Predicting Short-Term Water Consumption for Multi-Family Residences

by

Irish J. Medina

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2018

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Smart water meters have been installed across Abbotsford, British Columbia, Canada, to measure the water consumption of households in the area. Using this water consumption data, we develop machine learning and deep learning models to predict daily water consumption for existing multi-family residences. We also present a new methodology for building machine learning models to predict daily water consumption of new housing developments. This thesis contains three main contributions: First, we build machine learning models which include a feature engineering and feature selection step to predict daily water consumption for existing multi-family residences in the city of Abbotsford. This is motivated by the recent development direction towards denser living spaces in urban areas. We present the steps of the model building process and obtain models which achieve accurate predictive performance. Second, we present a new methodology for building machine learning models to predict daily water consumption for new multi-family housing developments at the dissemination area level. Currently, the models used in the industry are simple baseline models which can lead to an overestimation of predicted water consumption for new developments, leading to costly and unnecessary investments in infrastructure. Using this new methodology, we obtain a machine learning model which achieves a 32.35% improvement over our best baseline model, which we consider a significant improvement. Third, we investigate the use of deep learning models, such as recurrent neural networks and convolutional neural networks, to predict daily water consumption for multi-family residences. In our case, the main advantage of deep learning models over traditional machine learning techniques is the capability of deep learning models to learn data representations, allowing us to omit the feature engineering and feature selection steps and thereby allowing water utilities to save valuable time and resources. The deep learning models we build achieve comparable performance to traditional machine learning techniques.

Acknowledgements

Thank you, Professor Peter van Beek, for giving me the opportunity to work under your supervision and for your strong encouragement throughout this endeavour. Your encouragement and confidence greatly motivated me to keep persisting towards the goal and to never give up.

Thank you to my thesis committee members, Professor Jesse Hoey and Professor Krzysztof Czarnecki, for taking the time to examine my thesis and for your valuable input and feedback.

Next, I would like to thank the people who have helped me get to where I am today:

Thank you, Professor Carson Leung, for reaching out to me during undergrad and encouraging me to apply for a summer research assistantship under your supervision. This opportunity provided me with the first step towards pursuing a graduate degree, which I am tremendously grateful for.

Thank you, Dr. Gary Van Domselaar, for giving me the opportunity to be part of something bigger than myself. It was an extreme pleasure to have worked on a project so important and from which I was given so much autonomy and responsibility.

Carson and Gary, thank you for helping me realize my full potential and for your immensely strong encouragement throughout all my endeavours. I would not be where I am today if it wasn't for you.

Thank you to all the supervisors and mentors who I've had the privilege to work with along this journey. Thank you for seeing my potential and giving me every opportunity to succeed. Thank you for granting me the freedom and autonomy towards achieving my goals. Thank you for pushing me to further elevate myself in all aspects.

Thank you, Ness and Nestor Medina, for being incredibly amazing cousins to grow up with. I really appreciate the support you have given me throughout the years and for all the adventures that we've had growing up in Winnipeg.

Finally, thank you, Melanie Medina, for your unconditional support and for all the fun and joy you bring into my life. You inspire me to be the best person I can be every day.

This research was funded by the following scholarships and awards: ACM SIGHPC/Intel Computational & Data Science Fellowship, Generation Google Scholarship, HP Helion OpenStack Scholarship, NSERC Canada Graduate Scholarship - Master's Program, Ontario Graduate Scholarship, Queen Elizabeth II Graduate Scholarship in Science and Technology, QNX Graduate Scholarship, University of Waterloo President's Graduate Scholarship. Thank you to the selection committees for considering and selecting my application.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Smart water meters have been installed across Abbotsford, British Columbia, Canada, to measure the water consumption of households in the area. Using this water consumption data, we develop machine learning and deep learning models to predict daily water consumption for existing multi-family residences in the city of Abbotsford. In addition, a new methodology is introduced for building machine learning models to predict daily water consumption for new multi-family housing developments. This chapter discusses the main research contributions of the thesis and provides an outline of the following chapters of the thesis.

1.1 Research Contributions

In this section, we outline the research contributions of the thesis. This thesis contains three main contributions and are as follows:

- First, we build machine learning models to predict daily water consumption for existing multi-family residences in the city of Abbotsford. We discuss in detail the methodology used to build these machine learning models and the resulting model performance. As a second part to this contribution, we analyze the determinants of multi-family water consumption in detail. For this contribution, we obtain models with an accurate predictive accuracy and gain additional knowledge on the determinants of multi-family water consumption through our analysis.

- Second, we introduce a new methodology for building machine learning models which predict daily water consumption for new multi-family housing developments in the planning stage. We discuss in detail the steps of this new methodology and how the resulting machine learning models compare to baselines currently used in the industry. We use traditional machine learning techniques as opposed to deep learning models since features for new housing developments are not time series data. For this contribution, we obtain, using the new methodology introduced, machine learning models which significantly improve over baseline models currently used in the industry.

- Third, we build deep learning models which do not require a feature engineering and feature selection step to predict daily water consumption for existing multi-family residences in Abbotsford. We discuss the model building process in detail and elaborate on how the performance of deep learning models compares to traditional machine learning techniques such as SVR and decision trees. For this contribution, using deep learning models we obtain comparable performance to traditional machine learning techniques which require a feature engineering and selection step, saving a significant amount of time in the model building process.

To our knowledge, these three research contributions have not yet been attempted in the current literature.

1.2 Organization of Thesis

In this section, the organization of the thesis is outlined. Chapter 2 discusses the general and technical background used throughout the thesis. The next chapters discuss the three main contributions of the thesis:

- Chapter 3 focuses on building machine learning models to predict daily water consumption for existing multi-family residences in the city of Abbotsford. The chapter discusses the steps of the model building process, reports the performance of models, investigates model predictions, and discusses the determinants of multi-family water consumption.

- Chapter 4 introduces a new methodology for building machine learning models to predict daily water consumption for new multi-family housing developments. Each step of the new methodology is described in detail, the performance metric for each machine learning model is reported, model predictions are investigated, and the determinants of water consumption for new developments are discussed.

- Chapter 5 investigates the use of deep learning models, such as recurrent neural networks and convolutional neural networks, to predict daily water consumption for existing multi-family residences. The main advantage of deep learning is these models are capable of learning data representations, allowing us to omit the feature engineering and selection steps. The performance of deep learning models is compared with traditional machine learning approaches.

In Chapter 6, the thesis is concluded by discussing the main contributions of the thesis and the results.

Chapter 2

Background

In this chapter, we discuss the general background and the technical background of the thesis. The general background section covers the required background knowledge used throughout the thesis. The technical background section provides an introduction to the technical methods used throughout the thesis for readers who are unfamiliar with these methods.

2.1 General Background

In this section, we discuss the general background and the common terminology used throughout the thesis. We define what a dissemination area is, we describe the different types of multi-family residences, we introduce the concept of smart water meters, and investigate the datasets used throughout the thesis in detail.

2.1.1 Dissemination Areas

Cities in Canada, including Abbotsford, are partitioned into dissemination areas. A dissemination area (DA) is a small geographic area which contains a population of 400 to 700 people and is composed of adjacent dissemination blocks. Dissemination blocks are the smallest standardized geographic area from which census data is obtained. To protect privacy, census data is obtained from dissemination blocks and aggregated at the dissemination area level. The entirety of Canada is partitioned into dissemination areas.

A dissemination area must follow specific criteria to meet operational constraints and to be useful for data analysis, as specified in [10] and summarized here:

1. Dissemination areas remain geographically stable over time.
2. The boundaries of dissemination areas are formed by roads and features such as railways and power transmission lines.
3. The population of dissemination areas is kept uniform at 400 to 700 people.
4. Dissemination areas have been delineated based on the population counts of the prior census.
5. Dissemination areas are kept at a compact shape when possible.
6. A dissemination area may only contain a maximum of 99 dissemination blocks.

Abbotsford is partitioned into a total of 193 dissemination areas. A dissemination area is uniquely identified by a four digit code. This four digit code is preceded by a two digit province code and a two digit census division code. For Abbotsford, the two digit province code is 59 and the two digit census division code is 09. Together, these eight digits form a DAUID. Table 2.1 lists out the 87 dissemination areas of Abbotsford that are present in our water consumption dataset for multi-family residences. The table also includes the number of meters recording water consumption data for properties within the dissemination area and the total number of housing units in the dissemination area from which water consumption data is recorded. Figure 2.1 shows the city of Abbotsford as partitioned into dissemination areas. The areas shaded in grey are the 87 dissemination areas that are present in our dataset.

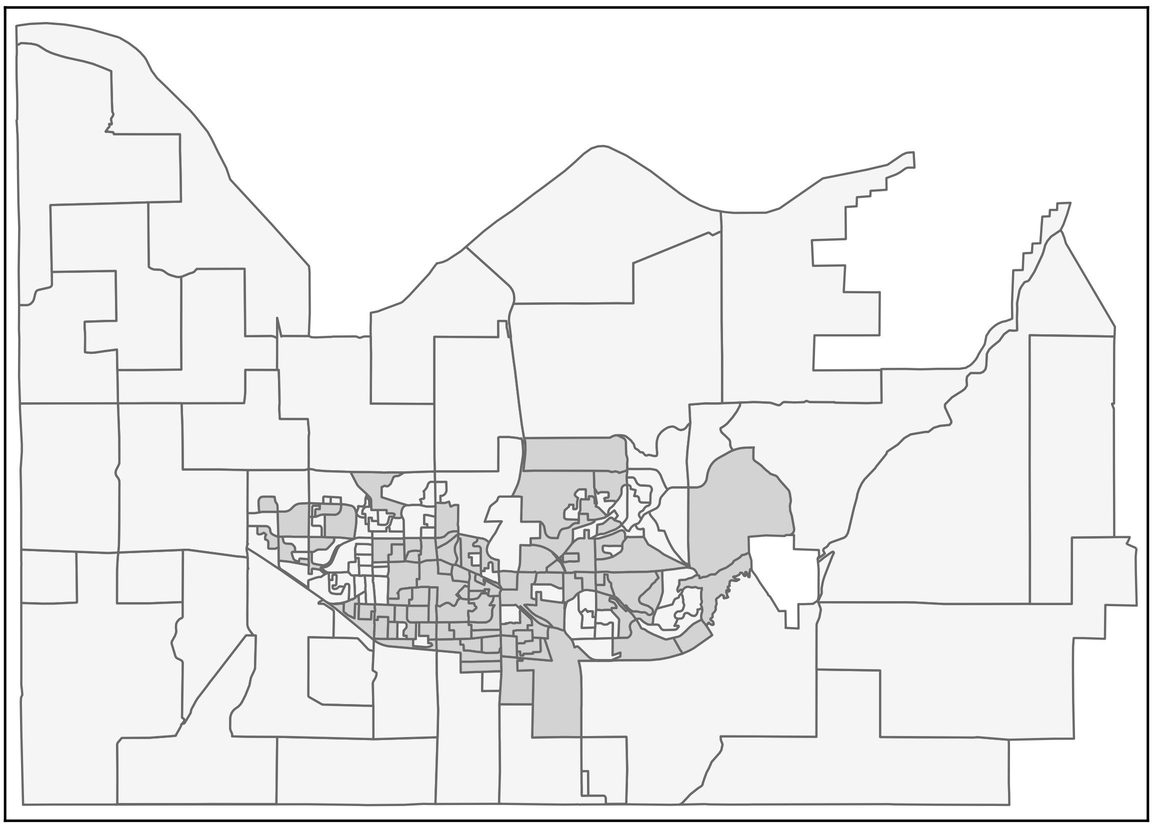


Figure 2.1: Abbotsford, British Columbia

Table 2.1: Dissemination areas in Abbotsford, each with a number of smart water meters and housing units for multi-family residences.

DAUID	# meters	# units	DAUID	# meters	# units
59090051	2	2	59090080	6	136
59090055	4	31	59090081	8	277
59090056	10	74	59090082	3	142
59090057	2	4	59090083	2	8
59090058	1	2	59090084	5	10
59090059	1	2	59090085	1	25
59090060	1	2	59090086	1	2
59090062	2	27	59090088	1	2
59090063	10	101	59090089	7	14
59090064	4	39	59090091	2	4
59090065	31	61	59090092	1	2
59090066	2	76	59090093	1	2
59090067	7	450	59090094	8	121
59090068	71	71	59090095	5	10
59090069	3	77	59090103	4	220
59090070	106	604	59090107	5	10
59090072	5	71	59090108	1	2
59090073	9	32	59090109	3	94
59090074	3	31	59090110	9	316
59090075	3	20	59090111	2	20
59090076	4	132	59090112	8	297
59090078	10	201	59090113	14	399

DAUID	**# meters**	**# units**	**DAUID**	**# meters**	**# units**
59090114	1	60	59090375	10	10
59090115	4	84	59090377	1	21
59090118	8	276	59090383	3	79
59090119	7	294	59090384	26	36
59090121	4	43	59090387	1	3
59090122	5	33	59090388	1	5
59090124	5	281	59090398	2	3
59090125	3	113	59090429	7	100
59090129	11	112	59090439	10	369
59090131	8	86	59090444	10	226
59090133	7	575	59090445	5	328
59090144	3	191	59090737	1	79
59090146	1	14	59090757	1	43
59090152	6	71	59090762	2	32
59090153	3	52	59090763	3	71
59090161	1	87	59090765	2	77
59090164	6	226	59090782	4	64
59090166	2	18	59090786	1	34
59090168	1	1	59090792	9	9
59090170	1	18	59090796	20	20
59090172	2	57	59090797	12	78
59090374	22	139			

2.1.2 Multi-Family Residences

Water consumers can be categorized into distinct sectors based on their water consumption patterns and usage. These customer sectors can be multi-family residences, single-family residences, commercial, and institutional. In this section, we define what a multi-family residence is and describe the different types of multi-family residences.

A multi-family residence is defined as a building which contains multiple dwelling units. Apartments and condominiums are examples of multi-family residences. In contrast, a single-family residence is a freestanding building which contains a single dwelling unit. Commercial consumers are those which provide products and services, and institutional consumers are schools, hospitals, and government [29]. Throughout this thesis, the focus will be on multi-family residences.

Multi-family residences come in a variety of building types. For the city of Abbotsford, the building type for the properties in our dataset was obtained from the BCAssessment property assessment tool[1]. Each property in our dataset can be categorized into a specific building type. Each building type is defined in Table 2.2 [41], and is depicted in Figure 2.2 and Figure 2.3.

Upon examining the building type categories in more detail, we find that each can be categorized into four broader categories: duplex, townhouse, multiple residence, and strata apartment. The reasoning behind categorizing into broader categories is to simplify the property features which are engineered in Section 3.7. This is because specific categories such as Fourplex only occur a few times throughout Abbotsford. Specific categories are binned into broad categories based on structural similarity. Table 2.3 summarizes how each building type is classified into a broader category. From this point forward, we only refer to these four building types in our discussion.

Each of the four building types has a specific distribution for each dissemination area. The spatial distributions for each building type in the city of Abbotsford are depicted and described in Section 2.1.4.

In general, it has been found that multi-family residences have lower per capita water usage compared to single-family residences [14], [11]. This is because multi-family residences typically have smaller yards and less landscaping compared to single-family residences, which reduces outdoor water usage. In [66], the results of an ordinary least squares regression analysis suggest that single-family residences with smaller lots and less landscaping lead to a decrease in water demand. These characteristics are common among multi-family residences.

[1]https://www.bcassessment.ca/

Duplex

Fourplex

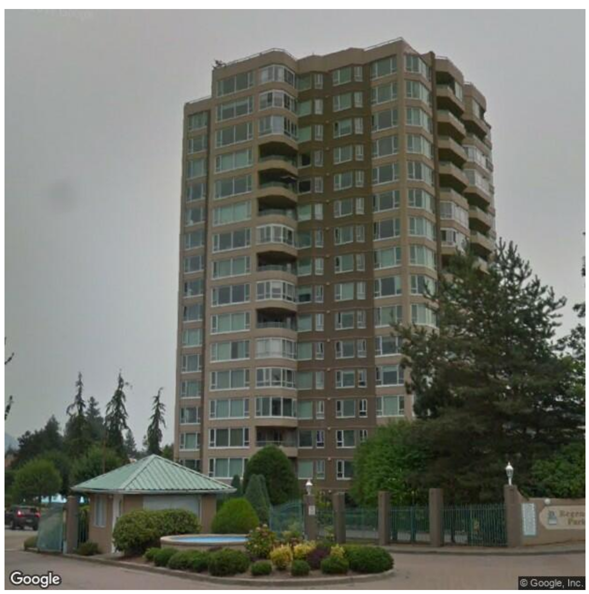
Townhouse

House

Figure 2.2: Multi-family residence building types in Abbotsford [33]

Multiple Residence

Strata Apartment - Frame



Strata Apartment - High-Rise

Figure 2.3: Multi-family residence building types in Abbotsford [33]

Table 2.2: Multi-family residence building type descriptions

Building type	**Description**
Duplex	An attached residence. Each unit is part of a group of two units that are adjoined by no more than two common walls. There are no other units above or below a duplex unit. Each duplex has an individual entry way.
Fourplex	An attached residence. Each unit is part of a group of four units that are adjoined by common walls.
Townhouse	An attached residence. Each unit is part of a group of two or more units that are adjoined by no more than two common walls. There are no other units above or below a townhouse unit. Each townhouse has an individual entry way.
House	A detached residence which is used to house more than one family. A house typically has only one main entry way.
Multiple Residence	A building containing multiple units with three stories or less.
Strata Apartment - Frame	A building containing multiple units with a wood frame structure.
Strata Apartment - High-Rise	A building containing multiple units with five stories or more. Built with a concrete structure.

Table 2.3: Multi-family residence building type categories

Broad category	**Specific categories**
Duplex	Duplex Fourplex House
Townhouse	Townhouse
Multiple Residence	Multiple Residence
Strata Apartment	Strata Apartment - Frame Strata Apartment - High-Rise

2.1.3 Smart Water Meters

Beginning in 2010, smart water meters were installed across Abbotsford with the purpose of reducing costs for meter reading, detecting leaks, and to collect data for water conservation programs [17]. Smart water meters record, store, and transmit water consumption data and typically record water consumption at an hourly rate [26]. In addition, water usage recorded by smart water meters can be processed and analyzed as time series data. The purpose and motivation behind the adoption of smart water meters is to provide water utilities with water consumption data to be used for several purposes, from forecasting water consumption to developing water conservation programs [59]. The article by Hauber and Idris [26] provides a more detailed description of how smart water meters work.

