met: 1) monthly or more frequent water use data for the residential households to be simulated is available, and 2) there is a set of households with detailed water end use event data that are representative of the households to be simulated. As a demonstration case, we picked the city of Logan, Utah, USA as a medium sized municipality to demonstrate the capability of the model to simulate the indoor residential water use of all households in a city. Logan City is the hub of a growing metropolitan area in northern Utah's Cache Valley (Figure 4.5) and relies entirely on springs and groundwater wells to supply municipal water needs. Logan’s drinking water is drawn from groundwater in DeWitt Spring located in Logan Canyon to the east of the city. Although the spring generally provides a sufficient amount of water to supply the City, it is supplemented by four culinary wells that assist the supply,
primarily in the summer. More than 70% of the total supplied fresh water is consumed by the residential sector in Logan City. The majority of residential buildings in Logan are classified as single-family household (SFH), with 7,500 SFH connections reported in the city’s monthly water records. Single-family households account for 90% of residential users in Logan.

The City of Logan water utility provided us with monthly water use records collected from 2012 to 2018 for all SFH connections within the city. The provided billing dataset contains the total monthly water use volume in gallons per household along with other secondary attributes, including the billed days, square footage of the home, property number, account number, and bill date (Atallah, 2021). To select a representative sample of households with water end use event data for input to the model, we used data from the 2016 Residential End Uses of Water Study (REUWS) collected by AquaCraft, Inc. (DeOreo et al. 2016). The 2016 REUWS dataset provides information about individual water use events derived from high temporal resolution smart metering data. AquaCraft monitored 762 single-family households across 11 cities in the U.S. and Canada between 2000 and 2016 for a period of two weeks. They used their TraceWizard software (DeOreo et al. 1996) to disaggregate the high resolution flow trace from each household’s water meter to identify and classify individual water use events. The resulting dataset contains individual water use events along with several event attributes, including the date, start time, volume, and peak flow rate. In addition to the detailed end use dataset, the 2016 REUWS recorded daily water use for each participating household. Table 4.3 summarizes the geographical coverage and other parameters collected in the 2016 REUWS.
To draw a sample of 2016 REUWS households that is representative of the Logan households, we used average daily water use as the metric for comparison. The monthly water use data provided by Logan City and the 2016 REUWS daily water use dataset were collected at different temporal aggregations (monthly versus daily), and over different time periods (2010-2016 for REUWS versus 2012-2018 for Logan). To enable comparison across the datasets, we arranged both into a similar temporal aggregation. We used the years of 2014-2018 for Logan as the most recent five years of data. We downscaled the monthly billing data for all households in Logan to average daily water use by dividing the monthly water use volumes of each year by the number of billed days for each month. To ensure we were only accounting for indoor water use, we excluded summer months from the dataset where outdoor water use is anticipated and considered winter months only (January to March and November to December). We estimated four values of average daily water use for each household in the Logan dataset for each year, one value for each winter month, then averaged them together to get one estimate of daily water use for each household. For the 2016 REUWS dataset, we estimated average daily water use volumes for all households across all days by excluding irrigation events where they existed for all years available.

After calculating the average daily water use for each household in the Logan dataset, we used a weighted random sampling approach to identify a set of households from the 2016 REUWS dataset that would generate a probability distribution of average daily water use representative of the one generated from Logan households. Weighted random sampling utilizes PDF curves to randomly sample data points from a distribution (in this case the 2016 REUWS households) based on weights assigned to each data point.
in the sampling dataset based on the PDF of another dataset (in this case the Logan households). The sampling weights effectively set the likelihood with which households in the 2016 REUWS dataset will be selected so that the sampling procedure generates a set of households having a distribution of average daily water use that represents the distribution of average daily water use for Logan households as closely as reasonably possible. The following steps summarize the weighted random sampling procedure:

- Identify the range of values of average daily water use volume for households in both Logan and the 2016 REUWS datasets. Remove households from the 2016 REUWS dataset with daily water use volumes beyond the range of water use volumes of Logan dataset.

- Generate a PDF curve of average daily water use volumes for households in the Logan dataset (Figure 4.6). The x-axis of the PDF represents the range of average daily water use volumes for Logan households. The y-axis represents the probability density, or the likelihood of the corresponding value on the x-axis occurring. Since a PDF is a graphical representation of a numerical distribution where the outcomes are continuous, for each household in the 2016 REUWS dataset with average daily water use within the range of average daily water use values from the Logan dataset, there is a probability density value on the Logan dataset’s PDF curve.

- Calculate the probability density value for the average daily water use volume for each 2016 REUWS household using the PDF curve of the Logan dataset (Figure 4.6). The calculated probability density value for a 2016 REUWS household is called the sampling weight. The sampling weight sets the importance of each
household in the 2016 REUWS dataset such that the likelihood of a household being selected is equal to the probability density of that point from the Logan PDF.

- Normalize the sampling weight of each 2016 REUWS household by dividing weights by the summation of weights for all 2016 REUWS households. The summation of normalized weights from all 2016 REUWS households should equal to 1.

- Use the random.choice function from the NumPy Python package to randomly select a subset of 2016 REUWS households using the normalized weights for the 2016 REUWS households as input to the function. We also set the replace parameter of the function to be true to sample with replacement. We chose to sample with replacement given the small number of 2016 REUWS households (less than 400 households) compared to 7,500 households in the Logan dataset.

- The sampling function requires predefining the number of 2016 REUWS households to be selected. To identify the optimal number of 2016 REUWS households to select, we used a statistical test of equality metric to evaluate different sample sizes. Many statistical tests can be used to test the equality of continuous, one-dimensional probability distributions. The most common ones include the Chi-square test (Looney 2008), the Anderson-Darling test (Nelson 1998), and the Kolmogorov-Smirnov (KS) test (Massey et al. 1951). A one-sample KS test can be used to compare a sample (i.e., daily water use volumes drawn from the 2016 REUWS dataset) with a reference probability distribution (i.e., daily water use volumes obtained from Logan dataset) to determine whether
they are the same. An attractive feature of the KS test is that it does not depend on the underlying CDF being tested. The KS test uses the p-value significance level to examine whether two distributions are equivalent. The KS test returns a D statistic and a p-value corresponding to the D statistic. The D statistic is the absolute max distance between the CDFs of the two samples. The closer this number is to 0, the more likely it is that the two distributions are equivalent. The p-value returned by the KS test has the same interpretation as other p-values. If the p-value is lower than some significance level (e.g., \( \alpha = 0.05 \)), then the null hypothesis is rejected, signifying the modeled and observed results are not from the same distribution. If the p-value is greater than the \( \alpha = 0.05 \) significance level, then both datasets were drawn the same distribution. The KS test was implemented using the SciPy 1.7.2 Python Package.

- For the KS test configuration, we used an initial sample size of 50 2016 REUWS households, and then increased the sample size by one household on each iteration, and stopped when the population size of 7,500 households was reached. We estimated the D statistic and a p-value for each sample size and selected the sample size that produced the least D static value with a p-value greater than the \( \alpha = 0.05 \) significance level.

Utilizing the procedure described above, a total of 92 households that generated a probability distribution of average daily water use representative of Logan households was drawn from the 2016 REUWS dataset. The selected households resulted in minimum D static value, indicating that the daily water use volumes of the drawn households most closely represent the overall daily water use volumes of households in Logan dataset. The
selected sample of 92 households included 69 unique households and 23 replicated households. We used the detailed end use event data for all 92 households in this set to simulate the detailed water end use events for Logan residents. The PDF of average daily indoor water use of Logan households during winter months of the years of 2014-2018 versus the sample of households drawn from the 2016 REUWS dataset is shown in Figure 4.7.

4.2.6. Model Validation and Comparison

The simulation process resulted in estimates of daily water use volumes for 7,500 residential households located in Logan City for a period of one week. To confirm that the water use volumes from the simulation model accurately represent the water use behavior of residents in Logan City, we compared the simulated water use volumes to observed water use volumes retrieved from the monthly billing dataset. Given that simulated water use volumes were generated using Monte Carlo simulation, comparing them directly with observed data is not possible. Instead, we compared the distribution and characteristics of simulation results to the distribution and characteristics of the observed data to ensure that they match. Since the simulated water use volumes and the observed water use volumes from the monthly billing dataset have different temporal scales (simulated daily water use volume for one week period versus observed monthly water use volume), we first arranged both datasets into a common temporal aggregation. For the monthly water use records, we calculated average daily water use volumes for each household by dividing the total monthly water use volume for that household by the number of billed days using winter months data for the years between 2014 and 2018, then we averaged across winter months to get one value of average daily water use
volume for each household in the dataset. For the model results, we calculated the average daily water use volume for each household by summing the daily water use volume for the whole one week simulation period and dividing the total by seven days. To evaluate whether the actual average daily water use volumes and the simulated average daily water use volumes were drawn from the same distribution we used the KS test.