For multi-family residences in Abbotsford, smart water meters record hourly water usage at two different levels, depending on how the smart water meter is installed in a property:

1. Hourly water usage is recorded at the unit level. In this case, individual meters, one for each unit, records hourly water consumption for each unit of a multi-family residence building.

2. Hourly water usage is recorded at the building level. Here, a single master meter is used to record the total hourly water consumption for all units of a multi-family residence building.

Townhouses and duplexes are typically metered at the unit level, while multiple residences and strata apartments are typically metered at the building level.

Additionally, smart water meters in Abbotsford record water usage using two different options:

1. Water consumption is recorded for each hour of the day. The water consumption for a particular hour is the actual amount of water consumed in that hour.

2. Water consumption is recorded in bursts for contiguous hours of the day. The water usage recorded is the aggregated water usage starting from the last recording and ending with the current recording, for some period of hours.

Option (1) is typical of smart water meters which record at the unit level. Option (2) is typical of meters which record at the building level.

2.1.4 Datasets

In this section, we describe the datasets used throughout the thesis in detail. The following outlines the datasets used throughout the thesis:

1. **Water consumption data** recorded at an hourly rate and collected from 900 multi-family residences during the period starting on September 1, 2012 and ending on August 31, 2013 for the city of Abbotsford, British Columbia. Hourly water consumption data was recorded using smart water meters.

2. **Climate data** obtained for the years 2012 to 2013 for the city of Abbotsford. Contains attributes such as average daily temperature, rainfall, wind speed, and air pressure.

3. **Property assessment data** obtained from BCAssessment. Contains several property attributes recorded at the household scale for the year 2012. These property attributes include property value, lot size, number of bathrooms, number of bedrooms, and whether a household has a pool.

4. **Demographic data** obtained from the 2011 National Household Survey. The National Household Survey is conducted by Statistics Canada and is distributed to a sample of households across Canada, with the results aggregated at the dissemination area level. We used the 2011 version of the National Household Survey since this is the version closest to the years 2012 and 2013, as the survey is not conducted every year. The National Household Survey measures a variety of demographic and socioeconomic characteristics, such as income, employment rate, education, and family size.

Next, we discuss and examine our water consumption, climate, property, and demographic datasets in additional detail. We discuss the state of the data after conducting the feature engineering step as described in Section 3.7. In this analysis, we do not discuss or include data which has been case deleted from the data preprocessing step described in Section 3.6.

Water consumption data is recorded across the city of Abbotsford by smart water meters for the period between September 1, 2012 to August 31, 2013. Figure 2.4 shows the average daily water consumption per household for each dissemination area. This is calculated by taking the average of the daily water consumption for each household in the dissemination area for the period between September 1, 2012 to August 31, 2013. In the figure, it is clear that most dissemination areas have an average daily water consumption close to the city wide average of 592 litres per day per unit. There are however some dissemination areas which have a greater than average daily water consumption per household.

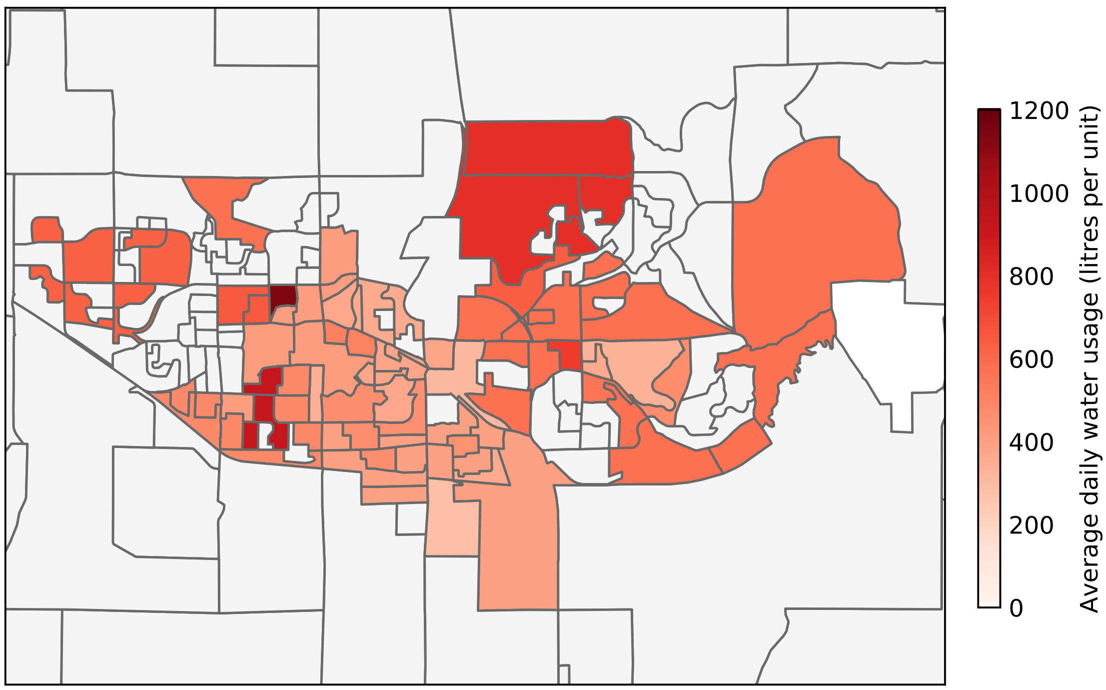


Figure 2.4: Spatial distribution of average daily water usage for multi-family residences across Abbotsford

Climate data occurs at the city level. Abbotsford has a temperate climate, with the area experiencing mild temperatures throughout most of the year and frequent rainfall, particularly in the winter months. Table 2.4 summarizes the climate data for the city of Abbotsford in the period between September 1, 2012 to August 31, 2013.

Table 2.4: Climate data summary for Abbotsford

Month	Average temperature (°C)	Total rainfall (mm)
September	15.54	5.50
October	10.44	306.30
November	6.59	240.20
December	3.25	184.20
January	2.53	152.20
February	5.05	103.40
March	7.07	206.40
April	9.12	157.60
May	13.70	101.40
June	16.24	85.00
July	19.38	1.60
August	19.10	57.00

Table 2.5: Distribution of multi-family building types across Abbotsford

Building type	Percentage across Abbotsford
Duplex	22.29%
Townhouse	57.96%
Multiple Residence	7.64%
Strata Apartment	12.10%

Next, we examine the spatial distributions of our property data in more detail. We examine these particular property features as they are investigated in further chapters. Figure 2.5 shows the average year that properties are built for each dissemination area. In general, the average year built for the dissemination areas tends to be clustered around the 1980s. There are few properties built before 1970 or after 1990 that are present in our dataset. The spatial distribution for the average number of bedrooms per household in a dissemination area is depicted in Figure 2.5. Most dissemination areas tend to average between 1.5 to 3.0 bedrooms per household. Figure 2.6 shows the spatial distribution of duplexes, townhouses, multiple residences, and strata apartments, respectively. The figures show which percentage of buildings within a dissemination area fall under a particular building type, in this case, either as a duplex, townhouse, multiple residence, or strata apartment. The figures make it clear that duplexes and townhouses are quite abundant across Abbotsford and make up a large proportion of the multi-family housing stock in Abbotsford. In contrast, multiple residences and strata apartments tend to be less abundant in Abbotsford. Table 2.5 shows the distribution of multi-family building types for the city of Abbotsford. Since duplexes and townhouses together, and multiple residences and strata apartments together tend to be quite similar, we bin these together to get two broader categories. The spatial distribution of duplexes or townhouses and multiple residences or strata apartments is depicted in Figure 2.7.

Demographic data is recorded at the dissemination area level. We examine the family size per household and the median household income calculated over the dissemination area level. These demographic features are investigated as they are discussed in later chapters. The average family size and median household income for each dissemination area are obtained from the 2011 National Household Survey. In the 2011 National Household Survey, data is collected at the household level and then aggregated at the dissemination area level. In Figure 2.8, it is clear that average family size tends to range between 2.0 to 4.5. Median household income has a greater variance, ranging from \$25,000 to \$95,000 per year.

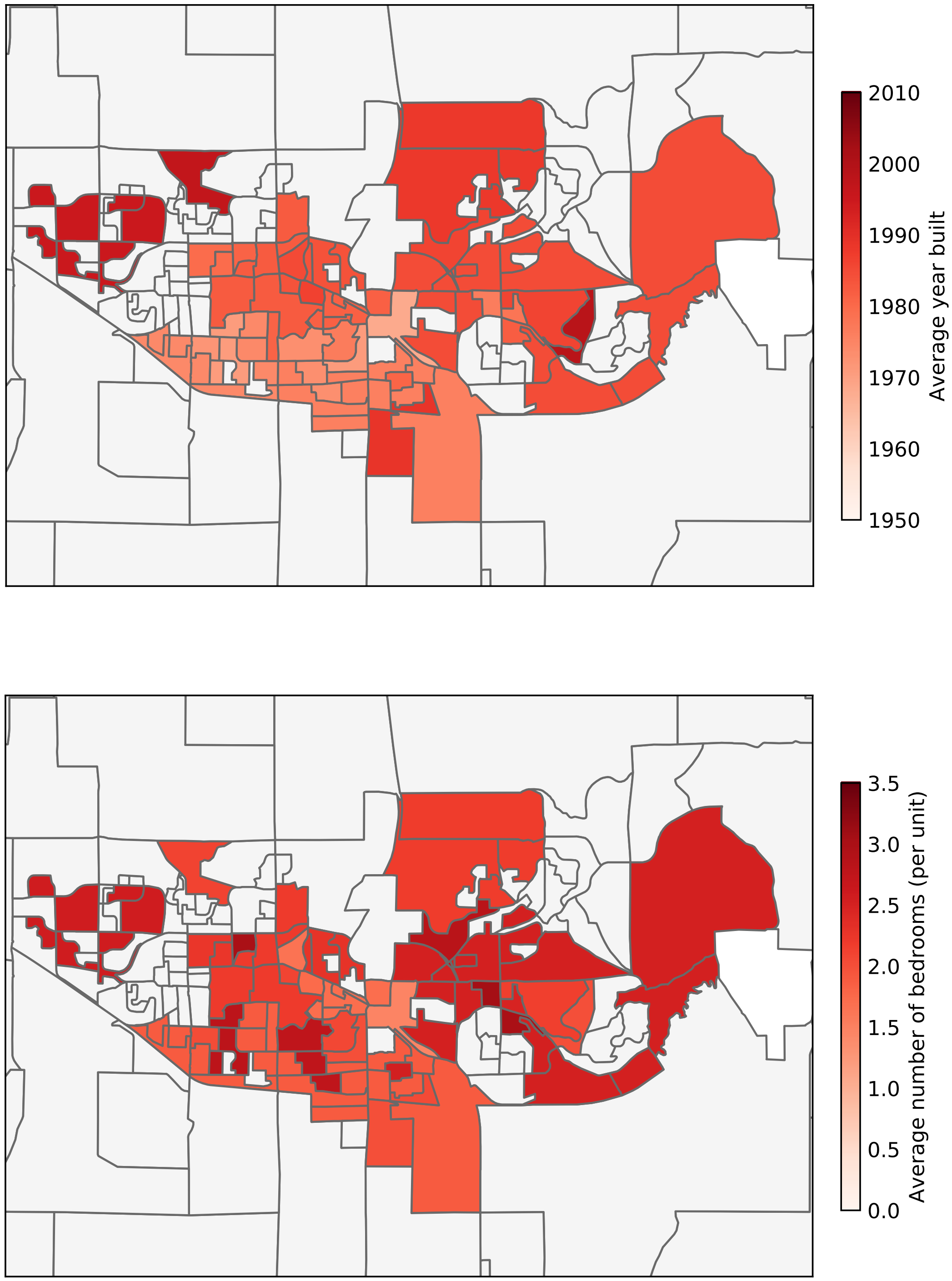


Figure 2.5: Spatial distributions of average year built and average number of bedrooms for multi-family residences across Abbotsford

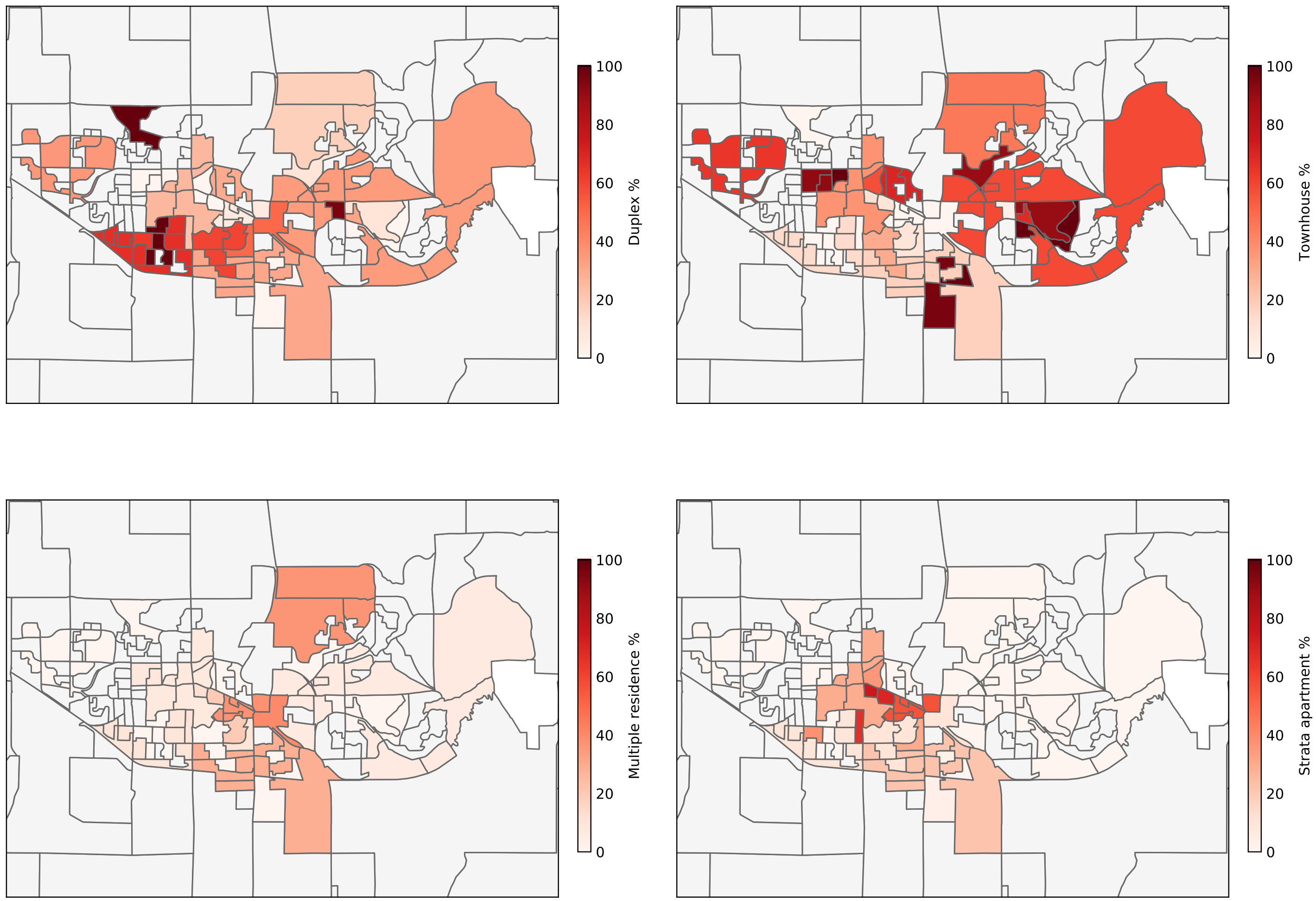
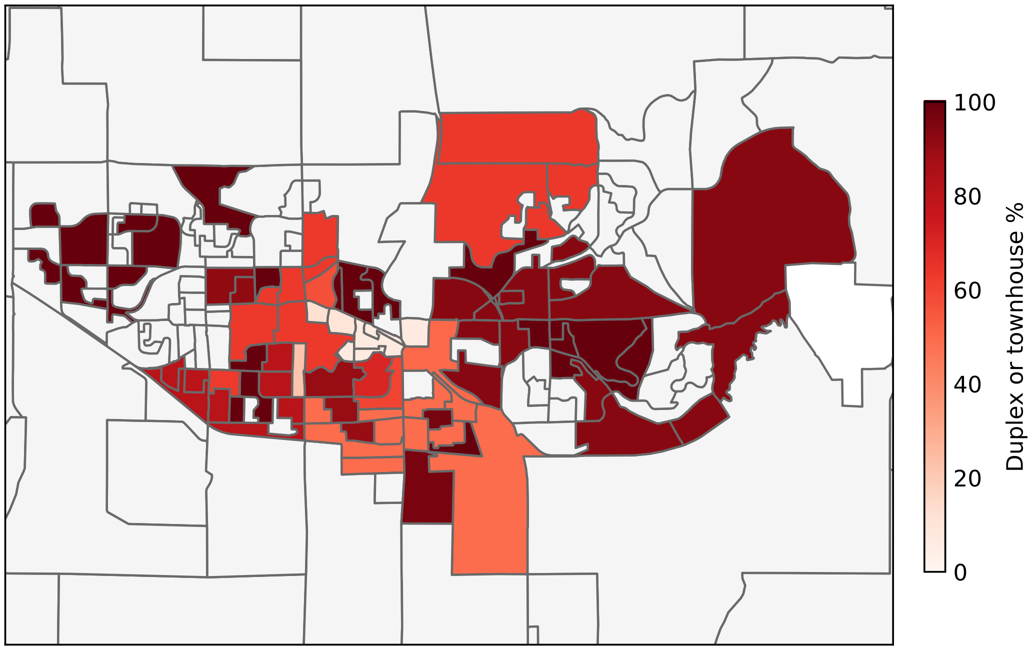


Figure 2.6: Spatial distributions of multi-family building types across Abbotsford



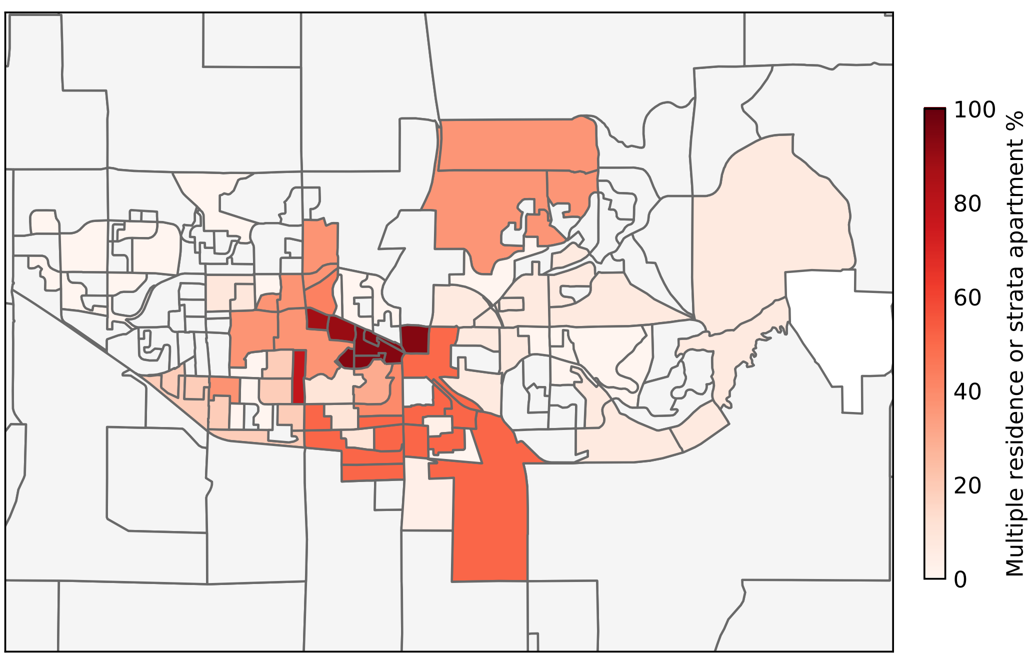


Figure 2.7: Spatial distributions of combined multi-family building types across Abbotsford

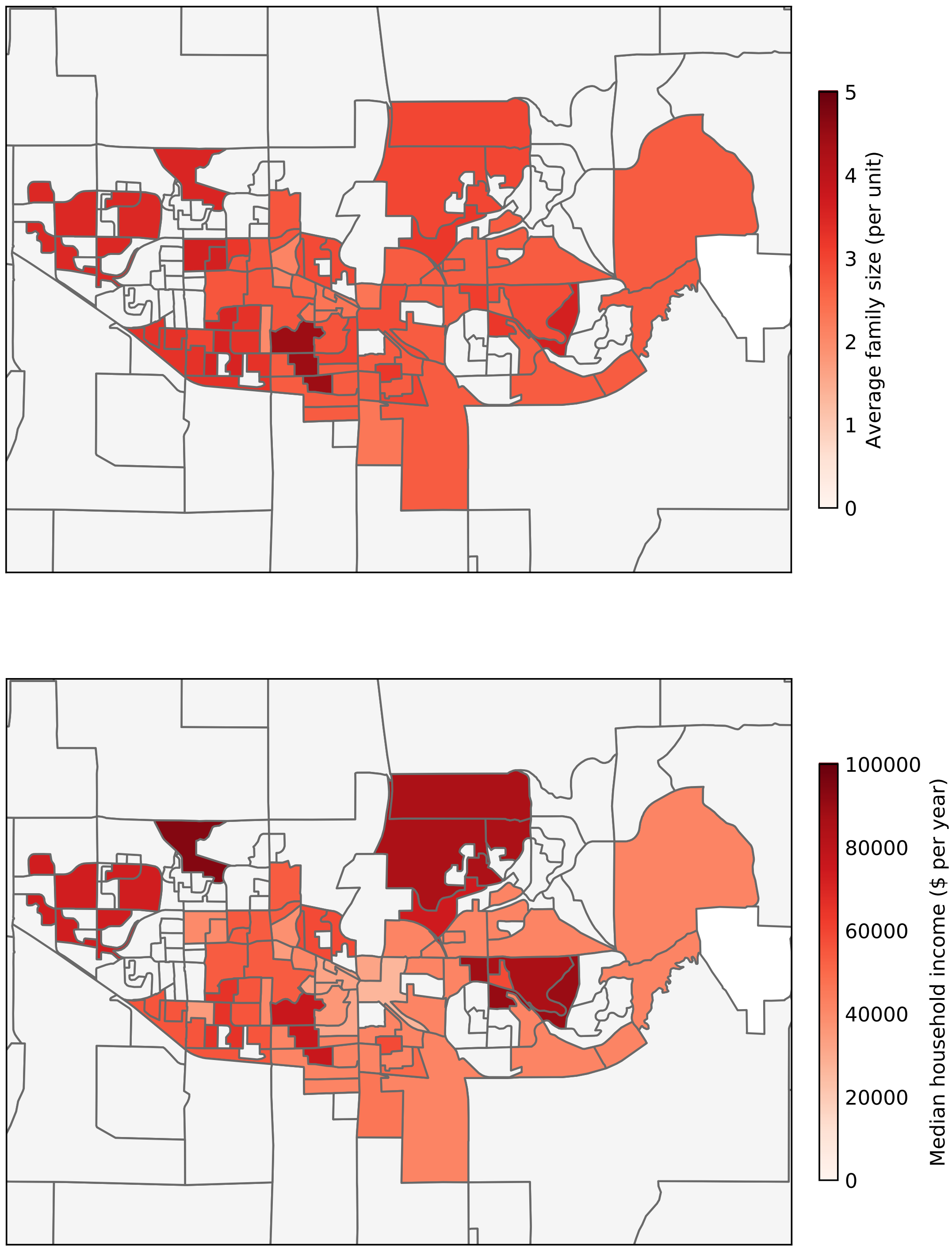


Figure 2.8: Spatial distributions of demographic features for multi-family residences across Abbotsford

2.2 Technical Background

In this section, we discuss the technical background of the thesis. The concept of time series forecasting formulated as a supervised learning task is discussed. We then provide an introduction to the machine learning and deep learning models of interest used throughout the thesis. Readers who are familiar with these methods may skip this section and refer to it when required.

2.2.1 Time Series Forecasting

Water consumption recorded by smart water meters can be thought of as a time series. In a time series, a value is observed at equally spaced points in time. These values form a sequence of observations which have a temporal ordering. In time series forecasting, the objective is to predict future values of a time series using previously observed values, called lagged values, and potentially other exogenous variables. In this thesis, we use the common machine learning terminology and refer to lagged values and exogenous variables as features. A set of features corresponds to an output value which is called the target.

Time series forecasting can be thought of as a supervised learning task. In supervised learning, an input matrix $\boldsymbol{X}$ and a corresponding output matrix $\boldsymbol{Y}$ is set as the training set:

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & x_{2,3} & \dots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{m,1} & x_{m,2} & x_{m,3} & \dots & x_{m,n} \end{bmatrix}$$

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Where the i^{th} row in $\boldsymbol{X}$ is the input vector $\boldsymbol{x}_i$ and is the set of feature values which correspond to the output vector $\boldsymbol{y}_i$ of $\boldsymbol{Y}$ which is the target value.

The output matrix $\boldsymbol{Y}$ can be generated by the unknown function:

$$\boldsymbol{Y} = f(\boldsymbol{X})$$

The goal of supervised learning is to find a function h, called the hypothesis, which approximates the function f:

$$\hat{\boldsymbol{Y}} = h(\boldsymbol{X})$$

The hypothesis uses a set of features as an input to predict the corresponding output. Learning the hypothesis consists of a search through a space of possible hypothesis. Ideally, the hypothesis must generalize well to unseen data. A hypothesis is said to generalize well if it is able to predict new instances of a problem adequately and does not differ too greatly from the performance evaluated on the training set, otherwise, the hypothesis is said to overfit, which occurs when the hypothesis fits too closely to the training data. Unseen instances are referred to as the test set and is the dataset from which model performance is evaluated.

In this thesis, model performance is evaluated using a technique called cross-validation; specifically, k-fold cross-validation. During each iteration of k-fold cross-validation, the data is partitioned into k folds, where $k - 1$ folds are set as the training set and the remaining fold is set as the test set. Performance is evaluated on the test set and is retained. This process is iterated until all folds have been used as a test set. Typically, the performance of each test set is averaged out to get an overall performance value. It should be noted that in Chapter 3 and Chapter 4 we only split our data into a train and test set. Splitting our data into a train, validation, and test set would likely result in underfitting due to the small size of our dataset. Underfitting occurs when the hypothesis does not capture the underlying data well and does not fit adequately to the training data.

Bontempi et al. [7] describes time series forecasting in detail. A description of supervised learning is provided in [55]. And [20] outlines model evaluation in depth.

2.2.2 Support Vector Machines

Support vector machines are a popular and powerful machine learning model used to perform linear and nonlinear classification or regression. In this section, we outline the regression case, commonly referred to as support vector regression. It is formulated in detail in [58] and described well in [55].

The objective of support vector regression is to find a function h such that its fit includes as many training instances inside a margin while limiting the instances that fall outside the margin. The hyperparameter ε controls the width of the margin, as shown in Figure 2.9.

If the relationship between the input and output variables, $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, is approximately linear, the model can be represented as:

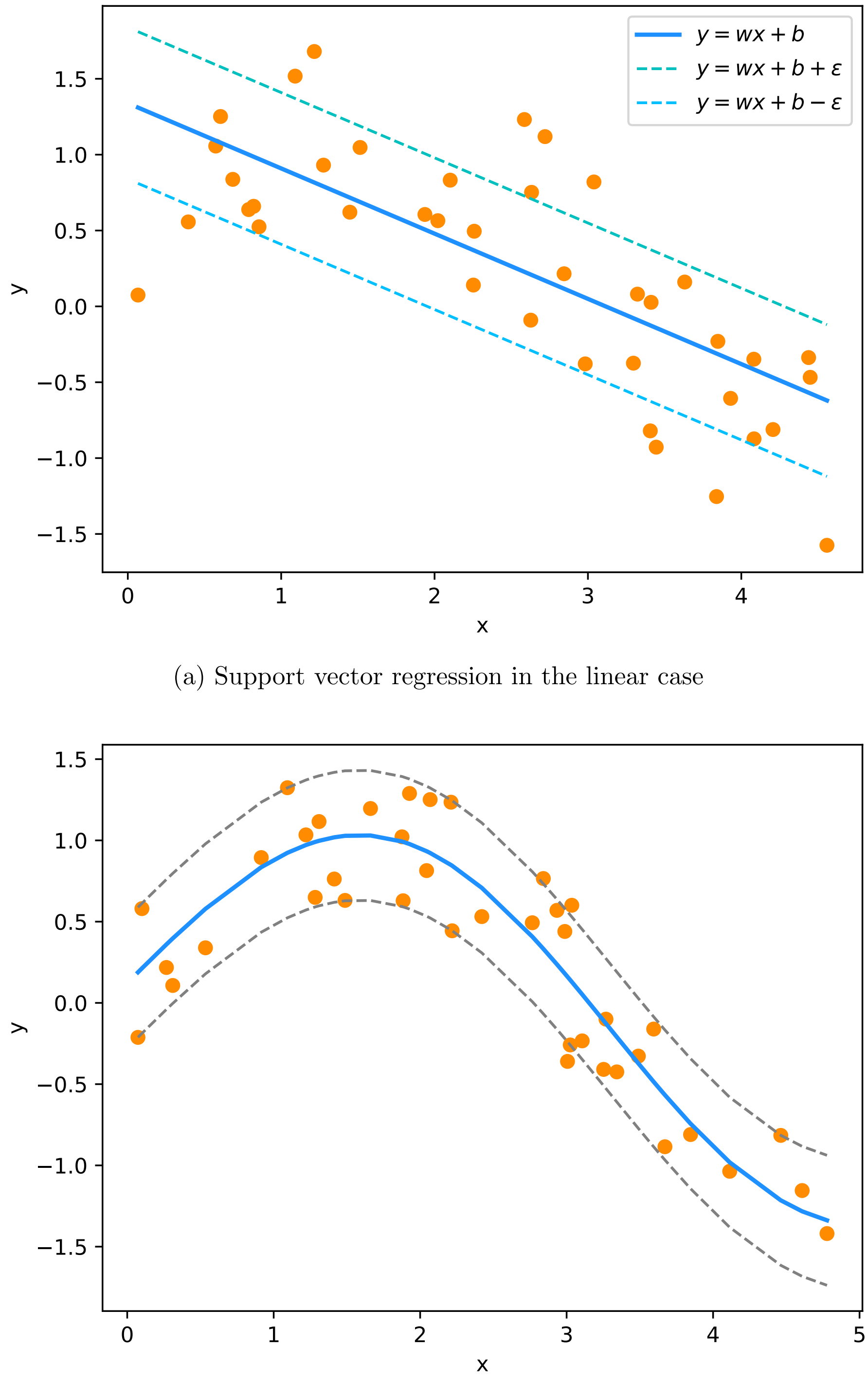
$$h(w_1, \ldots, w_n, b) = y = \boldsymbol{w} \cdot \boldsymbol{x} + b$$

The training objective can be expressed as a convex optimization problem, where the Euclidean norm is minimized subject to the margin constraints:

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{subject to} \begin{cases} y_i - (\boldsymbol{w} \cdot \boldsymbol{x_i}) - b & \leq \quad \varepsilon \\ (\boldsymbol{w} \cdot \boldsymbol{x_i}) + b - y_i & \leq \quad \varepsilon \end{cases}$$

For the nonlinear case, a nonlinear function of the data is learned by using a technique called the kernel trick. Here, a kernel function is used to project the data into a higher dimensional feature space, where a linear function is fitted. This linear function in the high dimensional space corresponds to a nonlinear function in the original feature space, as depicted in Figure 2.9. In this thesis, we investigate the use of both linear and nonlinear kernels.



(a) Support vector regression in the linear case

(b) Support vector regression in the nonlinear case

Figure 2.9: Linear and nonlinear cases of support vector regression

2.2.3 Artificial Neural Networks

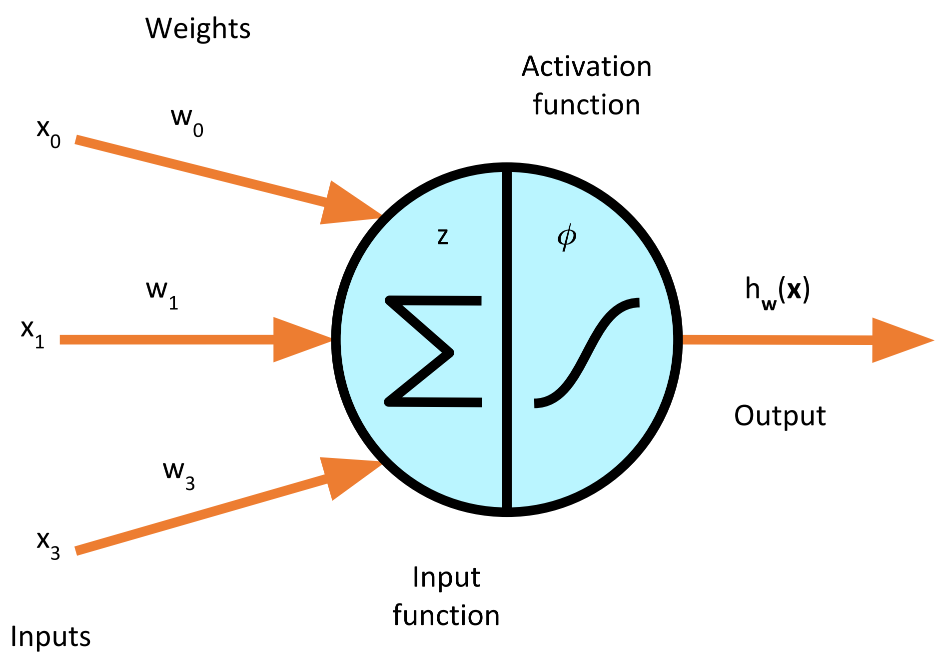


Figure 2.10: An artificial neural network node

Artificial neural networks (ANN) are a powerful and scalable model inspired by biological neurons in the brain. An artificial neural network is composed of nodes (neurons) connected by directed edges. Figure 2.10 depicts a single node of an ANN. Each edge is assigned a numeric weight which is an indication of the strength and sign of the connection. Each node computes a weighted sum of its inputs, where $\boldsymbol{w}$ is the weight vector and $\boldsymbol{x}$ is the input vector:

$$z = \boldsymbol{w}^T \cdot \boldsymbol{x}$$

The output of the node is obtained by applying an activation function to the weighted sum of the inputs:

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \phi(\boldsymbol{w}^T \cdot \boldsymbol{x})$$

Common activation functions used in practice include logistic, rectified linear (relu), and hyperbolic tangent (tanh) and are defined respectively:

$$logistic(z) = \frac{1}{1 + e^{-z}}$$

$$relu(z) = \begin{cases} 0 & \text{for} \quad z < 0 \\ z & \text{for} \quad z \geq 0 \end{cases}$$

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

To be able to solve complex problems, nodes are connected to form a network. A feed-forward network, as depicted in Figure 2.11, forms connections in only one direction and is a directed acyclic graph. Feed-forward networks are arranged in layers and contain an input layer, one or more hidden layers, and an output layer.

To train an ANN, the backpropagation training algorithm is used. The following outlines the backpropagation algorithm: During the forward pass, a training instance is inputted through the network and the output of every node in the network is calculated. Then, the error of the output is computed, which is the difference between the actual output and the predicted output of the network. Next, a reverse pass is run, which looks at each node in the last hidden layer and computes the amount that each node contributes to the error of the output layer. It then determines how much of these error contributions come from the nodes of the preceding hidden layer. This process continues until the input layer is reached. The reverse pass computes the error gradient of all the weights in the network by propagating the error gradient backwards. The final step of the backpropagation algorithm is a gradient descent on the weights of the network using the error gradients computed previously. The gradient descent algorithm tweaks the weights of the network to reduce the error. Overall, the training objective is to find the weights of the network which minimize a cost function. Goodfellow et al. [25] provides an introduction to artificial neural networks and describes the backpropagation algorithm in detail.