To evaluate how improving the representation of water uses at a detailed level within a water demand model can improve our ability to predict the water uses – our first research question – we evaluated the performance of the developed model in predicting the actual average daily water uses of households in Logan City dataset against other urban water demand simulation models. In our review of existing urban water demand models, we found that code is not openly available. In most cases, access was restricted or can only be obtained by contacting the authors. We could not replicate other end use models because source code was not available and their formulations/equations were not well enough described in the papers that we could re-implement them. Moreover, other end use modeling studies were restricted to communities where water end use data are available, which inhibits their ability to predict the detailed water use of other residential communities.

In response to these issues, we compiled a list of theoretical and empirical methodologies reported in urban water demand simulation papers published over the past two decades. We searched on different web search engines and scientific databases including Google Scholar, Zotero, and Mendeley for the following combination of words: “urban water demand model”, “water demand simulation”, and “residential water demand
model”. We then compiled a list with the methods and related publications retrieved with the above search, we reviewed and classified the list according to model replicability, equation availability, and directions to replicate the method presented in the paper. From this list, we selected a subset of models that meet the following criteria: 1) can simulate current water use conditions, 2) commonly used and recognized (e.g., regression), 3) we have input data for (e.g., landscaped area, census count), 4) well enough described in the paper that we could replicate them, and 5) the specific model selected is representative of a class of models reported in the literature. We then implemented those models to simulate current residential water use in Logan by generating 7,500 daily water use volumes that represent the number of residential water connections of Logan City.

Based on the aforementioned criteria, we replicated three different water demand models including an Ordinary Least Squares (OLS) model (Polebitski and Palmer, 2010), Piecewise Regression model (Chang et al., 2013), and Multiple Regression model (Arbués et al., 2010). Independent variables implemented to predict indoor water use in these models included socio-demographic variables (e.g., number of residents) and meteorological variables (e.g., precipitation). Demographic variables used as inputs in each model were retrieved from the Cache County GIS Parcel data website (https://www.cachecounty.org/gis/). These variables and the interaction between them were implemented differently in each model, however, where possible we used the same variables to simulate current residential water use in Logan City as did the authors of the prior modeling studies – e.g., like Polebitski and Palmer (2010), we used the building area (ft²), number of residents, income, property age, and household value to predict indoor water use using an OLS model.
To evaluate the reliability of each model in predicting current water use, we compared the cumulative distribution of average daily water use volumes estimated from the Logan dataset versus the cumulative distributions of average daily water use volumes obtained from different water demand simulation models including the model developed in this paper. Quantitatively, we utilized the KS statistical test on the CDFs output from the different models tested against the CDF of actual water use data from Logan dataset. We estimated the D statistic and a p-value for each model and presumed that the model that produced the smallest D static value with a p-value greater than the $\alpha=0.05$ significance level is the best performing model.

4.2.7. Water Conservation Actions

To quantify the water saving potential associated with different technological and behavioral conservation actions – our second research question – we examined the efficiency of water end uses of different types across all simulated households to identify households and end use types with water conservation potential. We then quantified the water saving potential associated with a set of potential technological and behavioral conservation actions (Table 4.4) for water use in the model. Technological actions include actions associated with the technical performance and water use efficiency of different end use types inside a household (e.g., retrofitting an inefficient showerhead). Behavioral actions include actions associated with the water use behavior of a household’s residents (e.g., reduce shower length).

Based on the end use type, we used either the volume or flowrate of the simulated water use events to investigate the technical performance of the existing end uses and compared them with typical and efficient end uses. Volume was used to reflect the
technical performance of toilet, bathtub, clothes washer, and dishwasher end uses.

Retrofitting actions on these end uses were applied on events with volumes exceeding efficient volumes, and the expected water use after retrofitting was calculated as the volume of events from retrofitted fixtures. For faucet and shower end uses, flowrate was used to reflect their technical performance. Retrofitting actions on these end uses were applied on events with flowrates exceeding efficient flowrates, and the expected water use after retrofitting was calculated as the flowrate of retrofitted fixtures multiplied by the duration of their corresponding events. For all retrofitting actions, we assumed that a retrofit would change an end use’s technical performance, but not user behavior.

Besides retrofitting actions, we used the duration, volume, and the number of simulated events per household per day to account for behavioral change in water use for those end uses that are associated with the behavior of residents (e.g., reduce shower duration). Four different actions were examined (Table 4.4). For the fixing leaks action, we assumed that 50% of total unclassified events are leaks and thus residents of a household can reduce unclassified water use inside their home by 50% by fixing leaks. Thus, the amount of water saved by fixing leaks was calculated as the volume of unclassified events divided by two. For the reducing shower length action, we first identified long shower events as events that last longer than the 80th percentile of all shower events in a simulated household and assumed that residents of that household can reduce their long shower events down to the 80th percentile of all shower durations. The amount of water saved by reducing shower duration per household was calculated for simulated shower events that exceeded the 80th percentile shower length as the difference in shower duration before and after the duration reduction of each event multiplied by the
flowrate of the event. The same procedure was used for the reducing faucet duration action. To reduce clothes washer event frequency, we assumed that residents can reduce their current frequency of laundry events by 10%, although other frequencies could easily be simulated.

4.3. Results and Discussion

The end use water demand model simulated 367,500 water use events for 7,500 households in the City of Logan over the period of one week. The average execution time for the water demand model, which simulates one week of water use for both existing conditions and the water conservation scenarios for all households was approximately six hours on a 2017 MacBook Pro laptop computer with a 3.1 GHz quad-core Intel i7 processor and 16 GB of RAM.

4.3.1. Model Comparison and Applicability

All of the models we tested resulted in a cumulative distribution curve relatively similar to the Logan data (Figure 4.8), but the end use model we developed most closely matched the distribution of the Logan City data. Moreover, it provides detailed end use results that could assist water suppliers in identifying opportunities for incentive programs to encourage water conservation and monitoring effectiveness of those programs where the other models do not. The disparity between our model and other models in simulating current water use indicates that using water end use events to predict total daily water use volumes instead of using regression approaches will likely generate more realistic results.

Using the KS test, the resulting p-value of the end use model was 0.84, which is higher than the 0.05 significance level. The D statistic value for our model was 0.049,
which is less than the D statistic value of the other models we tested (Table 4.5). Thus, both the observed average daily water use volume records calculated from monthly billing data for winter months during 2014-2018 and the simulated average daily water use volumes obtained from the end use model have very similar distributions.

To assess the applicability of the end use model in predicting current conditions given that it was based on data from 2014 – 2018, we explored the variability in indoor water use of households in the Logan dataset for those years (Figure 4.9). The white dots in the figure represent the medians of the distributions, the thick grey bar in the center represents the interquartile range, the thin grey line represents the whole distribution, except for outlier data points, wider sections of the violin represent a higher approximate frequency of data points in that section, and thinner sections represent a lower approximate frequency of data points in that section. As illustrated in Figure 4.9, the overall distribution of indoor water use for Logan City households was fairly stable between 2014 and 2018.

With respect to water end uses simulated by the model, toilet flushing accounted for the largest volume of indoor water use, followed by showers, faucets, clothes washers, and bathtubs, which matches the relative contribution of indoor water use type reported by the 2016 REUWS (Table 4.6). While each simulated household had a unique behavioral pattern, bulk behavior across all simulated households, matched that of the 2016 REUWS with the biggest difference between the two studies being 3%. Appendix A compares the distribution of duration, volume, and flowrate in all simulated households for different end use types.
4.3.2. End Use Efficiency and Water Conservation Potential

The maximum amount of water savings is expected when all retrofitting actions are implemented at the same time. However, toilets have two retrofit options that are mutually exclusive. To maximize water savings, we assumed that typical and inefficient toilets are retrofitted to highly efficient toilets since they save more water than low flush toilets (Table 4.4). Based on that, the expected proportion of water saved if all technological conservation actions are implemented together at the same time ranged from 0% to 50% for individual households and totaled approximately 23% of total water use across all households.