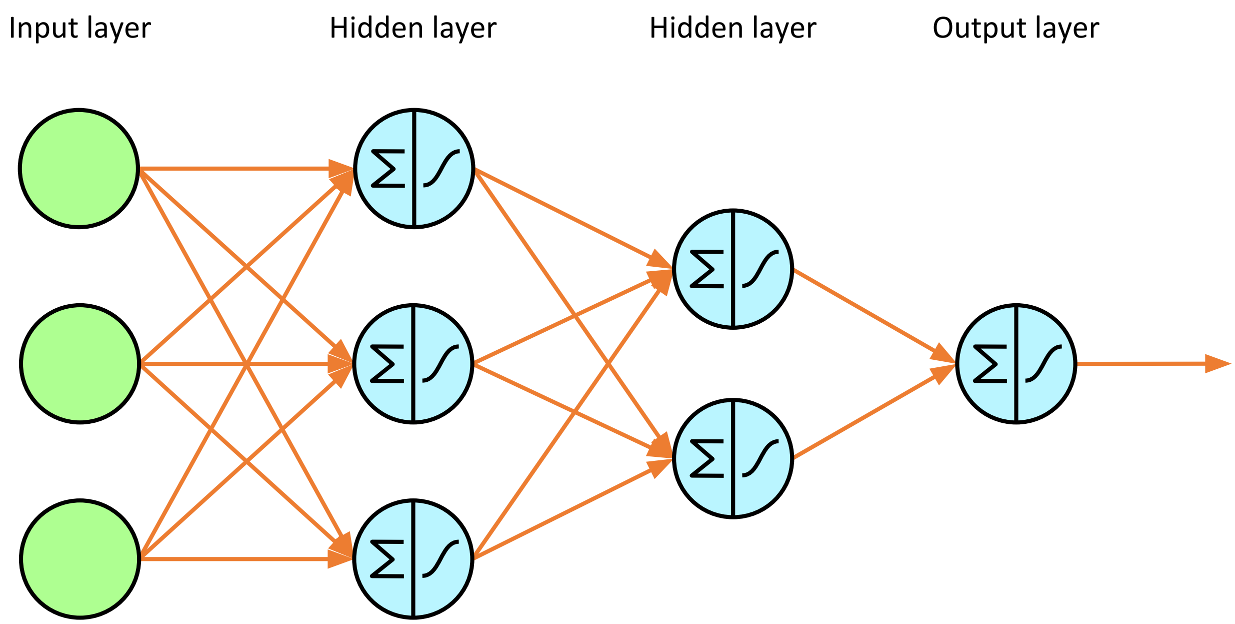


Figure 2.11: A feed-forward artificial neural network

2.2.4 Recurrent Neural Networks

Recurrent neural networks (RNN) are a type of model that are particularly useful when inputs are in the form of sequences, such as with time series data. In contrast with feed-forward neural networks, where activations flow in only a single direction, a recurrent neural network has outputs which are directed back to its inputs. Figure 2.12 depicts a single node of a recurrent neural network. Here, the initial input is passed through the node where an output is produced and is also directed back as an input. In addition, the recurrent node can be unrolled through time. When unrolling a network through time, the network is plot against a time axis. At each time step t, the input vector $\boldsymbol{x}_{(t)}$ and the output from the previous time step $\boldsymbol{y}_{(t-1)}$ is received by the node. We note that the total number of time steps refers to the number of steps that the output is redirected back as an input in the network.

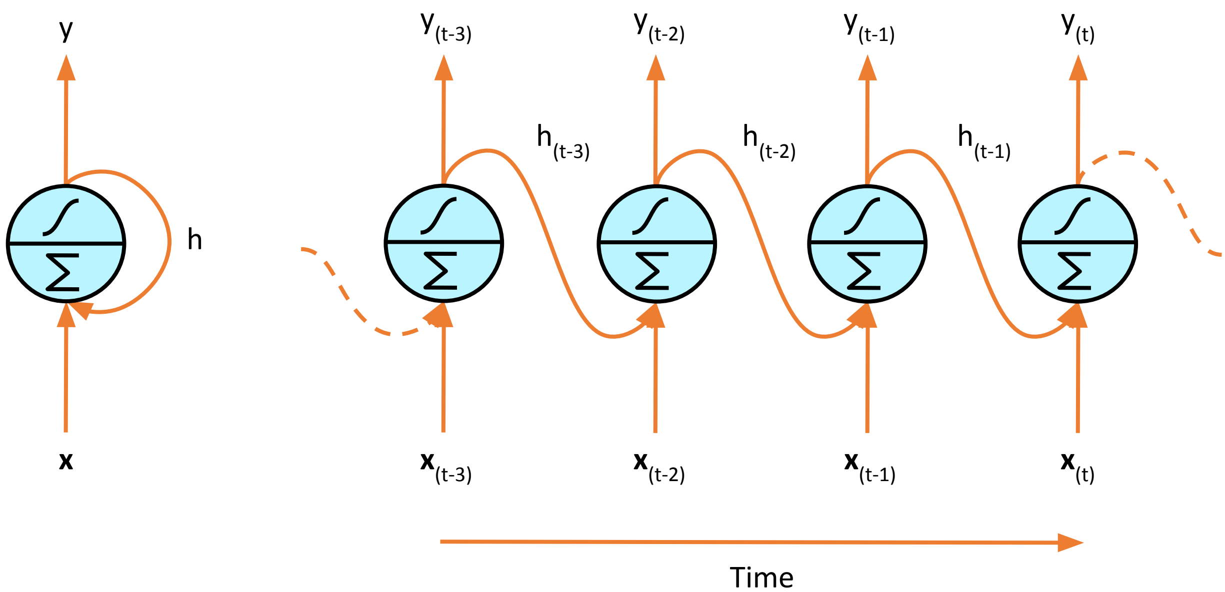


Figure 2.12: A recurrent neural network node and a node unrolled through time

Recurrent nodes can be arranged in a layer as shown in Figure 2.13. Each node in the recurrent layer receives the input vector $\boldsymbol{x}_{(t)}$ and the output from the previous time step $\boldsymbol{y}_{(t-1)}$. Each node also has two sets of weights, corresponding to the input $\boldsymbol{x}_{(t)}$ and the output of the previous time step $\boldsymbol{y}_{(t-1)}$ and are denoted as the vectors $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ respectively. When the entire layer is considered, the $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ vectors for each node can be placed in the matrices $\boldsymbol{W}_x$ and $\boldsymbol{W}_y$. The output of a recurrent layer for a single training instance can be computed as follows, where $\boldsymbol{b}$ is the bias vector and ϕ is the activation function:

$$\boldsymbol{y}_{(t)} = \phi({\boldsymbol{W}_x}^T \cdot \boldsymbol{x}_{(t)} + {\boldsymbol{W}_y}^T \cdot \boldsymbol{y}_{(t-1)} + \boldsymbol{b})$$

We note that a batch of inputs can also be inputted into a recurrent network. This is denoted by $\boldsymbol{X}_{(t)}$ and is an m x n_{input} input matrix containing m instances with n_{input} features. The output will also take the form of a matrix, denoted by $\boldsymbol{Y}_{(t)}$ and is an m x n_{node} matrix of size m instances and n_{node} nodes. It contains the outputs for each instance of the batch at time step t.

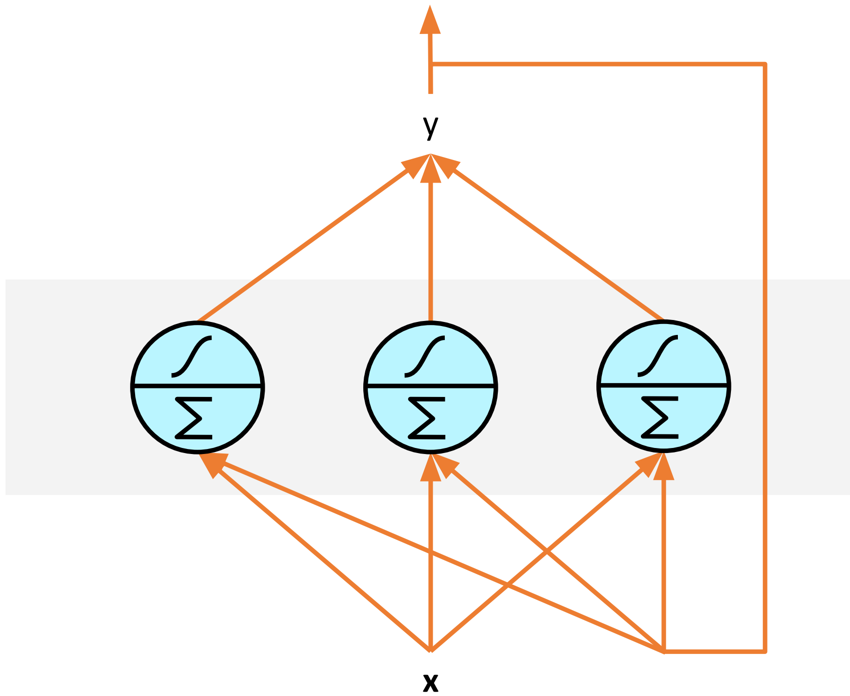


Figure 2.13: A layer of recurrent nodes

A recurrent node has a form of memory since its output at time step t is a function of the inputs from previous time steps. A memory cell is a component of a neural network which can store a state across time steps. A cell's state at time step t is denoted as $h_{(t)}$. It is a function of the input at the current time step and the state of the previous time step, $h_{(t)} = f(h_{(t-1)}, \boldsymbol{x}_{(t)})$. The output of the cell, denoted as $\boldsymbol{y}_{(t)}$, is a function of the input at the current time step and the state of the previous time step.

Training recurrent neural networks is similar to training feed-forward neural networks. The backpropagation algorithm is applied to a recurrent neural network which has been unrolled through time. This is defined as conducting a backpropagation through time. Backpropagation through time is as follows: A forward pass is run through the unrolled network. The cost is evaluated using the cost function $C(\boldsymbol{Y}_{(t_{min})}, \boldsymbol{Y}_{(t_{min}+1)}, \ldots, \boldsymbol{Y}_{(t_{max})})$, where t_{min} is the first time step and t_{max} is the last time step. Next, in the reverse pass, the gradients are calculated from the cost function and are propagated backwards through the unrolled network. Finally, the parameters of the model are updated using the gradients computed from the backpropagation.

When applied to time series data, as is done Chapter 5, our recurrent neural network takes as input a sequence of consecutive values from a time series. We refer to this fixed sequence as a window. The output of the network is also a window of fixed size but shifted one time step into the future. For the output, we are only concerned with the last value of the window, as this is the predicted next value of the input sequence. We frequently referred to [20] to obtain detailed information on recurrent neural networks.

2.2.5 Long Short-Term Memory

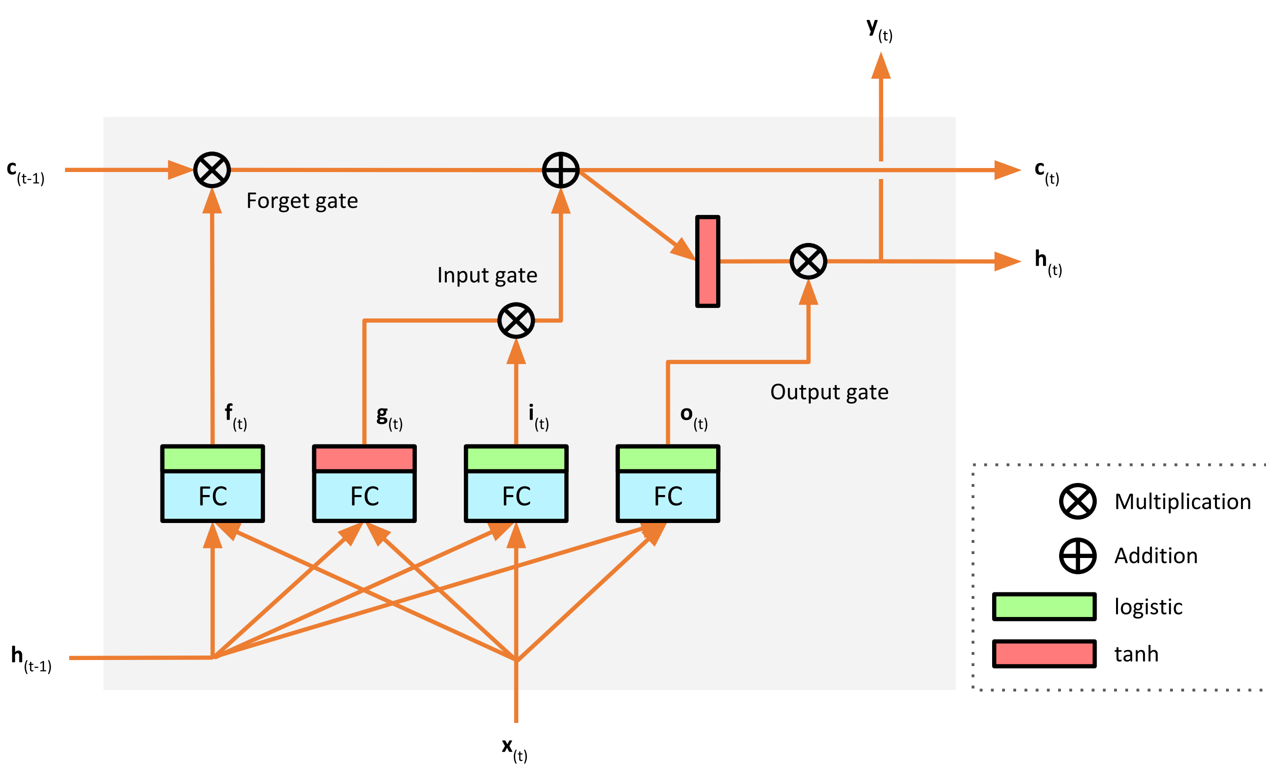


Figure 2.14: LSTM cell architecture

Long short-term memory (LSTM) units were introduced to address the shortcomings of RNNs using basic memory units. In this section, we refer to a recurrent neural network using LSTM units as an LSTM network. With RNNs, training over many time steps on long sequences of data results in slow training times, as the backpropagation through time is run on a deep unrolled network and is subject to vanishing and exploding gradients. To address this problem, the unrolled network can be reduced by unrolling over a limited number of time steps and running a truncated backpropagation through time. However, with this approach, long-term dependencies cannot be learned. Another disadvantage of RNNs is that over time, memory from earlier inputs gradually fades away. During each time step, some information is dropped. In contrast, LSTM units contain long-term memory which are able to retain long-term dependencies in data. In addition, LSTM networks tend to perform better and converge faster in practice.

Figure 2.14 displays a basic LSTM cell architecture. The state of the LSTM cell includes two vectors, the long-term state $\boldsymbol{c}_{(t)}$ and the short-term state $\boldsymbol{h}_{(t)}$. The LSTM cell can read and store memories from the long-term state and determine which memories should be dropped. When the long-term state $\boldsymbol{c}_{(t-1)}$ traverses the cell from left to right, it first goes through a forget gate where some information is dropped. Next, it goes through an addition gate where information is added based on what was selected by an input gate. The information is then sent out as $\boldsymbol{c}_{(t)}$ and is also passed through a tanh function. After the tanh function, the long-term state is passed through an output gate which produces the short-term state $\boldsymbol{h}_{(t)}$. So, at each time step, memory is dropped, added, and a long-term and short-term state is outputted by the cell.

Next, we describe how short-term memory and the input is handled. The input vector $\boldsymbol{x}_{(t)}$ and the short-term state from the previous time step $\boldsymbol{h}_{(t-1)}$ are inputted into four different fully connected layers. Three of these layers serve as gate controllers. Gate controllers use a logistic activation function which output values between 0 and 1. The outputs of these gate controllers pass through an element-wise multiplication operation, so a gate controller which outputs 0 will close the gate, and a gate controller which outputs 1 will open the gate. The four fully connected layers are as follows:

- The first fully connected layer analyzes $\boldsymbol{x}_{(t)}$ and $\boldsymbol{h}_{(t-1)}$ and outputs $\boldsymbol{g}_{(t)}$. $\boldsymbol{g}_{(t)}$ is partially stored in the long-term state.
- The second fully connected layer is a forget gate and outputs $\boldsymbol{f}_{(t)}$. It determines which part of the long-term state should be dropped.
- The third fully connected layer is an input gate and outputs $\boldsymbol{i}_{(t)}$. It determines which part of $\boldsymbol{g}_{(t)}$ should be added to the long-term state.

- The fourth fully connected layer is an output gate and outputs $\boldsymbol{o}_{(t)}$. It determines the two outputs of the cell, the short-term state $\boldsymbol{h}_{(t)}$ and the output $\boldsymbol{y}_{(t)}$.

To summarize, an LSTM cell can determine which inputs are important and store it in the long-term state (which is the task of the input gate), can determine which part of the long-term state should be preserved (the task of the forget gate), and can extract important parts of the long and short-term memory. This is why LSTMs have been successful at capturing long-term dependencies in data.

The following equations summarize how the long-term state, short-term state, and the output of the cell are calculated at a single time step for one training instance:

$$
\begin{aligned}
\boldsymbol{i}_{(t)} &= \sigma(\boldsymbol{W}_{xi}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hi}{}^{T} \cdot \boldsymbol{h}_{(t-1)} + \boldsymbol{b}_{i}) \\
\boldsymbol{f}_{(t)} &= \sigma(\boldsymbol{W}_{xf}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hf}{}^{T} \cdot \boldsymbol{h}_{(t-1)} + \boldsymbol{b}_{f}) \\
\boldsymbol{o}_{(t)} &= \sigma(\boldsymbol{W}_{xo}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{ho}{}^{T} \cdot \boldsymbol{h}_{(t-1)} + \boldsymbol{b}_{o}) \\
\boldsymbol{g}_{(t)} &= tanh(\boldsymbol{W}_{xg}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hg}{}^{T} \cdot \boldsymbol{h}_{(t-1)} + \boldsymbol{b}_{g}) \\
\boldsymbol{c}_{(t)} &= \boldsymbol{f}_{(t)} \otimes \boldsymbol{c}_{(t-1)} + \boldsymbol{i}_{(t)} \otimes \boldsymbol{g}_{(t)} \\
\boldsymbol{y}_{(t)} &= \boldsymbol{h}_{(t)} = \boldsymbol{o}_{(t)} \otimes tanh(\boldsymbol{c}_{(t)})
\end{aligned}
$$

Where,

- $\boldsymbol{W}_{xi}$, $\boldsymbol{W}_{xf}$, $\boldsymbol{W}_{xo}$, $\boldsymbol{W}_{xg}$ are the weight matrices for each of the four layers which correspond to the input vector $\boldsymbol{x}_{(t)}$.
- $\boldsymbol{W}_{hi}$, $\boldsymbol{W}_{hf}$, $\boldsymbol{W}_{ho}$, $\boldsymbol{W}_{hg}$ are the weight matrices for each of the four layers which correspond to the short-term state vector $\boldsymbol{h}_{(t-1)}$.
- $\boldsymbol{b}_{i}$, $\boldsymbol{b}_{f}$, $\boldsymbol{b}_{o}$, $\boldsymbol{b}_{g}$ are the bias vectors for each of the four layers.