Generally, technological conservation actions are effective and more likely to persist since they involve changing fixtures. Adoption of behavioral conservation actions may vary from one household to another and even in the same household from one day to another since they are associated with physiological, social, and behavioral changes of household’s residents (Addo et al. 2018). In the matter of durability of technological actions compared to behavioral actions, technological actions can perform up to 20 years (EPA 2021a), while behavioral actions have been shown to be effective for six months at most (Schultz et al. 2019). Retrofitting toilets to 4.8 LPF toilets had the most water saving potential (assuming behavioral water use does not change) at a total water savings of 10,550,179 LPD for Logan City. On the behavioral side, reducing shower lengths can save up to 1,946,922 LPD for Logan City. Figure 4.10 summarizes the water savings rates for both technological and behavioral conservation actions. The box plots in both figures show the distribution of daily water savings across all 7,500 Logan households, with each box showing a different implemented water conservation action.
4.4. Conclusions

We developed an end use water demand model that simulates detailed household water use using Monte Carlo techniques. The model advances existing end use modeling studies that used similar techniques by accounting for differences in event frequency among different days of the week, simulating variabilities in event volume or flowrate and duration for different end use types for different days of the week while constraining the technical performance of different end uses, incorporating all expected indoor water use events in the simulation process, providing estimates of baseline use and maximum conservation potential at the individual home and city levels, and developing a generic model that can be scaled to any number of single family residential homes.

The model uses event data from a sample of households in the 2016 REUWS dataset as input to simulate water use behavior of Logan residents. The input dataset consists of detailed end use event data for a sample of households that are representative of the households to be simulated. The model is generally applicable and can be modified to simulate the detailed water end uses of other cities with the following constraints: 1) the city to be simulated must have available water use records (e.g., monthly or more frequent billing data records), and 2) there must be a sample of households in the 2016 REUWS (or another dataset) such that the water use of the sample households is representative of the water use of the households to be simulated (e.g., similar daily water use distribution). Since the 2016 REUWS dataset collected data across 11 different cities in the U.S. and considered monthly data in selecting households with different water use behaviors, we anticipate that the likelihood of extracting a sample of households from the 2016 REUWS dataset with an overall water use representative to other cities is high,
although selection of households for simulation would be enhanced by the availability of more households with detailed end use data.

In our case study application, we demonstrated how existing water use event data can be used to predict the detailed water use of other residential communities, with the only required data from the city to be simulated being their monthly water use billing records. Since we used data from 2014-2018 only, we acknowledge that the water demand model quantifies the detailed water use and evaluates potential conservation in the context of the years of 2014-2018. However, Figure 4.9 shows that indoor water use was stable for Logan over this period, and we anticipate similar water use behavior from many other communities across the U.S. Thus, the model should reflect current conditions and conservation potential, but may need to be adjusted in the future to reflect changes in indoor water use behavior.

The retrofitting and behavioral conservation actions for selected end uses showed high potential for water conservation across the 7,500 residential households we simulated. The expected upper band of total water savings at the household level is 2,700 Liter/household-day and the expected total water savings at the city level is more than 20 million Liter/day, representing approximately 23% of all water currently used indoors by residential users in Logan City.

The type of detailed water end use simulation produced by the model, including practical water conservation actions and the ability to simulate their savings at the city level, could assist water utilities in identifying opportunities for incentive programs that will have the greatest impact and to encourage water conservation. Effectiveness of these efforts could be monitored using new methods for collection of high resolution water use
data or through more conventional comparison of pre- and post-retrofit monthly data, although effectiveness of multiple, simultaneous programs would be difficult to separate using only monthly data. Furthermore, this type of modeling can be used for forecasting demand and determining how water use patterns may change over time in response to population growth, demographic shifts, behavioral change, and improvements in technology. It may also be useful in better characterizing how and when water is being used inside of households and in the design of improvements to the residential water distribution infrastructure. Supplying this type of information to water users can also be a tool for impacting water use behavior and managing demand.

Software Design and Implementation

The water demand model was designed and developed using Version 3.7 of the Python programming language. It was implemented as a single script that can be executed using any Python programming environment and was developed using the SciPy 1.7.2, Pandas 1.3.4, NumPy 1.21.4, and scikit-learn 1.0.1 packages for Python.

Data Availability

Code for the water demand model is open-source, released under the Creative Commons Attribution CC BY license, and available in the HydroShare repository (Atallah et al. 2021). Documentation of hardware and software requirements, Python Jupyter notebooks with examples of workflows implementing each part of the code, and instructions for running the code are provided in the HydroShare resource.
Acknowledgments

This research was funded by the United States National Science Foundation under grant number 1552444. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional financial support was provided by the Utah Water Research Laboratory at Utah State University. We acknowledge Logan City for their cooperation in this study and for providing the monthly water use data for residential connections in the city.
REFERENCES


## Tables

### Table 4.1. Approaches used for different end use modeling studies.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Incorporated end uses</strong></td>
<td></td>
<td>Faucet, toilet, shower, clothes washer, dishwasher, bathtub, unclassified</td>
<td>Faucet, toilet, shower, clothes washer, dishwasher, bathtub, unclassified</td>
<td>Faucet, toilet, shower, clothes washer, dishwasher, bathtub, unclassified</td>
<td>Faucet, toilet, shower, clothes washer, dishwasher, bathtub, unclassified</td>
<td>Faucet, toilet, shower, dishwasher</td>
</tr>
<tr>
<td><strong>Water conservation prediction</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Software used</strong></td>
<td>Python</td>
<td>Matlab</td>
<td>Unknown</td>
<td>NetLogo</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Open source software license</strong></td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Daily variation per end use type</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Representative set of households used in the simulation</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Handling unclassified events</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Variable water use per end use type</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Model propagation at city scale</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

### Table 4.2. Technical performance by end use type.

<table>
<thead>
<tr>
<th>End use type</th>
<th>Inefficient event</th>
<th>Typical event</th>
<th>Efficient event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toilet</td>
<td>Volume &gt; 6.1 LPF&lt;sup&gt;a&lt;/sup&gt;</td>
<td>4.8 LPF &lt; Volume &lt; 6.1 LPF</td>
<td>Volume &lt; 4.8 LPF</td>
</tr>
<tr>
<td>Faucet</td>
<td>8.3 &gt; LPM&lt;sup&gt;b&lt;/sup&gt;</td>
<td>5.7 LPM &lt; Flowrate &lt; 8.3 LPM</td>
<td>Flowrate &lt; 5.7 LPM</td>
</tr>
<tr>
<td>Shower</td>
<td>Flowrate &gt; 9.5 LPM</td>
<td>7.6 LPM &lt; Flowrate &lt; 9.5 LPM</td>
<td>Flowrate &lt; 7.6 LPM</td>
</tr>
<tr>
<td>Clothes washer</td>
<td>Volume &gt; 110 liter/load</td>
<td>70 liter/load &lt; Volume &lt; 110 liter/load</td>
<td>Volume &lt; 70 liter/load</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>Volume &gt; 13 liter/cycle</td>
<td>6 liter/cycle &lt; Volume &lt; 13 liter/cycle</td>
<td>Volume &lt; 6 liter/cycle</td>
</tr>
<tr>
<td>Bathtub</td>
<td>Volume &gt; 200 liter/filling</td>
<td>100 liter/filling &lt; Volume &lt; 200 liter/filling</td>
<td>Volume &lt; 100 liter/filling</td>
</tr>
</tbody>
</table>
Table 4.2. Data collected in the 2016 Residential End Uses of Water study.

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical Coverage</td>
<td>Clay County, GA; Denver, CO; Fort Collins, CO; Peel, Ontario; San Antonio, TX; Scottsdale, AZ; Tacoma, WA; Toho, FL; Waterloo, Ontario; Aurora, CO; Austin, TX; Cary, NC; Chicago, IL; Edmonton, Alberta; Henderson, NV; Miami, FL; Mt. View, CA; New Haven, CT; Otay, CA; Philadelphia, PA; Portland, OR; Santa Barbara, CA; Santa Fe, NM</td>
</tr>
<tr>
<td>Temporal Coverage</td>
<td>AquaCraft recorded water flow through each individual customer’s water meter every 10 seconds for a period of two weeks. Detailed flow and end use datasets were successfully obtained from 762 households.</td>
</tr>
<tr>
<td>Demographics</td>
<td>Number of residents, rent versus own, highest level of education in the household, annual household income.</td>
</tr>
<tr>
<td>End uses</td>
<td>Toilet, bathtub, faucet, shower, clothes washer, dishwasher, evaporative/swamp cooler, pressure regulator, unclassified.</td>
</tr>
<tr>
<td>Fixture information</td>
<td>Presence of low-flush, ultra-low-flush, dual-flush toilets, number of showerheads in showers, whether toilets/shower heads/clothes washer/dishwasher have been replaced in past 10 years, irrigation system type (sprinkler, hose, automatic timer, drip irrigation, and rain sensor)</td>
</tr>
</tbody>
</table>

Table 4.4. Proposed water conservation actions and their associated characteristics in terms of water use*.

<table>
<thead>
<tr>
<th>Technological conservation actions</th>
<th>Characteristic flows/volumes</th>
<th>Behavioral conservation actions</th>
<th>Characteristics flows/volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrofit showerheads</td>
<td>7.6 LPM</td>
<td>Fix leaks</td>
<td>50% less leaks</td>
</tr>
<tr>
<td>Retrofit faucets</td>
<td>5.7 LPM</td>
<td>Reduce faucet use duration</td>
<td>Varies</td>
</tr>
<tr>
<td>Retrofit toilets with low flush toilets (LFT)</td>
<td>6.1 LPF</td>
<td>Reduce shower duration</td>
<td>Varies</td>
</tr>
<tr>
<td>Retrofit toilets with highly efficient toilets (HET)</td>
<td>4.8 LPF</td>
<td>Reduce clothes washer use frequency</td>
<td>10% less use</td>
</tr>
<tr>
<td>Retrofit top load washers</td>
<td>~ 100 liter/load</td>
<td></td>
<td></td>
</tr>
<tr>
<td>with front load washers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Values reported in this table were retrieved from the EPA WaterSense and EnergyStar Websites (EPA 2021a, EPA 2021b).
Table 4.5. Estimated D statistic and a p-value for each model.

<table>
<thead>
<tr>
<th>Model</th>
<th>D statistic</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>End use model</td>
<td>0.049</td>
<td>0.84</td>
</tr>
<tr>
<td>OLS model</td>
<td>0.204</td>
<td>2.1 X 10^{-15}</td>
</tr>
<tr>
<td>Multiple regression model</td>
<td>0.180</td>
<td>2.1 X 10^{-15}</td>
</tr>
<tr>
<td>Piecewise linear regression model</td>
<td>0.160</td>
<td>2.1 X 10^{-15}</td>
</tr>
</tbody>
</table>

Table 4.6. Relative contribution of indoor water use type.

<table>
<thead>
<tr>
<th>End use type</th>
<th>Simulation results</th>
<th>2016 REUWS</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathtub</td>
<td>4%</td>
<td>4%</td>
<td>0%</td>
</tr>
<tr>
<td>Clothes washer</td>
<td>15%</td>
<td>18%</td>
<td>3%</td>
</tr>
<tr>
<td>Faucet</td>
<td>22%</td>
<td>22%</td>
<td>0%</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>0.5%</td>
<td>2%</td>
<td>1.5%</td>
</tr>
<tr>
<td>Shower</td>
<td>26%</td>
<td>23%</td>
<td>3%</td>
</tr>
<tr>
<td>Toilet</td>
<td>31%</td>
<td>28%</td>
<td>3%</td>
</tr>
<tr>
<td>Unclassified</td>
<td>1%</td>
<td>3%</td>
<td>2%</td>
</tr>
</tbody>
</table>
Figures

Figure 4.9. Sampling process from the CDF of shower frequency.

Figure 4.10. Example CDF for bathtub filling events.

Figure 4.11. Panel a: flowrate CDFs for showers based on their technical performance, and Panel b: duration CDFs for showers based on their behavioral performance.
Figure 4.12. Panel a: sampling from flowrate CDF for shower events based on their technical performance, and Panel b: sampling from duration CDF for shower events based on their behavioral performance.

Figure 4.5. Distribution of single-family households in Logan.
Figure 4.6. Weight assigning to each 2016 REUWS dataset household. The average daily water use volume for each 2016 REUWS household is intersected with the PDF curve for the Logan dataset to obtain a probability density value for each 2016 REUWS household. These probability density values are used as the weights for 2016 REUWS households in the sampling procedure.

Figure 4.7. PDF for average daily indoor water use of Logan's households versus the sample of households from the 2016 REUWS.
Figure 4.8. Cumulative distributions for observed average daily water use volume and simulated average daily water use volume of different simulation methods for all residential connections in Logan.

Figure 4.9. Distributions of average daily indoor water use for Logan City households between 2014-2018.
Figure 4.10. Ranges of potential water savings for technological and behavioral conservation actions. Some outliers were removed to enhance the readability of the figure.
CHAPTER 5
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

The research presented in this dissertation builds upon existing smart water metering and end use disaggregation studies to develop and demonstrate a new, open source, and reproducible disaggregation and classification method that can be executed on a water meter datalogger as a computational node. The produced record of classified events can support a wide range of analysis and modeling applications typically undertaken by interdisciplinary research and integrated management teams. We combined innovative smart water metering technology with new scientific research in computer science, distributed computing, and water use behavior that will advance the scientific data and knowledge base and cyberinfrastructure for sustainably managing urban water supplies. The output of this work may help in creating and bringing key information to both water providers and water consumers, enabling better water system decision making on both the supply and demand sides.

The significance of the work presented in this dissertation includes presentation of a design and implementation of hardware and software tools that enable recording, analyzing, and transferring high temporal resolution data as well as case studies that demonstrate the suitability of these tools for addressing existing gaps in existing water end use disaggregation algorithms, centralized data management, and water end use models aimed at better understanding residential water use behavior. This research demonstrates how water end use measurement studies, including the targeted residential modeling effort, can provide detailed data (e.g., water end use events, volumes, and flows) for areas where water as a resource is scarce and where residential water systems
may be most vulnerable. By characterizing how residential water is utilized inside households, these results provide information that may be useful for city engineers and planners in better understanding how and when water is used, in the development of best management practices, and the design of improvements to the residential water distribution infrastructure.

Chapter 2 presented the CIWS disaggregator algorithm, an open source, semi-supervised water end use disaggregation and classification tool that can break down the total water use observed at the household level into different end uses (e.g., toilet, shower, faucet, etc.). The tool uses non-intrusive monitoring data collected at high temporal resolution from a residential household’s water meter along with machine learning techniques to disaggregate water use into discrete end use events. A data collection campaign was conducted in Utah with high temporal resolution data collected from 31 residential households in the cities of Logan and Providence. To overcome the challenges associated with classifying events, we used a semi-supervised classification approach, which requires an initial, small number of labeled events but can then identify and classify events from any residential household for which high temporal resolution flow data is available. The work presented in this chapter was driven in part by the fact that, for most other studies that have worked on end use disaggregation algorithms, neither the source code nor the data are available for testing or further advancement. We have openly shared the algorithm code and anonymized data to advance the availability and functionality of open tools for water end use disaggregation studies. To evaluate the generalizability of the tool, we successfully disaggregated and classified high temporal resolution data from different households with different meter sizes and types, providing
evidence that it can be used across a wide range of residential households with differing meter types and sizes. While we anticipate that the algorithm will work for any meter capable of providing high temporal resolution flow data, further testing with new datasets would be needed to confirm this.

Chapter 3 presented the design and implementation of a datalogging and computational device, called the CIWS Computational Node, designed to work on top of existing, analog, magnetically driven, positive displacement, residential water meters without disturbing the functionality of the meter. The CIWS Computational Node can collect, analyze, and transfer high temporal resolution data and their associated water end use events. The device can be deployed to the field autonomously and without supplemental power for approximately 4 weeks while collecting data at a 4 second time interval, executing computational codes on the data, and sending resulting data and computed end use information to a remote secured server for storage. The device advances currently available devices on the commercial market, which are limited to only collecting high temporal resolution water use data. It demonstrates how several challenges associated with collection and use of smart metering data can be addressed, including: 1) enabling high resolution data collection without replacing existing meters; 2) demonstrating how the big data generated by smart meters can be “shrunk” into actionable information; 3) reducing the need for storing and transmitting large data volumes; and 4) reducing the need for centralized databases, computational capabilities, and information technology for storing and processing smart metering data.

The CIWS Computational Node was successfully deployed and tested at the Utah Water Research Lab (UWRL) under optimal conditions (e.g., constant temperature with a
dedicated power supply) and on the water meter for a single-family residential household in Providence City, UT, USA under variable temperature and power configurations to demonstrate successful sensing, data logging, and computational capabilities on existing residential analog meters. The CIWS Computational Node is a low cost (~$200) device, which facilitates its use in cases where cost limits the collection and transferring of high temporal resolution data. The computational node’s hardware design and software are open source, available for potential reuse, and can be adapted to specific research needs.