2.2.6 Gated Recurrent Unit

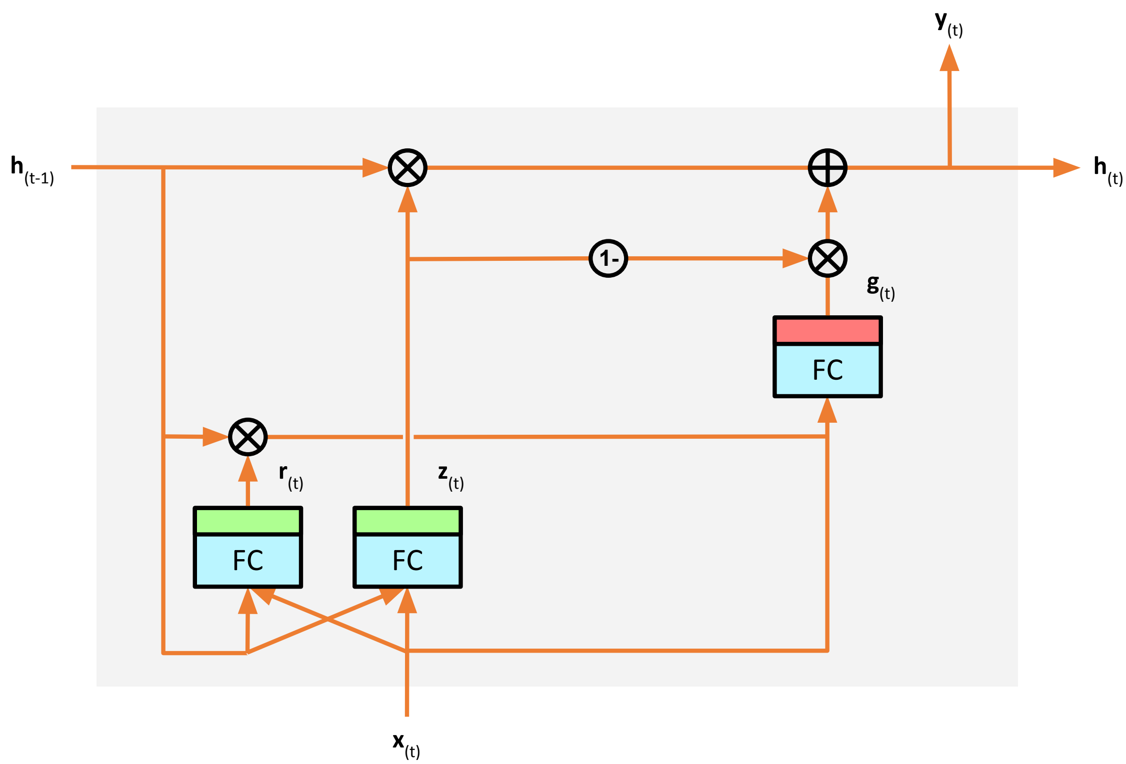


Figure 2.15: GRU cell architecture

The gated recurrent unit (GRU) is a simplified form of an LSTM unit and has been shown to work well in practice. We refer to a recurrent neural network using GRU units as a GRU network. Figure 2.15 depicts the GRU cell architecture. There are several simplifications made to the GRU architecture. For example, the long-term and short-term states have been merged into a single state vector $\boldsymbol{h}_{(t)}$. A single gate controller serves as both the forget gate and the input gate and outputs $\boldsymbol{z}_{(t)}$. If the gate controller outputs a 0, the input gate is opened and the forget gate is closed. If the gate controller outputs a 1, the input gate is closed and the forget gate is opened. In addition, the output gate is omitted and the state vector which contains both long and short-term states, $\boldsymbol{h}_{(t)}$, is output by the cell. A new gate controller, which outputs $\boldsymbol{r}_{(t)}$, is added which controls which part of $\boldsymbol{h}_{(t-1)}$ is inputted into the fully connected layer which outputs $\boldsymbol{g}_{(t)}$.

The following equations outline how the state is calculated during a single time step for a training instance:

$$
\begin{aligned}
\boldsymbol{z}_{(t)} &= \sigma(\boldsymbol{W}_{xz}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hz}{}^{T} \cdot \boldsymbol{h}_{(t-1)} + \boldsymbol{b}_{z}) \\
\boldsymbol{r}_{(t)} &= \sigma(\boldsymbol{W}_{xr}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hr}{}^{T} \cdot \boldsymbol{h}_{(t-1)} + \boldsymbol{b}_{r}) \\
\boldsymbol{g}_{(t)} &= tanh(\boldsymbol{W}_{xg}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hg}{}^{T} \cdot (\boldsymbol{r}_{(t)} \otimes \boldsymbol{h}_{(t-1)}) + \boldsymbol{b}_{g}) \\
\boldsymbol{g}_{(t)} &= tanh(\boldsymbol{W}_{xg}{}^{T} \cdot \boldsymbol{x}_{(t)} + \boldsymbol{W}_{hg}{}^{T} \cdot \boldsymbol{h}_{(t-1)} + \boldsymbol{b}_{g}) \\
\boldsymbol{h}_{(t)} &= \boldsymbol{z}_{(t)} \otimes \boldsymbol{h}_{(t-1)} + (1 - \boldsymbol{z}_{(t)}) \otimes \boldsymbol{g}_{(t)}
\end{aligned}
$$

2.2.7 Convolutional Neural Networks

We explore the use of convolutional neural networks (CNN) for the purposes of time series prediction in this thesis. Typically, CNNs are commonly used for image recognition tasks where two-dimensional convolutions are applied to image data. For image recognition tasks, CNNs are the preferred architecture over fully connected networks, as fully connected networks trained on image data typically result in a large number of trainable parameters (weights and biases) which presents a computational challenge. In CNNs, the layers are partially connected, reducing the number of trainable parameters. Here, we describe the one-dimensional case which can be applied to one-dimensional data such as a sequential time series. The 1D case of CNNs can be derived from 2D CNNs as a special case. We refer to [47] when looking at the use of 1D CNNs to conduct time series forecasting. The following describes the main ideas behind convolutional neural networks:

- CNNs are composed of convolutional layers and pooling layers and are stacked against each other. Dense fully connected layers can also be present at higher layers.
- In a convolutional layer, features are extracted from data. This is done by moving a filter over an input matrix (such as a 2D image, a 1D time series sequence, or an output matrix from the preceding layer) and is called a convolution. A filter is simply a matrix of weights, where each weight corresponds to an edge weight of a neuron.
- A feature map is formed by convolving over the input matrix. It is this convolution which generates a feature map of learned features. A convolution layer may contain multiple filters which are convolved over an input matrix to detect multiple features.

- Feature maps are downsampled to reduce the number of trainable parameters in the dense layers. In a pooling layer, downsampling is conducted by convolving an arithmetic function, such as max or average, on a feature map. The result is a downsampled feature map. Downsampling reduces overfitting and decreases computational and memory usage.

In Figure 2.16 and Figure 2.17, we depict a 1D convolution and pooling on a time series sequence. A 1D convolution can be formulated as a special case of a 2D convolution. Here, a time series sequence is similar to an image input with a height of a single pixel. Instead of using a 2D filter, a 1D filter is used and is only convolved in a horizontal direction. In the 1D convolution example, we use a time series sequence of size 1x10 as an input and a 1D filter of size 1x3. Using a stride of one unit, the first convolution is calculated as $(1 \times 1) + (2 \times 0) + (1 \times 1) = 2$. Here, the weight of each neuron in the filter is multiplied by the corresponding value in the time series sequence to produce a weighted sum. Then, an optional bias is added to the weighted sum and is passed through an activation function. Here, a rectified linear activation function (relu) is used. Since all the weighted sums are positive, the relu activation function behaves like an identity function. The filter is moved over the time series until it hits the end of the sequence and a feature map of size 1x8 is produced. Next, the feature map produced by the convolutional layer can be downsampled by a pooling layer. In this example, a max pooling layer of size 1x2 is used and downsamples the feature map to size 1x4 using a horizontal stride of two units.

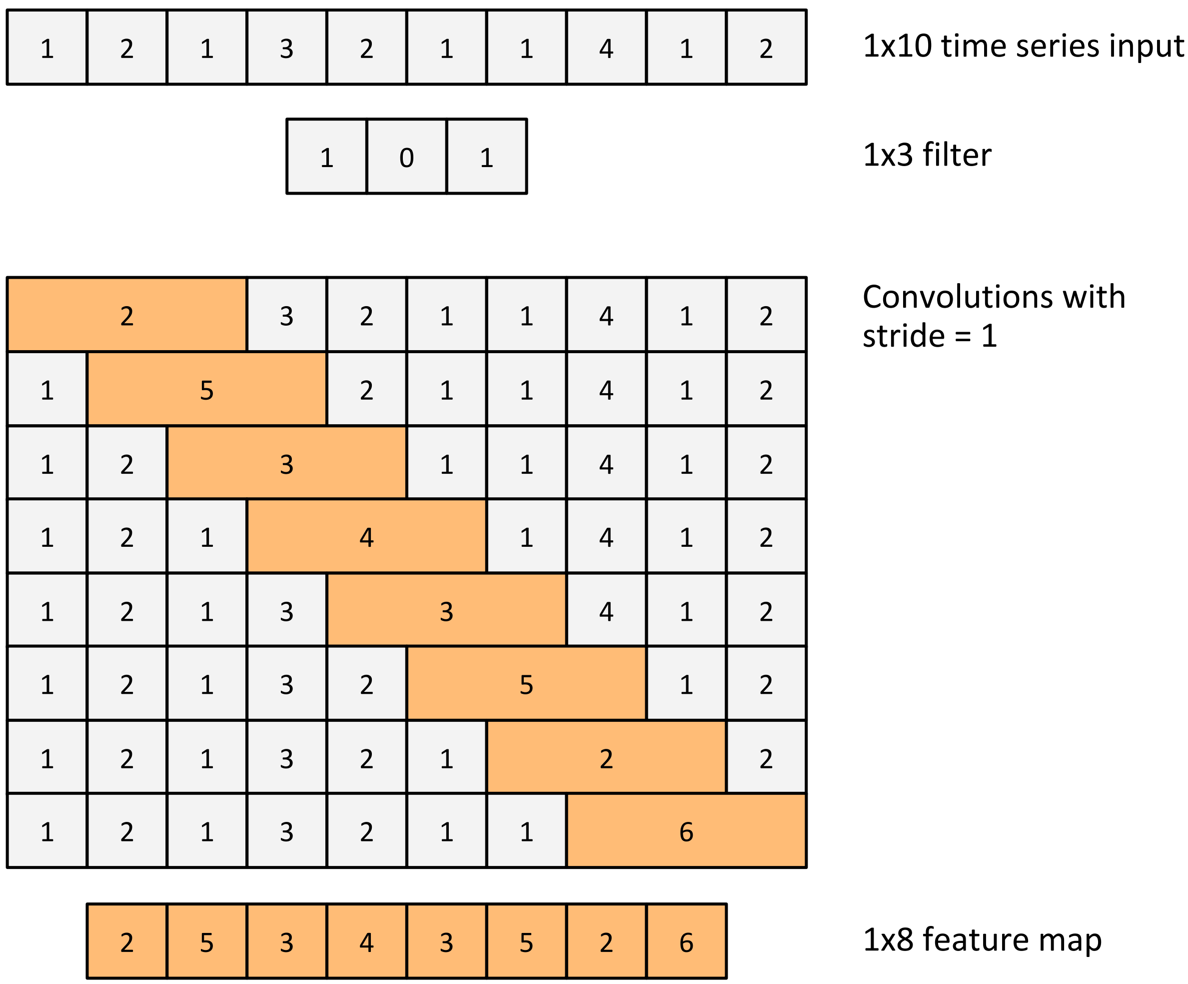


Figure 2.16: 1D convolution on time series data

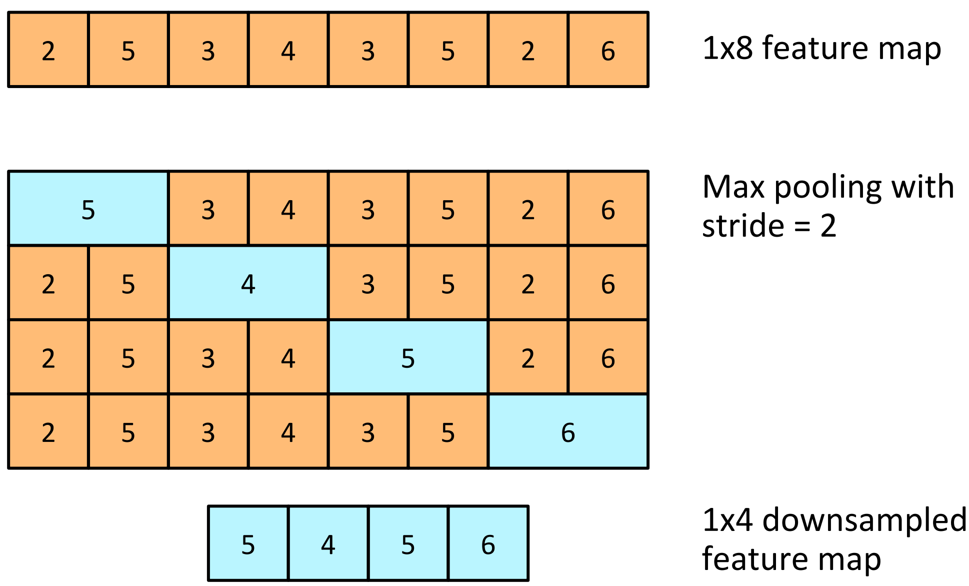


Figure 2.17: Downsampling a feature map using max pooling

Figure 2.18 depicts an example CNN architecture. In this architecture, convolutional layers are stacked against each other, followed by a pooling layer, and then more convolutional layers. Higher up, fully connected dense layers are stacked and produce the model prediction. As more layers are added, the network becomes increasingly deep. In general, lower layers of a CNN capture low-level features of the dataset, intermediate layers combine low-level features from the preceding layers into intermediate level features, and higher layers combine intermediate features into high-level features of the dataset.

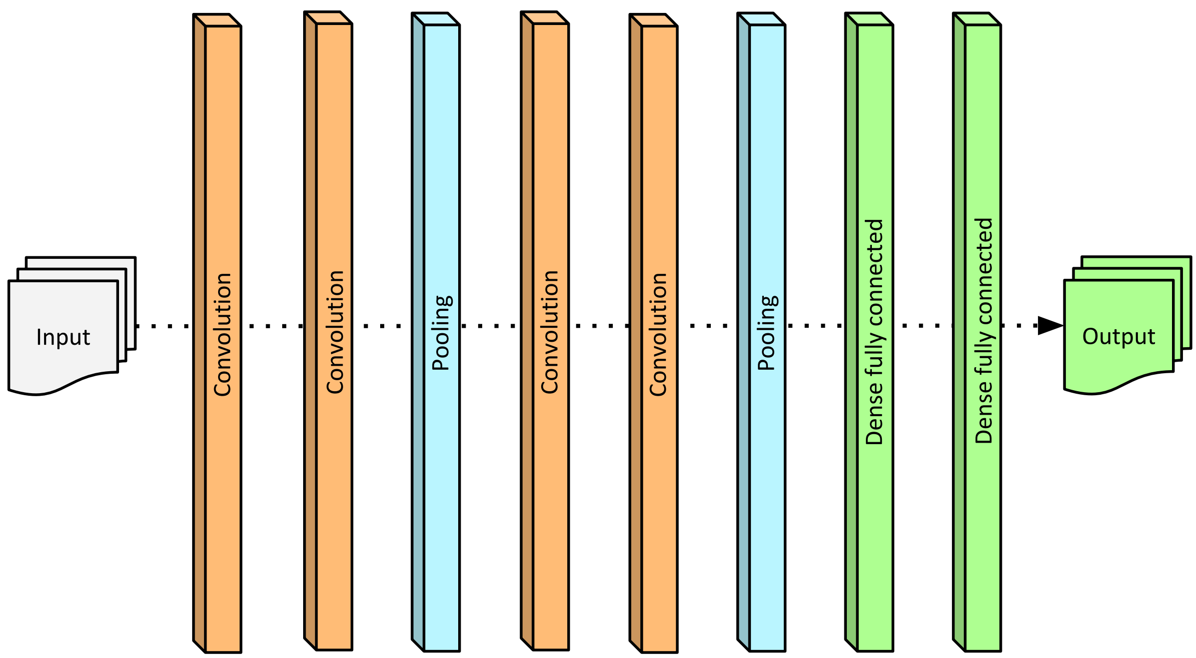


Figure 2.18: Example CNN architecture

Chapter 3

Multi-Family Residence Models

3.1 Introduction

Smart water meters have been installed across Abbotsford, British Columbia, Canada, to measure the water consumption of consumers in the area. Using this water consumption data, we develop machine learning models to predict daily water consumption at the dissemination area level for multi-family residences in the city of Abbotsford. In the related literature, the focus has been on developing machine learning models to predict water consumption for single-family residences, due to the lack of data available for multi-family residences. However, the recent development trend in large urban areas has been towards denser living spaces in the form of multi-family residences.

The focus of this chapter is on building machine learning models for predicting the daily water consumption of multi-family residences in the city of Abbotsford. We discuss the steps of the model building process: data preprocessing, feature engineering, feature selection, and grid searching. We report the performance of models, visualize model predictions, and conclude with an explanation of the determinants of multi-family water consumption. To our knowledge, this work has not been attempted in the current literature. In the next section, we discuss the motivation behind predicting water consumption for multi-family residences and investigate the related work in the literature.