Chapter 4 presented a new method for simulating water demand from residential users. The model is based on water end use event data derived from high resolution monitoring of residential water use and simulates residential water use behavior in individual households by randomly sampling events for different end use types for a selected day of the week and then aggregating the sampled water end use events per day to estimate the daily water use per household. The model was used to simulate “existing conditions” within the City of Logan, Utah and then was used to test a set of short and long term conservation actions to quantify the conservation potential in each individual household. The case study presented in Chapter 4 demonstrates the scalability and usability of water end use records in revealing the water use behavior of residential communities where detailed water end use data is not available. In the case study, we analyzed the variability of water use, in terms of timing and distribution of end uses, efficiency of end uses, and water conservation potential of residential households in the city of Logan, Utah. The data used in this study were collected at different temporal resolution scales (monthly, daily, and individual classified events). The model is generally applicable and can be modified to simulate the detailed water end uses of other
cities. The applicability of the model is constrained by the availability of at least monthly water use records for the simulated place and the existence of a sample of households in the detailed water end use dataset such that the water use behavior of the selected households is similar to the water use behavior of the simulated place.

This dissertation presents hardware and software that advance the suite of existing cyberinfrastructure for residential smart water metering and its associated applications. The tools presented here enhance the capabilities of existing technologies or advance the availability and functionality of open tools for smart metering applications. There is a clear need for open and reproducible approaches that enable other researchers to test, replicate, reuse, and build upon existing work. Further improving the replicability of smart metering applications, including water end use disaggregation and classification tools, can narrow the gap between research and practice and promote replicability as a vital practice in science and engineering. In an attempt to advance the availability of smart water metering applications and ensure that other scientists are able to access, use, and contribute to the tools and components developed, the output of this dissertation including the hardware and software was documented and stored in online, open access repositories. These repositories will also enable other researchers to contribute for advancing beyond the limitations imposed by lack of available data and the proprietary nature of existing software.

Internationally, smart metering is a new technology, and there have been few large-scale deployments to date (Boyle et al., 2013). As such, a small number of studies have examined residential water use behavior using smart metering data. Most of these have been conducted in Australia (e.g., Mead and Aravinthan, 2009; Beal et al., 2013;
Makki et al., 2013; Willis et al., 2013), where severe water shortages have required innovations in both demand management and water conservation strategies. Very few comparable studies have been conducted in the U.S. This leaves questions about how water use behavior might be different in Australia, where extreme water shortages have fundamentally changed the perceptions of water users, versus the western U.S., where water is in short supply but where conditions as severe as those in Australia have not yet been realized. Additional limitations of existing studies include small and unrepresentative samples sizes, bias associated with volunteer sample groups, and short sampling periods that may not reveal longer-term behavior patterns (e.g., seasonal changes in use). More comprehensive data collection campaigns are needed to address existing gaps in understanding of residential water use, including short-term and long-term behavior patterns.

Smart metering can provide water providers and consumers with near real-time information on consumption. One key way in which these data can be used is in providing feedback to consumers about their water consumption in efforts to promote water savings (Sønderlund et al., 2016). However, available cyberinfrastructure for doing this is limited, and, to date, few studies have tested the role and potential for behavior change through direct feedback to water consumers (Boyle et al., 2013). The premise is that by providing more information about water use (e.g., more than a monthly bill) water literacy among consumers will increase, empowering them to better understand their consumption and make informed decisions about water use. Providing detailed water use information to consumers follows on developments in the energy sector, where research into electricity use feedback suggests that a 5-15% reduction in demand is achievable
However, there are still many questions about what is the most effective medium for (e.g., letters, emails, apps, web portals, in home displays, etc.) and what types of reports and/or visualizations are most effective in communicating feedback to consumers (Sønderlund et al., 2016).

The promises of smart metering have been well described in this dissertation, and some of them have been demonstrated to some degree in existing studies. However, many of the potential benefits of smart metering systems are still unrealized due to the cost of upgrading existing meter networks and the lack of focus on the supporting cyberinfrastructure. For example, there has been important work in developing algorithms for disaggregating high frequency water use data into individual end uses (DeOreo et al., 1996; Nguyen et al., 2013; Nguyen et al., 2014; and this dissertation), but the cyberinfrastructure to do this consistently on a large scale does not yet exist. Stewart et al. (2018) summarize many of the still unfulfilled benefits of smart metering, including: better citywide urban water planning; near real-time water distribution network analysis; targeted water demand management; evidence-based water demand forecasting; proactive water loss management; targeted demand efficiency; addressing water-related energy demand; evidence based economic assessments; reform of water pricing schemes; and heightened customer satisfaction. While the results of this dissertation provide steps in the right direction, realizing all of these benefits will require continued development of new hydroinformatics techniques (e.g., data collection and management techniques, end use disaggregation algorithms, visualization and reporting, etc.) and an infusion of this expertise into the current and next generation of water utility operators.
Besides the challenges described in the literature, we identify a list of potential challenges and future directions that need to be addressed including:

1) **Design of centralized or distributed systems to store and process high temporal resolution data collected by smart meters:** While a centralized system would allow regular checks on the quality of the collected data, the bandwidth limitations of conventional telemetry systems and the associated transmission costs can inhibit the practicality of such systems. These limitations of centralized systems can be minimized by using edge computing distributed systems where data processing is performed at or near where the data are collected to extract and transmit only actionable data products to a centralized location. However, the quality of collected raw data cannot be quantified since only data products are transmitted. We designed an edge computing water meter capable of transferring both raw pulse data and classified water use events. Yet, further research and implementations are needed to quantify the frequency and amount of raw data transmission required for regular data quality checks.

2) **Privacy of detailed water end use data:** Information privacy is defined as the right of an individual over their personal data, determines who can collect and manage such data and determines to what extent personal data can be communicated to others (Westin, 1966). Privacy concerns in the water field are currently underestimated in most countries. The rapid increase in usage of smart metering technologies and their associated applications, including collection of detailed water use and behavioral information implies increased privacy risks. Further
research is needed to define privacy concerns in the water field associated with the collected data within and between communities.

3) **Water use behavior:** Most previous water end use studies focused on correctly characterizing water end uses associated with different indoor water use fixtures. However, understanding the timing of different water end uses is still unrealized. Timing represents key information to understand water use behaviors and may be useful in designing personalized demand management strategies (e.g., suggesting deferral of the use of some appliances to off-peak hours). Consequently, understanding timing and peak hours could provide crucial information for identifying both typical consumption behaviors and patterns as well as consumption outliers (e.g., leaks).

4) **More effective strategies for influencing water use behavior change:** The impact of technological water conservation actions is limited to households with inefficient water fixtures (e.g., inefficient showerheads). Behavioral conservation actions are generally less effective but can be applied to a broader range of households regardless of the efficiency of their water fixtures. In the energy sector, providing feedback to users about their energy use and conducting regular educational intervention has been shown to be effective in controlling energy use (e.g., Abrahamse et al., 2007). Further research is needed in the water sector regarding the use of feedback to reduce water use, particularly with respect to the most effective feedback format, whether the effect persists over time, as well as assessments of costs and benefits of feedback (Buys et al., 2013).
REFERENCES


APPENDICES
Appendix A. Overall Water Use and Characterizing Individual End Uses

Overall Water Use

Average indoor and outdoor water use for each of the studied houses is reported in Table A.1, and Figure A.1 shows the distribution of indoor water use across the component end uses for the study period. We observed different water use behavior in both indoor and outdoor water use across the five households. Outdoor water use had greater variability across the households than indoor water use, although behavioral differences were observed in both. Both daily and daily per capita outdoor water use were calculated using summer data collection period only. Outdoor water use ranged from 764.2 LPD in Household 1 to a high of 15,216.6 LPD in Household 4, and the per capita outdoor water use ranged from 191.1 LPCD in Household 1 to 2173.8 LPCD in Household 4 (Table A.1). The variability in outdoor water use is attributed to the sizes of landscapes irrigated and the irrigation practices used (automatic versus manual irrigation system). On a per capita basis, houses three through five drastically exceeded the Utah Division of Water Resources estimate of average per capita outdoor water use for the state of Utah of 106 gpcd (Utah Division of Water Resources 2005).

Toilet flushing accounted for the largest volume of indoor water use, followed by showers, clothes washers, faucets, and bathtubs. While each household exhibited a unique behavioral pattern, toilet flushing and showers constituted the largest volume of water across all five households. Across all five houses, showers accounted for an average of approximately 32% of total indoor water use, toilets accounted for an average of 28%, clothes washer events contributed an average of 13%, faucet and dishwasher events contributed an average of 17%, bathtub filling events contributed an average of
6%, and unclassified events contributed the least with an average of only 3% of total indoor water use volume.

We compared the relative contribution of indoor water use to the 2016 Residential End Uses of Water study (DeOreo et al. 2016) (Table A.2). The biggest difference between the two studies was observed in the unclassified events category, which was reduced from 16% in the 2016 Residential End Uses of Water study to 3% in this study. The big difference in unclassified water uses between the two studies is due to the fact that 32% of households in the 2016 Residential End Uses of Water study had water leakage as high as 2,271 L per household per day. Given the small number of houses in our study set, we did not observe this level of leakage.

While it is tempting to compare the indoor water use from this study to the 2016 Residential End Uses of Water study, it is important to point out that the participating households differed between the two studies. In this study, we analyzed the water use of five households located in one geographic area. The 2016 Residential End Uses of Water study examined a much larger set of households spread more diversely throughout the United States. However, both sets of results demonstrate the utility of having high resolution data and disaggregating it to reveal water use behavior.

Characterizing Individual End Uses

We used the classified events to calculate the frequency of use for each end use type in each study household (Table A.3). We also segregated the events by season (the summer data collection period versus the winter data collection period) to investigate the seasonal variation in total water end uses (Figure A.2). The frequency of indoor water use remained mostly homogeneous across both seasons, where faucets were the most
frequently utilized end use fixture during both summer and winter days at approximately 87 uses per household per day. This is 36 more per household per day than the 51 uses per household per day reported by the 2016 Residential End Uses of Water study. However, the average number of people per household for this study (5.2 residents per household) is almost twice the average number of people per household in the 2016 Residential End Uses of Water Study (2.76 resident per household). Toilet flushing was the second most frequently utilized fixture at approximately 21 uses per household per day of water use events, which is almost twice the frequency of toilet use reported by the 2016 Residential End Uses of Water study at 12.4 uses per household per day. Bathtub filling was the least frequently utilized indoor water end use accounting for only 0.4 uses per household per day, which is consistent with the findings from the 2016 Residential End Uses of Water study. The right panel of Figure A.2 shows that there was a larger number of events during the summer data collection period than the winter period, which may reflect the fact that residents of the homes are at home and using water more often during summer than winter. This is consistent with the findings from the Hussien et al. (2018) study. Despite the relatively small water use volume constituted by unclassified events, their frequency was high and thus were excluded from Figure A.2 for better visualization.

We also used the classified events to investigate the variation in indoor water use for each end use type in each study household. Appendix B provides a detailed comparison of indoor water use behavior for each of the houses, including the total number of events, number of events per day, number of events per capita per day, event volume, event duration, and event flowrate for each of the end uses we identified. We
also provide a comparison of the distributions of event duration, volume, and flowrate to illustrate differences in technical performance of fixtures across the five houses and compare our results to those of the 2016 Residential End Uses of Water study (DeOreo et al. 2016). Here, we provide a brief summary of the overall performance of fixtures across the five houses.

Toilet event data revealed that the majority of flushes used less than 12 liters per flush. The average toilet flush volume across all five houses was 8 liters per flush, which exceeds the current U.S. federal standard of 6.3 liters per flush (U.S. Environmental Protection Agency 2014). Shower data revealed that the average shower event used approximately 48 liters and lasted for 6.5 minutes with an average flowrate of 7.6 LPM, which is less than the shower flowrate mandated by the U.S. national plumbing code standard of 9.5 LPM (U.S. Environmental Protection Agency 2014). Clothes washer data revealed that the average water use volume per load of clothes was 63.5 liters, which is less than the standard 76 liters per load for washing machines (U.S. Environmental Protection Agency 2014). Faucet data revealed that most faucet events had flowrates less than the 5.7 LPM maximum faucet flowrate specified by the national plumbing code standard (U.S. Environmental Protection Agency 2014).
Tables

Table A.1. Indoor and outdoor water use per household.

<table>
<thead>
<tr>
<th>Household ID</th>
<th>Outdoor water use (LPD)</th>
<th>Per capita outdoor water use (LPCD)</th>
<th>Indoor water use (LPD)</th>
<th>Per capita indoor water use (LPCD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>764.2</td>
<td>191.1</td>
<td>698.4</td>
<td>174.6</td>
</tr>
<tr>
<td>2</td>
<td>801.3</td>
<td>200.3</td>
<td>456.9</td>
<td>114.2</td>
</tr>
<tr>
<td>3</td>
<td>6597.2</td>
<td>1099.5</td>
<td>920.8</td>
<td>153.5</td>
</tr>
<tr>
<td>4</td>
<td>152216.6</td>
<td>2173.8</td>
<td>619.6</td>
<td>88.5</td>
</tr>
<tr>
<td>5</td>
<td>4172.5</td>
<td>834.5</td>
<td>640.1</td>
<td>128</td>
</tr>
<tr>
<td>Average</td>
<td>5510.4</td>
<td>899.8</td>
<td>667.2</td>
<td>131.8</td>
</tr>
</tbody>
</table>

Table A.2. The relative contribution of indoor water use in this study versus the 2016 Residential End Uses of Water study (DeOreo et al. 2016).

<table>
<thead>
<tr>
<th>End use type</th>
<th>This study</th>
<th>2016 Residential End Uses of Water Study</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathtub</td>
<td>6%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>Clothes washer</td>
<td>13%</td>
<td>16%</td>
<td>3%</td>
</tr>
<tr>
<td>Faucet and Dishwasher</td>
<td>17%</td>
<td>22%</td>
<td>5%</td>
</tr>
<tr>
<td>Shower</td>
<td>32%</td>
<td>20%</td>
<td>12%</td>
</tr>
<tr>
<td>Toilet</td>
<td>28%</td>
<td>24%</td>
<td>4%</td>
</tr>
<tr>
<td>Unclassified or Leaks</td>
<td>3%</td>
<td>16%</td>
<td>13%</td>
</tr>
</tbody>
</table>

Table A.3. Total number of water use events by household and end use.

<table>
<thead>
<tr>
<th>Household</th>
<th>Bathtub</th>
<th>Clothes washer</th>
<th>Faucet</th>
<th>Irrigation</th>
<th>Shower</th>
<th>Toilet</th>
<th>Unclassified</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>62</td>
<td>4,578</td>
<td>48</td>
<td>185</td>
<td>621</td>
<td>78,828</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>24</td>
<td>2,825</td>
<td>13</td>
<td>80</td>
<td>444</td>
<td>1,502</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>64</td>
<td>1,710</td>
<td>50</td>
<td>163</td>
<td>801</td>
<td>3,311</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>11</td>
<td>2,945</td>
<td>77</td>
<td>151</td>
<td>711</td>
<td>792</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>55</td>
<td>1,866</td>
<td>36</td>
<td>136</td>
<td>771</td>
<td>1,535</td>
</tr>
<tr>
<td>Total</td>
<td>64</td>
<td>216</td>
<td>13,924</td>
<td>224</td>
<td>715</td>
<td>3,348</td>
<td>85,968</td>
</tr>
</tbody>
</table>
Figures

Figure A.1. Average fraction of indoor water use volume by end use type for the five homes in the case study dataset.

Figure A.2. Seasonal variation in the frequency of water use events by end use type.
Appendix B. Detailed Analysis of Indoor Water Use

Toilets

According to the walk-through survey results, each of the five houses had at least three toilets. A total of 3,348 toilet flushes were recorded over the combined data collection periods for the five houses, averaging 21 flushes per household per day and 4 flushes per capita per day. Distributions of toilet flushing duration, volume, and flowrate and their probability densities for each participating household are shown in Figure B.1. The white dot in the figure represents the median, the thick grey bar in the center represents the interquartile range, the thin grey line represents the whole distribution, except for outlier data points, wider sections of the violin represent a higher approximate frequency of data points in that section, and thinner sections represent a lower approximate frequency of data points in that section. The majority of flushes used less than 12 liters per flush. The average toilet flush volume across all five houses was 8 liters per flush with a standard deviation of 4.8 liters per flush, which is, on average, consistent with the current federal standard of 6.3 liters per flush (U.S. Environmental Protection Agency 2014). The average time to refill the toilet tank was 1 minute with a standard deviation of 0.5 minutes. Figure B.1 shows that households 1, 2, and 5 have multiple toilets that perform differently, resulting in the lobes in the violin plots. Households 3, and 4 appear to have very homogenous toilet flushes. A summary of the other statistics for toilet flushes from the study households is provided in Table B.1.

Showers

According to the walk-through survey results, there were at least 3 showers per household in the study. A total of 715 showers were recorded over the combined data
collection period for the five houses. The distributions of shower duration, volume, and flowrate and their probability densities are shown in Figure B.2. Household 2 has a larger proportion of shorter showers averaging around 5 minutes and 50 liters. The variability in duration and volume is higher in the other houses. The average shower event used 48 liters and lasted for 6.5 minutes with an average flowrate of 7.6 liters per minute, which is less than the shower flowrate mandated by the U.S. national plumbing code standard of 9.5 liters per minute (U.S. Environmental Protection Agency 2014). The majority of shower events used less than 75 liters per shower, and most shower events lasted less than 10 minutes. While the distribution of both shower volumes and durations was right-skewed, the distribution of shower flowrates appeared to be normally distributed with 82% of all showers operating at a flowrate less than 9.5 liters per minute. A summary of the other statistics for showers from the study households is provided in Table B.2.