3.2 Motivation

In the existing literature, much of the work to predict water consumption over the short-term has been focused on single-family residences. Due to the lack of data available, there has been little research done in predicting water consumption for multi-family residences [22], [35]. However, the general direction is towards more data availability in the future as smart water meter use becomes increasingly prevalent in large urban areas [59].

Recently, the general trend observed in urban areas has been an increase in the development of denser living spaces in the form of multi-family residences. This has been observed in cities such as New York [35], Seattle [51], Auckland [22], Kuala Lumpur [9], and California's coastal areas [29]. This inclination towards denser living spaces is attributed towards efforts in reducing urban sprawl, economic uncertainty due to the housing market crash in 2008, and to meet housing demand due to increased employment opportunities in urban areas [22], [29]. The following describes the direction from single-family living towards multi-family residence living in more detail:

In the Puget Sound Region, the general trend has been towards the development of apartments and condominiums and the redevelopment of single-family residences into multi-family residences to accommodate the rising population in the area [51]. As a result, total water demand in the Puget Sound Region has decreased over the last 20 years due to urban densification.

There has been a decreasing occupancy rate in Malaysia due to the breakdown of nuclear family living in single-family households and into multi-family households [9]. This trend is particularly pronounced in the urban areas of Kuala Lumpur, Petaling Jaya, and Georgetown. In general, there has been a decrease in per capita water consumption in Southeast Asia due to the shift towards living in multi-family residences which contain fewer landscaping and gardens.

Along California's coastal areas, such as in the city of Pasadena, a similar direction towards denser residential developments in the form of multi-family residences is also taking place [29], [46]. This direction is due to a shift in tastes due to the 2008 economic crisis and as a strategy used by urban planners to reduce urban sprawl in cities. As a result, per capita water consumption has decreased due to smaller lots with less landscaping which are typically found in multi-family residences.

Table 3.1: Related work on finding the determinants of water consumption for multi-family residences

Paper	Model type	Dependent variable(s)	Independent variable(s)	Significant determinants of water use	Spatial scale	Location
Agthe and Billings (2002) [3]	Ordinary least squares regression	Monthly water use per apartment	Property variables	Swimming pool present, water-saving toilets present, number of bedrooms, value per bedroom, building age, water price (10% significance for winter model). Water-saving toilets present, number of bedrooms, value per bedroom, building age, timer irrigation for non-grass landscaping present, water price, percentage of unoccupied apartments (10% significance for summer model).	Property	Tucson, Arizona, USA
Dias et al. (2018) [13]	Stepwise regression	Monthly building water consumption, per capita water consumption	Property variables	Distance to downtown core, percentage of tenants, sewage system present, number of apartments, number of floors, building age, household size, alternative water supply present, building value (5% significance for building model). Distance to downtown core, percentage of tenants, sewage system present, property age, household size, measurement type, swimming pool present, alternative water supply present, apartment value (5% significance for per capita model).	Property, per capita	Joinville, Brazil
Fox et al. (2009) [18]	Univariate classification, multivariate classification	Monthly water demand, average daily water demand (year, summer, winter), peak daily water demand (summer, winter)	Property variables	Number of bedrooms, architectural type, garden presence.	Property	Stevenage, UK
Fullerton and Cardenas (2016) [19]	Linear transfer function	Water usage per customer	Previous water consumption values, demographic variables, climate variables	Property value, total cooling degree days, number of days with rainfall, economic conditions index (5% significance).	Household	Phoenix, Arizona, USA
Ghavidelfar et al. (2016) [22]	Ordinary least squares regression	Annual average daily water consumption (per household, per census tract)	Previous water consumption values, property variables, demographic variables, economic variables, climate variables	Water price, average temperature, total yearly rainfall, time trend, quadratic time trend (1% significance for household model). Household size, water price, average temperature, is low income area, is high income area, time trend, quadratic time trend (1% significance for per capita model).	Household, census area	Auckland, New Zealand
Kontokosta and Jain (2015) [35]	Weighted robust regression, geographically weighted regression	Water use intensity per property	Property variables, demographic variables	Household size, floor area, median household income, renter percentage, energy use intensity, swimming pool present, built between 1910 and 1929, built between 1950 and 1969, is cooperative ownership, is mixed-use condominium ownership (5% significance).	Property	New York City, USA
Polebitski et al. (2010) [51]	Ordinary least squares regression	Total water demand	Property variables, climate variables	Single-family homes per acre, lot size, household size, built after 1992, per capita income, price, average maximum temperature, total precipitation.	City	Puget Sound Region, USA
Wentz et al. (2014) [67]	Stepwise regression	Daily water consumption per bedroom (summer, winter)	Property variables	Pool size, average rent, dishwasher present, building age, heated pool present, unit washer and dryer present, laundry facility present, vegetation present (1% significance).	Household	Tempe, Arizona, USA

Table 3.2: Related work on predicting water consumption for multi-family residences

Paper	**Model type**	**Independent variable(s)**	**Temporal scale**	**Spatial scale**	**Location**
Adamowski (2008) [2]	ANN, multiple linear regression, time series analysis	Previous water consumption values, climate variables	Daily (peak summer)	Pressure zone	Ottawa, Ontario, Canada
Adamowski et al. (2012) [1]	ANN, coupling discrete wavelet transforms	Previous water consumption values, climate variables	Daily	City	Montreal, Quebec, Canada
Ghiassi et al. (2008) [23]	Dynamic artificial neural network	Previous water consumption values, climate variables	Hourly, weekly, monthly	City	San Jose, California, USA
Herrera et al. (2010) [30]	ANN, projection pursuit regression, multivariate adaptive regression splines, random forests, support vector regression	Previous water consumption values, climate variables, day of the week	Hourly	Hydraulic sector	City in south-eastern Spain
Odan and Reis (2012) [45]	ANN, dynamic neural network, hybrid ANNs	Previous water consumption values, climate variables, hour of the day	Hourly	City	Sao Paulo, Brazil
Tiwari and Adamowski (2013) [62]	ANN, Hybrid wavelet-bootstrap-neural network, autoregressive integrated moving average, autoregressive integrated moving average model with exogenous input variables, wavelet analysis-based NN, bootstrap-based NN, naive persistence index model	Previous water consumption values, climate variables	Daily, weekly, monthly	City	Montreal, Quebec, Canada
Tiwari and Adamowski (2014) [63]	Hybrid wavelet-bootstrap-artificial neural network	Previous water consumption values, climate variables	Weekly, monthly	City	Calgary, Alberta, Canada

3.3 Related Work

Table 3.1 and Table 3.2 summarize the related work. Table 3.1 describes work related to finding the determinants of water consumption for multi-family residences. Table 3.2 describes work related to predicting water consumption at several different temporal and spatial scales. The tables briefly describe the type of model, the dependent variables, independent variables, the significant determinants of water use, the time scale, spatial scale, and the location of the study. As we can see, the existing literature only contains work that explains the determinants of water consumption for multi-family residences or predicts water consumption at a differing spatial and temporal scale from the models we build. In our line of work, we predict water consumption at a daily temporal scale and at the dissemination area spatial scale, which we describe in Section 3.4.

The following describe the related work for investigating the determinants of water consumption for multi-family residences: In [3], [13], [19], [22], [35], [51], [66], and [67], the authors investigate the determinants of water consumption in the cities of Tucson, Joinville, Phoenix, Auckland, New York, Seattle, and Tempe. These papers use an ordinary least squares regression analysis to find the determinants of water consumption. Table 3.1 describes the significant determinants of water consumption for multi-family residences. Many of these determinants include various property features such as building age and lot size, climate features such as rainfall and temperature, and demographic features such as occupancy levels. The spatial scale of these studies is across a wide range. Some studies look at the determinants of water usage at the household level, some at the property or building level, while some are at a larger scale at the census area[1] or city level. We use the significant determinants found in these studies to determine the suitable features to engineer, as described in more detail later in the chapter, and to select the appropriate features for our urban planning models in Chapter 4.

There has been much work in the literature which investigates the prediction of water consumption at various spatial and temporal levels. A non-exhaustive list of this work is summarized in Table 3.2. A survey by Donkor et al. [15] provides an extensive review of water demand forecasting methods over the 2000 to 2010 period. For short-term forecasts within one year, artificial neural network models are commonly used. Econometric models are typically used for long-term water demand forecasting within a period of several years. Similarly, water demand forecasting work in the 2005 to 2015 period is reviewed by Ghalehkhondabi et al. [21]. The survey outlines several recent water demand forecasting

[1] A "census area" is a term used in the United States. It is synonymous to what is termed a "dissemination area" in Canada.

models, which range from artificial neural network models, fuzzy and neuro-fuzzy models, and support vector machine models. In general, artificial neural networks have been shown to perform well for predicting water consumption in the short-term.

The most recent related work in the literature, as shown in Table 3.2, investigates a variety of machine learning models to predict water consumption. The models predict water consumption across various cities, mostly in North America. The temporal scale of the models range from predicting hourly consumption to predicting monthly water consumption. In addition, water consumption predictions are mostly across the city scale. When predicting water consumption at the city level, the water consumption for all customer sectors, from residential, commercial, to institutional, are aggregated at the city level. Our models differ from the most recent related work since we are predicting water consumption exclusively for multi-family residences. Our models also predict over a finer spatial scale at the dissemination area level. The model we build in this chapter is not directly comparable to the models depicted in Table 3.2 since none of these models predict daily water consumption at the dissemination area level for multi-family residences. In general, the more aggregated water consumption values are on a spatial level, the easier the prediction task becomes [6], which is why the models which have a spatial scale at the city level are able to obtain accurate performance.

In this chapter and in the following chapters, we focus on predicting water consumption with a daily temporal scale and a spatial scale at the dissemination area level. This is the finest temporal and spatial scales that we are capable of predicting over to obtain adequate performance. Predicting at a finer spatial scale, such as at the household or building level will not result in adequate performance as water consumption at this scale is too variable. We cannot predict at an hourly temporal scale due to some smart water meters recording usage in bursts for contiguous hours of the day rather than for each hour of the day, as specified in Section 2.1.3. Therefore, we opt to predict at a daily temporal scale.

3.4 Problem Statement

In this section, we detail the problem definition of the chapter. We predict daily water consumption for multi-family residences in the city of Abbotsford using water consumption data collected from the period between September 1, 2012 to August 31, 2013. We build a single machine learning model to predict the daily water consumption per unit at the dissemination area level. The inputs of the model are those specified in Section 3.8 and the output of the model is the predicted daily water consumption per unit for a particular dissemination area, where per unit water consumption is aggregated at the dissemination area level. We are not predicting water consumption for a particular household, but rather a typical daily per unit water consumption for a dissemination area. For each dissemination area, we have daily aggregated water consumption data as the target variable from the period between September 1, 2012 to August 31, 2013. The data for each dissemination area is concatenated into a single dataset. This dataset contains the water consumption for all dissemination areas as the target variable along with the corresponding features and is used as the training dataset, minus the dissemination area being predicted. Table 3.3 provides an outline of the model we are building. Overall, the main objective of this chapter is to build a machine learning model which predicts daily water consumption for multi-family residences at the dissemination area level.

Table 3.3: Model characteristics for predicting daily water consumption of multi-family residences at the dissemination area level

Single or multiple models?	Single
Model input(s)	Selected features from Section 3.8
Model output(s)	Predicted daily water consumption per unit for a dissemination area, where per unit water consumption is aggregated at the dissemination area level.
Temporal scale	Daily
Spatial scale	Dissemination area
Train dataset	Target variable: Daily aggregated water consumption data from September 1, 2012 to August 31, 2013 for each dissemination area. Data for each dissemination area is concatenated into a single training dataset.
Test dataset	Target variable: Daily aggregated water consumption data from September 1, 2012 to August 31, 2013 for the dissemination area to predict.

3.5 Performance Metrics

Before proceeding further, we discuss the performance metric used to evaluate model performance and the reasoning behind choosing the particular performance metric. We selected the mean absolute error (MAE) as the performance metric used to evaluate model performance as it is easier to interpret compared to the mean squared error (MSE) which is another commonly used metric. The mean absolute error retains the same unit as the target variable. In our case, the MAE will be expressed in litres. The MAE is the average of the absolute difference between two continuous variables and is expressed below. For our purposes, we use the mean absolute error to measure the absolute difference between actual and predicted values of the target variable.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

In contrast, the MSE takes the average of the squared difference between two continuous variables, and is expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

In addition, we select the MAE over the MSE as a metric since the MSE more greatly penalizes larger errors and is sensitive to outliers due to taking the squared difference between variables. In general, we want both large and small errors to be penalized equally, as we want to obtain as robust a fit as possible and to predict well in the general case. As mentioned in the data preprocessing section, outlying values for the daily water consumption dataset were not removed unless it was due to an error. We are not concerned with predicting these outlying values over predicting typical water consumption. In addition, there are certain dissemination areas which are more difficult to predict, as described later in Section 5.11. We want the model performance on these dissemination areas to be penalized equally with the other dissemination areas.

3.6 Data Preprocessing

The next sections outline the steps taken to prepare the water consumption data to be suitable for our machine learning task. For the climate, property, and demographic data, minimal preprocessing was required. Detailed information on the datasets used throughout the thesis is described in Section 2.1.4.

3.6.1 Mapping Meters to Dissemination Areas

Each meter in the water consumption data for Abbotsford was not originally mapped to a particular dissemination area. Originally, meters were mapped to the street address which the meter records water consumption usage over. Since we are predicting water consumption at the dissemination area level, we are required to map each meter to a dissemination area. This was done in the work by Platsko [49], [50] and the information was provided to be used in this thesis. In [49], the latitude and longitude is determined for each meter's street address and is then used in conjunction with a dissemination area boundary file to map each street address to a dissemination area.

3.6.2 Time Zone Adjustment

Water consumption for the city of Abbotsford was recorded in Coordinated Universal Time (UST) rather than in the local Pacific Standard Time (PST). We adjusted the water consumption data from UST to PST.

3.6.3 Case Deletion

Water consumption data contains a meter reading for each multi-family residence. Since water consumption is recorded at the hourly level from September 1, 2012 to August 31, 2013, each meter reading will have 8760 (24 hours $\times$ 365 days) records. We started with 878 meters in total. Certain meters had to be case deleted due to excessive missing data or meter errors. Since these meters had missing data or errors in excess, it would have been difficult to accurately impute missing values, which is why these meters were removed entirely. A record is considered missing if the missing indicator attribute is set to true for that record. Meters were removed with the following criteria and in this order:

1. Remove meters with more than two weeks of contiguous data missing.
2. Remove meters with an annual water utilization rate of less than 27%. This excludes meters which are programmed to record in bursts for contiguous hours of the day.
3. Remove meters with zero daily usage occurring more than 120 days.

After this case deletion step, we were left with 635 meters in total. It should be noted that in step (1), two weeks was chosen as we are not able to accurately impute more than two weeks of contiguous missing data. For step (2), we aimed to keep meters with at least 30% water utilization. This means that water usage is recorded during at least 30% of the hours in a day for a household. Any less meant that a unit was vacant or vacant for a majority of the entire recording period. After removing meters with less than 30% water utilization, we found households which had low utilization but had water usage patterns indicative of a non-vacant unit. These households had the typical diurnal pattern, but with lower than average utilization. These meters were added back which resulted in 27% being the threshold value. In step (3), 120 days was found to be a robust value. Any number of days within a reasonable range of this value would remove a similar number of meters.

3.6.4 Imputation

After performing the case deletion step, we worked on imputing missing values in the water consumption data. First, we ensured that water consumption was recorded across a consistent unit. In this case, water consumption data was recorded in cubic meters and this is consistent across the whole dataset. Throughout the entire thesis, such as when training machine learning models, we work with the water consumption data in cubic meters. However, we present the results in litres as this unit is easier to interpret. Second, for each meter, we checked for outlying values. No outliers were clear measurement errors, so all outlying values were included in the water consumption dataset.

Next, since the goal of our machine learning models is to predict daily water consumption, we calculate the daily water consumption for all meters by summing up the hourly usages over 24 hours for each day of the year. This resulted in each meter reading having 365 records, one for each day of the year.

For the next step, we looked at each meter individually and imputed the days which had missing hours. A day is considered to be missing if an hour is missing from that day. This excluded the days where there were widespread water meter outages in the city, as this issue was resolved later on in the data cleaning process. To impute the missing values, we built a support vector regression (SVR) model to predict the missing values. The SVR model had two input features: *Daily water usage one week ago* and *Daily water usage next week*. The particular model, model parameters, and input features were chosen based on performance. We ran a k-fold cross-validation to compare different models, where $k = 5$. Different values of k within this range produced similar results.

After this step, we discovered five meters which suffered from widespread outages throughout the year. We case deleted these five meters which left us with 630 meters in total. Figure 3.1 shows the locations of the 630 meters after conducting the case deletion and imputation step.