Clothes Washers

According to the walk-through survey results, all households had at least one clothes washing machine. All clothes washing machines observed were newer, front loading, high efficiency machines. A total of 216 clothes washing events were recorded over the 160 logged days. Across all households in the study, the average number of laundry loads per day was 1.35, which is almost twice the average number daily loads recorded by (DeOreo et al. 2016). However, the average number of loads washed per capita per day remained the same at 0.3 load per capita per day between the two studies. The distributions of clothes washing event duration, volume, flowrate and their probability densities are shown in Figure B.3. The average water use volume per load of clothes was 63.5 liters with a standard deviation of 14 liters per load. A summary of the
other statistics for clothes washer events from the study households is provided in Table B.3.

Faucets

Faucets were the most frequently used fixture with a total of 13,924 recorded events. The distributions of faucet event duration, volume, and flowrate and their probability densities are shown in Figure B.4. The average faucet event used approximately 1.5 liters and lasted for less than half a minute. The standard deviation of both volume and duration was greater in magnitude than their mean, which indicates that faucet events are highly variable. This result was anticipated since faucet water use is highly influenced by the behavior of the user and the type of water use activity for which faucets are used (e.g., washing dishes uses more water and takes more time than washing hands). Most faucet events had flowrates less than 4.5 liters per minute with an average flowrate of 3 liters per minute, which is less than the Environmental Protection Agency’s maximum faucet flowrate of 4.5 liters per minute (U.S. Environmental Protection Agency 2014). A summary of the other statistics for faucet events from the study households is provided in Table B.4.

Bathtubs

Bathtubs were the least utilized indoor water use fixture with 64 total events. The distributions of bathtub filling duration, volume, and flowrate and their probability densities are shown in Figure B.5. The average bathtub filling event used around 100 liters and required 7 minutes.
Dishwashers

According to the walk-through survey, all participating households used dishwashers for dishware cleaning. Using the set of labeled events, we were able to identify the physical features of dishwasher events. However, their physical features were indistinguishable from faucet events, so we were not able to separate dishwasher and faucet events. Thus, dishwasher events were grouped with faucet events. In an attempt to investigate why we were unable to distinguish dishwasher events from faucet events while other studies have reported results for both, we purchased the labeled event dataset for the City of Denver, Colorado, USA (DeOreo et al. 2016). These data were classified and labeled by AquaCraft. We investigated the physical features of their labeled faucet and dishwasher events and did not find any distinguishable features between them; moreover, some of their labeled dishwasher events were identical to their labeled faucet events in the same dataset. Although the 2016 Residential End Uses of Water report was based on this data, and results for both faucet and dishwasher events were reported separately, we were unable to determine how AquaCraft distinguished between dishwasher and faucet events using what we found to be an indistinguishable set of physical features.

Unclassified Events

The events we placed in the “Unclassified” category encompass two types of events: 1) outlier water use events we were not able to attribute to a certain end use type, which may include dual toilet flushes or long duration faucet events; and 2) events with durations of one-time step and one pulse volume. Unclassified events accounted for only 3% of the total water use volume across the five houses; however, these events appeared
frequently in the trace data and accounted for more than 40% of all recorded events in some households. Given the uncertainty around the nature of events in these categories, we excluded them from the analysis.

Tables

Table B.1. Summary statistics for toilet use.

<table>
<thead>
<tr>
<th>Household</th>
<th>Total number of flushes</th>
<th>Average daily flushes (flush/day)</th>
<th>Average daily flushes per capita (flush/capita-day)</th>
<th>Average flush volume (L)</th>
<th>Average flush duration (min)</th>
<th>Average flush flowrate (LPM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>621</td>
<td>17.3</td>
<td>4.3</td>
<td>8.6</td>
<td>1.3</td>
<td>6.9</td>
</tr>
<tr>
<td>2</td>
<td>444</td>
<td>14.3</td>
<td>3.6</td>
<td>10.7</td>
<td>1.1</td>
<td>10.1</td>
</tr>
<tr>
<td>3</td>
<td>801</td>
<td>28.6</td>
<td>4.8</td>
<td>7.5</td>
<td>1.0</td>
<td>7.3</td>
</tr>
<tr>
<td>4</td>
<td>711</td>
<td>25.4</td>
<td>3.6</td>
<td>7.2</td>
<td>1.0</td>
<td>7.4</td>
</tr>
<tr>
<td>5</td>
<td>771</td>
<td>20.8</td>
<td>4.2</td>
<td>11.1</td>
<td>1.1</td>
<td>11.5</td>
</tr>
<tr>
<td>Average</td>
<td>670</td>
<td>21.3</td>
<td>4.1</td>
<td>8.0</td>
<td>1.1</td>
<td>8.6</td>
</tr>
</tbody>
</table>

Table B.2. Summary statistics for showers.

<table>
<thead>
<tr>
<th>Household</th>
<th>Total number of showers</th>
<th>Average daily showers (shower/day)</th>
<th>Average daily showers per capita (shower/capita-day)</th>
<th>Average shower volume (L)</th>
<th>Average shower duration (min)</th>
<th>Average shower flowrate (LPM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>158</td>
<td>5.1</td>
<td>1.28</td>
<td>43.0</td>
<td>7.3</td>
<td>5.9</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>2.3</td>
<td>0.65</td>
<td>44.0</td>
<td>4.9</td>
<td>9.1</td>
</tr>
<tr>
<td>3</td>
<td>163</td>
<td>5.8</td>
<td>0.97</td>
<td>50.4</td>
<td>5.9</td>
<td>9.0</td>
</tr>
<tr>
<td>4</td>
<td>151</td>
<td>3.4</td>
<td>0.77</td>
<td>48.0</td>
<td>6.7</td>
<td>7.3</td>
</tr>
<tr>
<td>5</td>
<td>136</td>
<td>3.7</td>
<td>0.73</td>
<td>53.0</td>
<td>7.0</td>
<td>8.0</td>
</tr>
<tr>
<td>Average</td>
<td>143</td>
<td>4.5</td>
<td>0.88</td>
<td>47.8</td>
<td>6.5</td>
<td>7.6</td>
</tr>
</tbody>
</table>

Table B.3. Summary statistics for clothes washer loads.

<table>
<thead>
<tr>
<th>Household</th>
<th>Total number of loads</th>
<th>Average daily loads (loads/day)</th>
<th>Average daily loads per capita (loads/capita-day)</th>
<th>Average load volume (L)</th>
<th>Average load duration (min)</th>
<th>Average load flowrate (LPM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62</td>
<td>1.7</td>
<td>0.4</td>
<td>63.5</td>
<td>69.1</td>
<td>10.3</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>0.8</td>
<td>0.2</td>
<td>68.9</td>
<td>4.3</td>
<td>16.1</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>2.3</td>
<td>0.4</td>
<td>65.7</td>
<td>5.6</td>
<td>12.1</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>0.4</td>
<td>0.1</td>
<td>56.6</td>
<td>6.1</td>
<td>9.3</td>
</tr>
<tr>
<td>5</td>
<td>55</td>
<td>1.5</td>
<td>0.3</td>
<td>53.5</td>
<td>4.7</td>
<td>11.6</td>
</tr>
<tr>
<td>Average</td>
<td>43.2</td>
<td>1.35</td>
<td>0.3</td>
<td>63.5</td>
<td>5.6</td>
<td>11.7</td>
</tr>
</tbody>
</table>
Table B.4. Summary statistics for faucet events.

<table>
<thead>
<tr>
<th>Household</th>
<th>Total number of faucet events (events)</th>
<th>Average daily faucet events (events/day)</th>
<th>Average daily faucets events per capita (events/capita-day)</th>
<th>Average faucet event volume (L)</th>
<th>Average faucet event duration (min)</th>
<th>Average faucet flowrate (LPM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4,578</td>
<td>127.2</td>
<td>31.8</td>
<td>0.89</td>
<td>0.43</td>
<td>1.77</td>
</tr>
<tr>
<td>2</td>
<td>2,825</td>
<td>91.1</td>
<td>22.8</td>
<td>1.35</td>
<td>0.33</td>
<td>4.10</td>
</tr>
<tr>
<td>3</td>
<td>1,710</td>
<td>61.1</td>
<td>10.2</td>
<td>2.04</td>
<td>0.54</td>
<td>3.80</td>
</tr>
<tr>
<td>4</td>
<td>2,945</td>
<td>105.2</td>
<td>15.0</td>
<td>1.41</td>
<td>0.51</td>
<td>3.12</td>
</tr>
<tr>
<td>5</td>
<td>1,866</td>
<td>50.4</td>
<td>10.1</td>
<td>1.48</td>
<td>0.31</td>
<td>3.39</td>
</tr>
<tr>
<td>Average</td>
<td>2,785</td>
<td>86.0</td>
<td>18.0</td>
<td>1.31</td>
<td>0.44</td>
<td>2.99</td>
</tr>
</tbody>
</table>

Table B.5. Summary statistics for bath filling events.

<table>
<thead>
<tr>
<th>Household</th>
<th>Total number of baths (baths)</th>
<th>Average daily baths (baths/day)</th>
<th>Average daily baths per capita (baths/capita-day)</th>
<th>Average bath volume (L)</th>
<th>Average bath duration (min)</th>
<th>Average bath flowrate (LPM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>0.28</td>
<td>0.07</td>
<td>86.6</td>
<td>6.9</td>
<td>12.7</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.06</td>
<td>0.02</td>
<td>113.2</td>
<td>7.2</td>
<td>15.9</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>1.11</td>
<td>0.18</td>
<td>106.6</td>
<td>7.1</td>
<td>15.0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.04</td>
<td>0.01</td>
<td>113.2</td>
<td>8.8</td>
<td>12.9</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>0.54</td>
<td>0.11</td>
<td>96.7</td>
<td>7.2</td>
<td>13.7</td>
</tr>
<tr>
<td>Average</td>
<td>12.8</td>
<td>0.41</td>
<td>0.08</td>
<td>100.7</td>
<td>7.1</td>
<td>14.2</td>
</tr>
</tbody>
</table>

Figures

Figure B.1. Distributions of toilet flushing duration, volume, and flowrate.
Figure B.2. Distributions of shower event duration, volume, and flowrate. Some outliers with a duration of more than 30 minutes have been removed to improve the readability of the plot.

Figure B.3. Distributions of clothes washer event duration, volume, and flowrate. Some outliers with a duration of more than 12 minutes have been removed to improve the readability of the plot.
Figure B.4. Distributions of faucet event duration, volume, and flowrate. Some events with a duration of more than 2.5 minutes have been removed to improve the readability of the plot.

Figure B.5. Distributions of bathtub filling event duration, volume, and flowrate.
Appendix C. Distribution of Duration, Volume, and Flowrate in all Simulated Households by End Use

Figures C1 to C6 compare distributions of duration, volume, and flowrate in all simulated households for different end use types. These plots have been scaled to exclude some outliers to facilitate easier comparisons. Among different households, the average toilet flushing volume was significantly lower in efficient toilets at an average of 3 LPF, compared to an average 13 LPF observed in households with inefficient toilets. Inefficient showerheads had an average flowrate of 10 LPM, which is almost twice the average flowrate of efficient showerheads at 6 LPM.

Figures

Figure C.1. Distributions of toilet flushing flowrate, duration, and volume across simulated households.
Figure C.2. Distributions of shower event flowrate, duration, and volume across simulated households. Some outliers with a duration of more than 15 minutes have been removed to improve the readability of the plot.

Figure C.3. Distributions of clothes washer event flowrate, duration, and volume across simulated households. Some outliers with a duration of more than 10 minutes have been removed to improve the readability of the plot.

Figure C.4. Distributions of dishwasher event flowrate, duration, and volume across simulated households.
Figure C.5. Distributions of bathtub event flowrate, duration, and volume across simulated households. Some outliers with a volume of more than 300 liters have been removed to improve the readability of the plot.

Figure C.6. Distributions of faucet event flowrate, duration, and volume across simulated households. Some outliers with a duration of more than five minutes have been removed to improve the readability of the plot.
CURRICULUM VITAE
Nour A. Attallah, PhD
Department of Civil and Environmental Engineering
Utah Water Research Laboratory
Utah State University, 8200 Old Main Hill, Logan, UT 84322-8200 USA
Phone: (435) 754-5722
Email: nour.atallah@usu.edu

WATER CYBERINFRASTRUCTURE SUMMARY

- **Six+ years** of national and international research experience focused on cyberinfrastructure for intelligent water supply, water demand data disaggregation for residential and nonresidential users, and development of processing codes for water engineering applications
- Worked with the city of Logan, UT to design cyberinfrastructure solutions, including streamlining processing of high temporal (< 1 minute) residential water use data for visualization and analysis
- Proficiency in programming and database software including Python, C#, VBA (Visual Studio), and SQL
- Experience in Hydraulic/Hydrological modeling tools including Bentley’s CAD solutions, HEC-RAS, HEC-HMS and EPANet
- Experience teaching drinking water courses and leading labs
- Internship and part-time work experience working with international donors, helping develop and design water distribution networks using WaterCAD

EXPERIENCE

Graduate Research Assistant: Full-time
Utah Water Research Laboratory, Logan UT
*July 2018 – Ongoing*

- Developed cyberinfrastructure to better support the collection, management, and use of smart water metering data
- Designed and manufactured reliable, low-cost smart water meters that can collect high resolution water use data, and serves as a field-based computational mote capable of executing classification and disaggregation algorithms on the trace of high-resolution data collected at the end use point
- Installed a network of datalogger in more than 30 residential properties in Cache Valley, Utah, to collect residential water use data
- Designed cyberinfrastructure to streamline processing high temporal (< 1 minute) residential water use data for visualization and analysis
- Quantified residential water use for different water fixtures using high temporal resolution data collected on smart meters
- Quantified the demographics of water users to advance understanding of water use behavior and the ability to extract decision-relevant information from smart meter data streams
- Developed and tested an end use water demand model that can simulate and predict residential water use behavior at a city level using high resolution smart metering data
- Designed data base solutions based on City of Logan’s Utilities Department needs for non-residential water users
- Evaluated drought-tolerant plants, including turfgrasses, and studied methods of propagating drought-tolerant native plants and releasing new plant varieties
- Developed new practices that specify simple rules to determine the appropriate amount and timing for irrigating trees in urban landscapes
- Quantified industrial water use for different shift schedules and different manufacturing operations
- Developed open-source tools to disaggregate and characterize non-residential water use into their end use components
Research and Teaching Assistant: Full-time
An-Najah National University, Nablus, West Bank, Palestine
December 2014 – December 2015
- Responsible for 4 lab sections each semester
- Assisted with grading assignments and exams
- Led review sessions and taught 500+ students
- Courses taught: Fluid Mechanics and Hydraulics Lab, and Environmental Engineering Lab

Hydraulic and Water Resource Engineer: Part-time
GFA Consulting Group
December 2014 – December 2015
- Worked as part of multi-disciplinary project delivery teams providing technical hydraulic engineering and water resources support for district duties
- Supported field tests and inspections of pump stations and pipelines
- Participated in a citywide infrastructure development plan that involved design and production of entire hydraulic systems including pumps, valves, hydraulic motors, and storage tanks
- Developed HEC-HMS and HEC-RAS sensitivity models for two new pump station designs

Summer Intern
GFA Consulting Group
May 2014 – September 2014
- Helped develop and design water distribution networks using WaterCAD for five villages in the West Bank through a grant by the German International Development Bank (KfW)
- Oversaw the construction and filed progress reports, and helped run stakeholder meetings
- Project served a population of +100,000

SOFTWARE AND PROGRAMMING LANGUAGES

- Geographic Information System (GIS) tools: ArcMAP
- Programming Languages: Python, C#, and VBA (Visual Studio)
- Statistical Computing and Graphics: RStudio, Stata, and Tableau
- Structured Database Management Systems (SQL): MySQL, SQL Lite, and PostgreSQL
- Hydraulic Modeling Software: AutoCAD, WaterCAD, SewerCAD, StormCAD, EPANet, HY-8, and HEC-RAS
- Hydrological Modeling Software: SWAT, WEAP, and HEC-HMS
- Optimization Modeling Software: GAMS
PUPLICATIONS AND PRESENTATIONS

Journal Papers in Print or Press

Journal Papers Under Review
- Bastidas Pacheco, Horsburgh, J. S., Attallah, N.A., (2021) Variability in Consumption and End Uses of Water for Residential Users in Logan and Providence, Utah, USA

Theses and Dissertations

Conference Presentations

EDUCATION
- Ph.D., Civil and Environmental Engineering, Utah State University, Logan, UT, Spring 2022
- M.S., Civil and Environmental Engineering, Utah State University, Logan, UT, Spring 2018
- B.S., Civil Engineering, An-Najah National University, Nablus, Palestine, Fall 2014