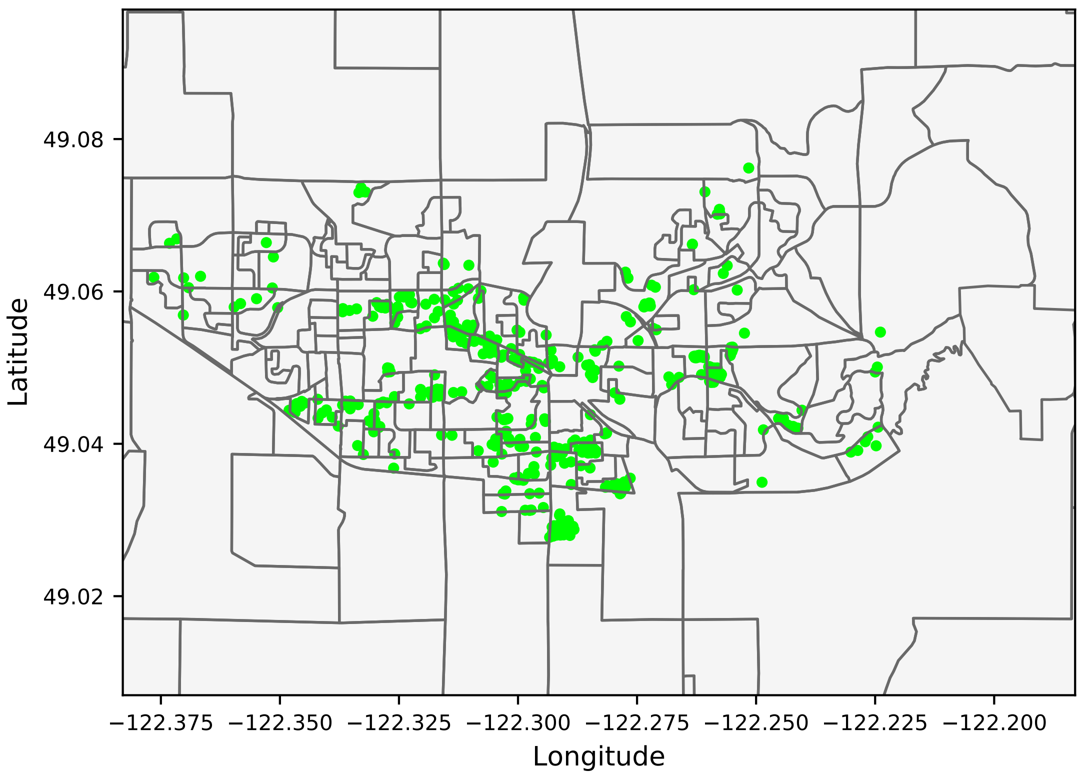


Figure 3.1: Locations of smart water meters installed in multi-family residences across Abbotsford

3.6.5 Merging Dissemination Areas

Since our machine learning models predict at the dissemination area level, we need to ensure that each dissemination area contains at least ten meters. If a dissemination area contains less than ten meters, this would be similar to predicting water consumption at the household level, which is likely not to perform well or provide accurate results. Originally, several of the dissemination areas contained less than ten meters. Since we did not want to lose data, we chose to merge dissemination areas which were geographically adjacent to each other until there were at least ten meters in the merged dissemination areas. However, it is possible to have several different options of which dissemination areas to merge since a dissemination area can be adjacent to many other dissemination areas. To deal with this issue, we used k-means clustering, as specified in [39], to cluster dissemination areas based on demographic features. So, to merge dissemination areas, we merge it with the dissemination areas it is adjacent to and which also fall into the same k-means cluster. This way, we are merging dissemination areas which are geographically adjacent and are roughly similar to each other in terms of demographics. Table 3.5 shows the demographic features obtained from the 2011 National Household Survey which were used in the k-means clustering. In addition, we did not merge dissemination areas which originally contained ten or more meters. And, the only case where we dropped a dissemination area was when it was not adjacent to any other dissemination areas and did not have the adequate number of meters. Originally, there were 87 dissemination areas in total. After merging dissemination areas so that each contains at least ten meters, we end up with 30 dissemination areas, as depicted in Figure 3.2 and listed in Table 3.4. Each entry in the legend of Figure 3.2 shows the dissemination areas which have been merged together, with the exception of areas shaded in grey. If a dissemination area is shaded in grey, this means that particular dissemination area was not merged with any other dissemination areas due to having an adequate number of meters. It should be noted that we retained a few dissemination areas with slightly less than ten meters as it would have been more difficult to merge these with other dissemination areas. We also note that when we refer to a merged dissemination area in the thesis, we address it by the first DAUID listed in Table 3.4 for simplicity.

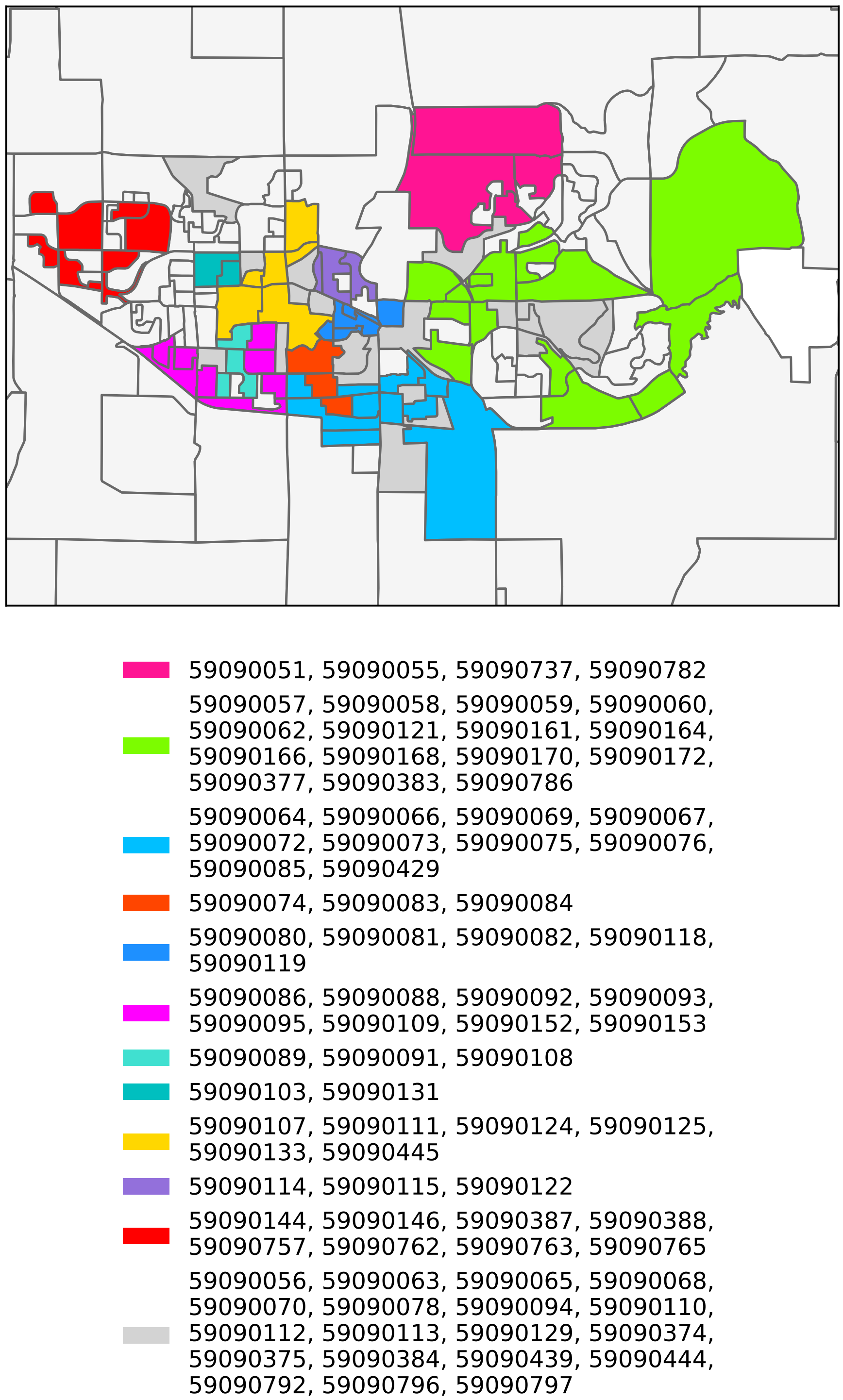


Figure 3.2: Merged dissemination areas in Abbotsford

Table 3.4: Merged dissemination areas in Abbotsford, each with a number of smart water meters and housing units for multi-family residences.

DAUID	# meters	# units
59090051	11	176
59090055		
59090737		
59090782		
59090056	10	74
59090057	29	621
59090058		
59090059		
59090060		
59090062		
59090121		
59090161		
59090164		
59090166		
59090168		
59090170		
59090172		
59090377		
59090383		
59090786		
59090063	10	101
59090064	45	1014
59090066		
59090067		
59090069		
59090072		
59090073		
59090075		
59090076		
59090085		
59090429		
59090065	31	61

DAUID	# meters	# units
59090068	71	71
59090070	106	604
59090074	10	49
59090083		
59090084		
59090078	10	201
59090080	32	1125
59090081		
59090082		
59090118		
59090119		
59090086	21	235
59090088		
59090092		
59090093		
59090095		
59090109		
59090152		
59090153		
59090089	10	20
59090091		
59090108		
59090094	8	121
59090103	12	306
59090131		
59090107	27	1327
59090111		
59090124		
59090125		
59090133		
59090445		
59090110	9	316

DAUID	# meters	# units
59090112	8	297
59090113	14	399
59090114	10	177
59090115		
59090122		
59090129	11	112
59090144	14	436
59090146		
59090387		
59090388		
59090757		
59090762		
59090763		
59090765		
59090374	22	139
59090375	10	10
59090384	26	36
59090439	10	369
59090444	10	226
59090792	9	9
59090796	20	20
59090797	12	78

Table 3.5: Demographic features used in k-means clustering, each described in Table 3.11

Demographic feature
No certificate, diploma or degree %
High school diploma or equivalent %
Postsecondary certificate, diploma or degree %
Participation rate
Employment rate
Unemployment rate
Average family size
Average number of rooms per dwelling
Median value of dwellings ($)
Renter %

3.6.6 Aggregating Meters

Next, we aggregate the meters in each dissemination area. After the aggregation, each dissemination area will have 365 records, one for each day of the year. Each record contains the per unit water usage for a particular day of the year. The per unit water usage is calculated by taking the sum of the water usage for all meters in the dissemination area for a particular day and dividing it by the total number of units in the dissemination area. We ensured that the number of units for each meter was correct by observing the per unit water usage. If the per unit water usage was unrealistically high or low, we used the BCAssessment property assessment tool to verify the number of units for each address. If there was a discrepancy in the number of units, we would change the value to what was recorded by the BCAssessment tool. We use the per unit water consumption since it is a way of standardizing the usage across dissemination areas, as each dissemination area has a different size and total usage.

3.6.7 Fixing Recording Outages

The final step of the data cleaning process is imputing missing values for days affected by a recording outage across the city of Abbotsford. These outages were due to hardware maintenance. Table 3.6 shows the days when these outages occur.

Table 3.6: Smart water meter outages across Abbotsford

Date of meter outage
February 16, 2013
February 17, 2013
March 9, 2013
March 10, 2013
March 11, 2013
March 30, 2013
March 31, 2013
April 28, 2013
July 27, 2013
July 28, 2013
July 29, 2013

To impute the missing values, we built a linear regression model for each dissemination area to predict the missing values. The features of the model are *Daily water usage three weeks ago*, *Daily water usage two weeks ago*, *Daily water usage one week ago*, *Daily water usage next week*, *Daily water usage in the next two weeks*, and *Daily water usage in the next three weeks*. The model type and features were selected by comparing across a wide range of different models and feature sets and selecting the configuration which achieved the best performance in terms of the mean absolute error. The performance was obtained by running a repeated k-fold cross-validation where $k = 5$. We note that using a k value within this range produced similar results.

As a final sanity check, we made sure that we had all days of the year for each dissemination area, all missing data was imputed, the average, minimum, and maximum water usage values were within realistic limits, and that there were no outliers that were clear measurement errors.

3.7 Feature Engineering

In this section, we discuss the feature engineering step of the model building process and describe the features we engineer from our data. Feature engineering is the process of creating features for machine learning models and requires domain knowledge of the dataset. During the feature engineering step, we constructed features from water consumption data, climate data, property data, and demographic data. We constructed features based on what features have been shown to work well in the past and also based on what we thought would work well intuitively.

For the water consumption data, we had daily water usage per unit along with the timestamp and DAUID as attributes. In the literature, previous water consumption values, or lagged values, have been shown to work well [2], [1], [23], [30], [45], [62], [63]. Based on this knowledge, we constructed features based on previous water consumption values. We also constructed statistical water consumption features such as taking the average of previous water consumption. The constructed water consumption features are summarized in Table 3.7.

Table 3.7: Engineered water consumption features

Feature	Description
Water usage four weeks ago	Daily aggregated water usage at four weeks ago (cubic meters per unit)
Water usage three weeks ago	Daily aggregated water usage at three weeks ago (cubic meters per unit)
Water usage two weeks ago	Daily aggregated water usage at two weeks ago (cubic meters per unit)
Water usage one week ago	Daily aggregated water usage at one week ago (cubic meters per unit)
Water usage three days ago	Daily aggregated water usage at three days ago (cubic meters per unit)
Water usage two days ago	Daily aggregated water usage at two days ago (cubic meters per unit)
Water usage one day ago	Daily aggregated water usage at one day ago (cubic meters per unit)
Average of water usage at three, two, and one week ago	Average of daily aggregated water usage at three, two, and one week ago (cubic meters per unit)
Average of water usage at three, two, and one day ago	Average of daily aggregated water usage at three, two, and one day ago (cubic meters per unit)

Table 3.8: Engineered calendar features

Feature	Description
Month	Numerical index for each month: 1 for January and 12 for December
Day	Numerical index for the day of the month: 1 for first day of the month and 30 or 31 for last day of the month
Weekday	True if day is a weekday
Weekend or holiday	True if day is a weekend or holiday
Friday	True if day is a Friday
Fall	True if month is between October to December
Winter	True if month is between January to March
Spring	True if month is between April to June
Summer	True if month is between July to September

Table 3.9: Engineered climate features

Feature	Description
Daily temperature	Average daily temperature (°C)
Average temperature over last seven days	Average daily temperature over the last seven days (°C)
Daily barometer	Average daily air pressure (hPa)
Daily wind speed	Average daily wind speed (m/s)
Daily rainfall	Average daily rainfall (cm)
Total rainfall over last seven days	Total rainfall over the last seven days (cm)
Days since last rainfall	Number of days since last rainfall
Cooling degree day	Number of degrees over 18 °C for the day

Next, we constructed calendar features such as the season, whether it is a weekend or weekday, and the time of month and day. These features were constructed since it was observed in the water consumption data that weekend water consumption tends to be greater than weekday water consumption and Friday water consumption tends to be smaller than other weekdays. Some dissemination areas tend to have more of a seasonal pattern where water usage is greater in the spring and summer months compared to the fall and winter months. The constructed calendar features are described in Table 3.8.

For climate data, we use the average daily temperature and daily total rainfall features already present in the dataset to construct additional climate features. It has been shown in the literature that climate features such as temperature, rainfall, and wind speed are determinants of water consumption. The climate features we constructed are described in Table 3.9.

The property assessment data we obtained from BCAssessment was missing a large amount of data for multi-family residences. Therefore, we manually obtained property features from the BCAssessment property assessment tool. For each property in a dissemination area, we looked up its address using the tool. The search result provides property assessment information for the property. In the previous literature, it has been shown that property type, the number of bedrooms, and the number of bathrooms are determinants of water consumption. Since the model we are building predicts at the dissemination area level, property features are aggregated at the dissemination area level. The resulting property features are outlined in Table 3.10.

Table 3.10: Engineered property features

Feature	**Description**
Duplex or townhouse %	Percentage of properties in DA which are duplexes or townhouses
Multiple residence or strata apartment %	Percentage of properties in DA which are multiple residences or strata apartments
Total units	Total number of units in DA
Total property area	Sum of property areas in DA (square feet)
Average property area	Average of property areas in DA (square feet)
Total number of stories	Sum of the number of stories of properties in DA
Average number of stories	Average number of stories of properties in DA
Average year built	Average year built of properties in DA
Total number of bedrooms	Total number of bedrooms for properties in DA
Average number of bedrooms	Average number of bedrooms for properties in DA
Total number of bathrooms	Total number of bathrooms for properties in DA
Average number of bathrooms	Average number of bathrooms for properties in DA

We constructed demographic features using the data provided in the 2011 National Household Survey. Features were constructed based on features which have been shown to work well in the past for predicting water consumption, such as income and education. All demographic features occur at the dissemination area level and for multi-family residences only. For merged dissemination areas which are made up of multiple dissemination areas, features are built by taking a weighted sum of the feature values. To do this aggregation, we take the sum of each feature value and divide it by the total number of units in the dissemination area. Table 3.11 describes the constructed features in detail.

Table 3.11: Engineered demographic features

Feature	**Description**
No certificate, diploma or degree %	Percentage of population in DA with no certificate, diploma or degree.
High school diploma or equivalent %	Percentage of population in DA with a high school diploma.

Postsecondary certificate, diploma or degree %	Percentage of population in DA with a postsecondary certificate, diploma or degree.
Participation rate	Participation rate in DA. The participation rate is defined as the percentage of the population who are employed or looking for work.
Employment rate	Employment rate in DA. The employment rate is defined as the percentage of the labour force who are employed, where the labour force is defined as those who are employed or unemployed.
Unemployment rate	Unemployment rate in DA. The unemployment rate is defined as the percentage of the labour force who are unemployed, where the labour force is defined as those who are employed or unemployed.
Population who did not work in 2010 %	Percentage of population in DA who did not work in 2010.
Population who worked full-time in 2010 %	Percentage of population in DA who worked full-time in 2010.
Population who worked part-time in 2010 %	Percentage of population in DA who worked part-time in 2010.
Population worked at home in 2010 %	Percentage of population in DA who worked at home in 2010.
Couple-only economic families %	Percentage of households in DA which are couple-only economic families.
Couple-with-children economic families %	Percentage of households in DA which are couple-with-children economic families.
Lone-parent economic families %	Percentage of households in DA which are lone-parent economic families.
Average family size	Average household family size in DA.
Average number of rooms per dwelling	Average number of rooms per dwelling in DA.
Median household income	Median household income in DA ($).
Median value of dwellings	Median value of dwellings in DA ($).
Renter %	Number of private households in DA who are renters.

3.8 Feature Selection

In this section, we perform a feature selection step which selects the features we use in our models to predict water consumption for multi-family residences. After the feature engineering step, we are left with many features, some of which may be irrelevant or redundant. The purpose of performing a feature selection step is to exclude irrelevant and redundant features, which improves generalization by reducing overfitting, results in shorter training times, and makes machine learning models more interpretable.

There are three main categories of performing feature selection. These include wrapper methods, filter methods, and embedded methods. We opt out of using filter and embedded methods since these techniques tend to be simpler and typically result in features which provide lower prediction performance compared to computationally intensive wrapper methods. To perform feature selection, we select a wrapper method from the Scikit-learn framework called recursive feature elimination with cross-validation (RFECV) [48].

Recursive feature elimination is a greedy optimization algorithm which repeatedly trains an estimator, such as a support vector machine, on smaller and smaller feature sets. Initially, the estimator is trained on the whole set of features. In each iteration, a feature importance value is obtained from the estimator for each feature. The feature with the lowest importance value is pruned from the current set of features being considered. This process continues recursively until there is only one feature in the set of features being considered. The optimal set of features is chosen by performing a k-fold cross-validation [48].

We run recursive feature elimination with 6-fold cross-validation on six randomly selected dissemination areas (out of the 30 dissemination areas total). These six dissemination areas are used exclusively for conducting feature selection. For each iteration of the 6-fold cross-validation, one dissemination area is set aside as the test set and the remaining five dissemination areas are set aside as the training set. Table 3.12 shows the features that were selected by RFECV, along with its feature ranking value. A lower value for the rank is considered a more important feature.

We chose SVR with a linear kernel and default parameters (C = 1.0, epsilon = 0.1) to use as the estimator for recursive feature elimination. This particular model with default parameters was chosen because it provided consistent results over multiple runs, unlike the regression tree and tree ensemble estimators where the selected features were inconsistent over multiple runs. The selected features using the SVR model are robust since using other estimators also produce similar selected features.

Table 3.12: Selected features from recursive feature elimination

Feature	Feature ranking
Average of water usage at three, two, and one week ago	1
Water usage one week ago	2
Daily temperature	3
Average temperature over last seven days	4
Average of water usage at three, two, and one day ago	5
Water usage two weeks ago	6
Water usage three weeks ago	7

Since the seven selected features have differing units, we standardize our data before passing it through any machine learning algorithm. This applies throughout the rest of the thesis as well. We standardize our data by subtracting the mean and dividing by the standard deviation for each feature, where the mean is the mean value of that feature and the standard deviation is the standard deviation of that feature. The mean and standard deviation are obtained only from the training data to prevent test leakage.

3.9 Training Models

In this section, we train machine learning models using the selected features obtained from the previous section to predict daily water consumption for multi-family residences in the city of Abbotsford. We train a variety of models provided in the Scikit-learn machine learning framework. For now, we will use the default model parameters in Scikit-learn. In the next section, we will tune the model parameters to be the most optimal.

When training the machine learning models, we use the remaining 24 dissemination areas for training and testing. The previous six dissemination areas are used exclusively for feature selection. We perform k-fold cross-validation where the number of folds is set to the number of dissemination areas. The reasoning behind this is so that there is no test leakage on the dissemination area being used as the test set. In each iteration of the k-fold cross-validation, one dissemination area is set as the test dataset and the remaining 23 dissemination areas are set as the training dataset. This repeats until all dissemination areas become the test set at some point. The performance metric (mean absolute error) for the test set is averaged over all the cross-validation folds. Table 3.13 shows the performance metric on the test set using default model parameters.

Table 3.13: Multi-family residence model performance using default model parameters

Model	**Test MAE (litres)**
Linear Regression	42.49
LinearSVR	42.62
Gradient Boosting	43.91
Random Forest	44.23
Neural network with two hidden layers	44.83
Decision Tree	45.70
Neural network with one hidden layer	45.79
SVR	51.49
KNN	52.41
AdaBoost	67.83

3.10 Grid Search

In this section, we tune the parameters of our machine learning models. Typically, the parameters of a machine learning model need to be tuned to obtain good performance. Parameters in a machine learning model can be thought of as settings and each setting can take on a different value. For a machine learning model to perform well, these settings must be fine-tuned. The objective is to find an optimal set of parameters which enable our model to perform well. Finding the optimal set of parameters is typically done by performing a parameter search. We use two different methods for parameter searching: exhaustive grid search and randomized grid search, both available in the Scikit-learn framework. Exhaustive grid search evaluates model performance on all combinations of parameter values specified in a grid. Randomized grid search evaluates a random subset of parameter sets rather than exhaustively [20]. For both methods, the parameter set which obtained the best performance is reported.

The structure of performing a parameter search is as follows: We use the 24 remaining dissemination areas as the train and test datasets. We structure our code such that there is an outer cross-validation loop and an inner cross-validation loop. The outer cross-validation splits the 24 dissemination areas into a train and test set. For each fold of the cross-validation, one dissemination area is set as the test set and the remaining 23 dissemination areas are set as the train set. Grid search (either exhaustive or randomized) is performed within the inner cross-validation loop. Inside the inner cross-validation, a grid search is performed on the 23 dissemination areas. These 23 dissemination areas are those set aside as the train data in the outer cross-validation loop. The optimal parameters are returned in each run of the inner cross-validation. Back to the outer cross-validation, a model is built using the optimal parameters returned by the grid search. We evaluate the performance of this model on the test set which is the one dissemination area. After the completion of the outer cross-validation loop, the performance metric of the test set is averaged out to get the final test result. The pseudocode for the parameter search code can be viewed in Listing 3.1 and Listing 3.2.

To determine the optimal parameters for each model, we perform a series of coarse grid searches followed by a series of fine grid searches. We run the code in Listing 3.1 and Listing 3.2 multiple times, using either an exhaustive or randomized grid search. The reasoning behind this is that performing only a single exhaustive grid search over a large parameter grid is too computationally expensive. If the grid is large and exhaustive enough, performing grid search becomes computationally intractable. As the number of parameters in the grid increases, the cost of running an exhaustive grid search grows exponentially.

Therefore, the strategy is to first focus on conducting a series of randomized grid searches that search over a coarse parameter grid. During these searches, we note which model parameter values tend to perform well. Once we have narrowed down the parameter values to a reasonable number, we then conduct a series of exhaustive grid searches that search over a finer parameter grid. Similarly, during these finer searches, we note which model parameter values tend to perform well. Using this information, we revise the parameter grid again, to an even finer grid. We repeat this process until we find the optimal parameter set for the model.

We run through this strategy of finding the optimal parameter set for each type of model we are building. The optimal parameters for each model are reported in Table 3.14 and are described in detail in [48]. Parameters not reported use the Scikit-learn framework defaults. The test results using optimal parameter sets is outlined in Table 3.15. As we can see, the improvement in performance ranges from being minimal to significant compared to using default model parameters. For each model, we find that adequate performance can be obtained from a reasonable range of values for each parameter.

Listing 3.1: Outer cross-validation loop for grid searching

```
def outer_cross_validation(DA_list):

    test_errors = []

    for train_index, test_index in kfold_split(DA_list):

        test_DA = get_kth_DA(DA_list, test_index)
        train_DAs = get_remaining_DAs(DA_list, train_index)
        best_model_parameters = inner_cross_validation(train_DAs)
        model = train_model(best_model_parameters, train_DAs)
        test_error = compute_test_error(model, test_DA)
        test_errors.append(test_error)

    average_test_error = get_average(test_errors)
    return best_model_parameters, average_test_error
```

Listing 3.2: Inner cross-validation loop for grid searching

```
def inner_cross_validation(DA_list):

    model_parameter_grid = get_parameter_grid()

    for parameters in model_parameter_grid:

        parameter_errors = []
        test_errors = []

        for train_index, test_index in kfold_split(DA_list):

            test_DA = get_kth_DA(DA_list, test_index)
            train_DAs = get_remaining_DAs(DA_list, train_index)
            model = train_model(parameters, train_DAs)
            test_error = compute_test_error(model, test_DA)
            test_errors.append(test_error)

        average_test_error = get_average(test_errors)
        parameter_errors.append((parameters, average_test_error))

    best_model_parameters = get_best_parameters(parameter_errors)
    return best_model_parameters
```

Table 3.14: Grid searched parameters for multi-family residence models

Model	**Optimal parameters**
Linear Regression	-
KNN	n_neighbors: 49 p: 1 weights: "distance"
LinearSVR	C: 1 epsilon: 0.01 loss: "epsilon_insensitive"
SVR	C: 100 epsilon: 0.001 gamma: 0.001 kernel: "rbf"
Neural network with one hidden layer	activation: "identity" alpha: 0.0001 batch_size: 250 hidden_layer_sizes: (2) learning_rate_init: 0.0001 solver: "lbfgs"
Neural network with two hidden layers	activation: "identity" alpha: 0.0001 batch_size: 300 hidden_layer_sizes: (2, 2) learning_rate_init: 0.0001 solver: "lbfgs"
Decision Tree	criterion: "mse" max_depth: 10 max_features: None min_samples_leaf: 60 splitter: "best"
Random Forest	criterion: "mse" max_depth: 50 max_features: None min_samples_leaf: 30 n_estimators: 80

AdaBoost	learning_rate: 1.0 loss: "exponential" n_estimators: 140
Gradient Boosting	criterion: "mse" loss: "huber" max_depth: 4 max_features: None min_samples_leaf: 50 n_estimators: 70 subsample: 0.9

3.11 Model Performance

Table 3.15: Multi-family residence model performance using grid searched parameters

Model	**Test MAE (litres)**
SVR	42.05
LinearSVR	42.31
Neural network with one hidden layer	42.43
Neural network with two hidden layers	42.44
Linear Regression	42.49
Gradient Boosting	43.42
Random Forest	43.88
Decision Tree	45.73
KNN	47.81
AdaBoost	49.87

Overall, the best performing model after conducting a grid search is SVR with a test MAE of 42.05. This means that on average, the difference between the actual and predicted values for daily water consumption per unit at the dissemination area level is 42.05 litres. Since there are currently no models in the literature which predict water consumption for multi-family residences at the same temporal and spatial scale, no fair comparison can be made. Typical water consumption in Abbotsford for multi-family residences is 592 litres per day per unit, meaning that the test error makes up only 7% of the average daily water usage. We note that the 592 litres is calculated by taking the per capita water usage from

[57] and multiplying it by the average family size in Abbotsford which was obtained from the 2011 National Household Survey. The required accuracy of prediction models depends on a multitude of factors, such as the scarcity of water as a resource, the current state of the water infrastructure, and the cost of infrastructure expansion [6]. In general, greater accuracy can be expected from predicting water consumption over a larger spatial scale, such as at the city level, as opposed to a smaller spatial scale such as over a dissemination area.

3.12 Model Predictions

In this section, we visualize model predictions by plotting the actual and predicted water consumption for a specific dissemination area. We chose a large dissemination area (DAUID: 59090068), containing 71 multi-family units with consistent water usage throughout the year. We train the best performing model, SVR, using the seven selected features and the grid searched parameters found in the previous sections. The SVR model is trained on 29 dissemination areas, excluding the dissemination area that we are plotting. We plot the usage from January 1, 2013 to June 30, 2013, as shown in Figure 3.3.

As we can see, the peaks and troughs of water consumption are difficult to predict well. In the current literature, this is also a common issue, such as in the work by Walker et al. [65]. Analyzing the data further, we find that peaks tend to occur on the weekends, and troughs tend to occur on Fridays. The reason as to why peaks and troughs are not being captured well is likely due to the lack of weekend and Friday data that we have, and also due to the features we are using. The seven features used in our model do not include features relating to weekends or Fridays, so we included the *Weekend or Holiday* feature and the *Friday* feature to our feature set. We trained the SVR model again, this time using nine features total, as outlined in Table 3.16. In Figure 3.4, we can see that weekends and Fridays are captured better. The test MAE for the SVR model improves slightly when using nine features, at 40 litres per day per unit.

The wrapper method we used to do feature selection, RFECV, did not select the *Weekend or holiday* feature or the *Friday* feature. We experimented with several different estimators to use for RFECV along with varying the parameters, as well as experimenting with different cost functions, but no approach selected the two features. The reasoning behind this is likely because RFECV does not consider every possible feature set. Instead, in each step, it prunes out features with the lowest importance value from the current feature set. Exhaustive feature selection methods do not work in practice for large initial feature sets, as an exhaustive search would be computationally intractable.

Table 3.16: Revised feature set to better capture peaks and troughs

Feature
Water usage three weeks ago
Water usage two weeks ago
Water usage one week ago
Average of water usage at three, two, and one week ago
Average of water usage at three, two, and one day ago
Daily temperature
Average temperature over last seven days
Weekend or holiday
Friday

In general, water utilities are interested in determining peak daily water usage, since this information helps utilities determine capacity requirements and water rates [6]. For Abbotsford, determining peak usage is important as the city is currently under a lack of water infrastructure [64]. We removed the *Average of water usage at three, two, and one day ago* feature from the features specified in Table 3.16, since including this feature seemed to have the effect of smoothing out the larger weekend peaks. We train our SVR model again and see that we are able to capture peaks even better, as shown in Figure 3.5. However, the test MAE of the SVR model suffers slightly, at 47 litres per day per unit. Also, it is still difficult to predict larger than average peak values. This is likely due to the lack of data to train from.

As future work, model performance will likely improve if one model is trained for predicting weekday water consumption and another model is trained for predicting weekend water consumption.

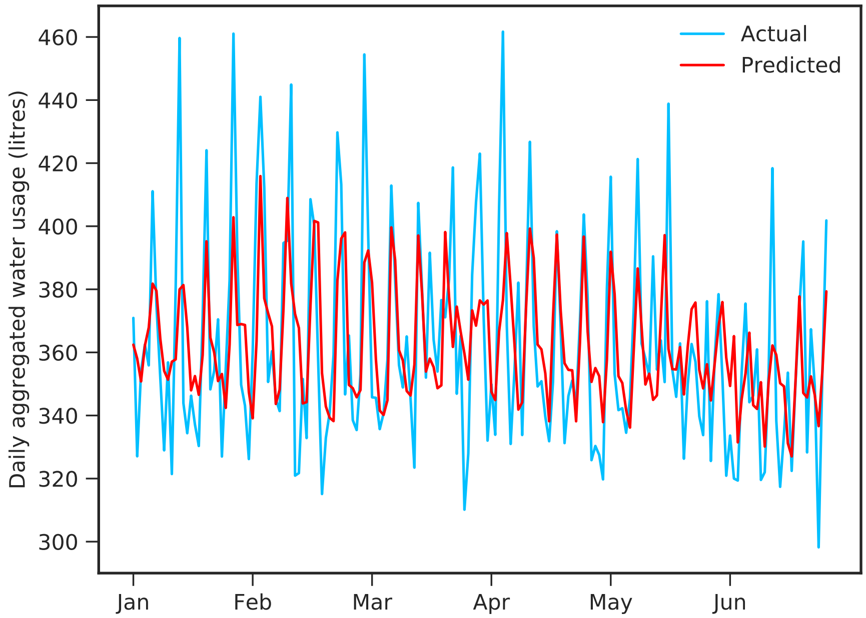


Figure 3.3: Actual vs. predicted water usage using SVR model with selected features for DA 59090068

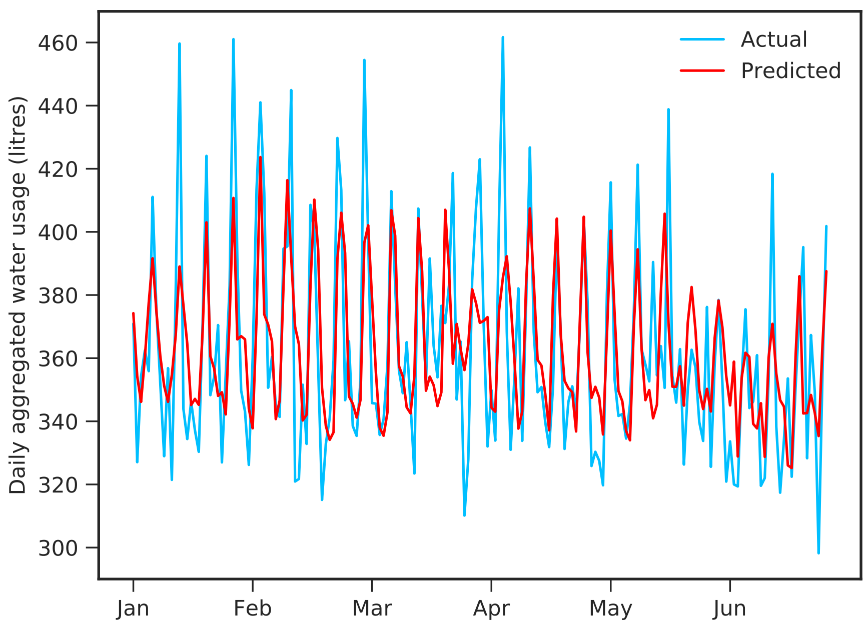


Figure 3.4: Actual vs. predicted water usage using SVR model with Friday and Weekend features for DA 59090068

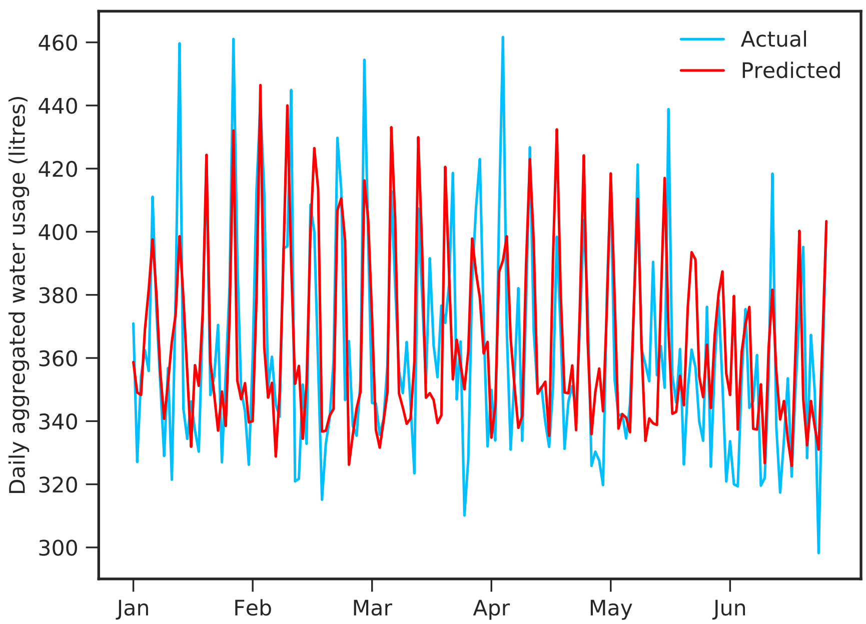


Figure 3.5: Actual vs. predicted water usage using SVR model with features to capture peaks and troughs for DA 59090068

3.13 Explanation

In this section, we conduct an ordinary least squares regression analysis and examine Pearson and Spearman correlation coefficients to explain the determinants of daily water consumption for multi-family residences at the dissemination area level for the city of Abbotsford. It is helpful to investigate the determinants of water consumption for multi-family residences separately since multi-family residences are distinct from single-family residences in many aspects. In particular, multi-family residences have smaller lots and tend to have less landscaping. Investigating the determinants of water consumption for multi-family residences is useful for water utilities when optimizing water supply and for developing water conservation programs [22].

3.13.1 Ordinary Least Squares Regression

There are four ordinary least squares assumptions as described in Watson and Stock [60]. Under the ordinary least squares assumptions, the estimated coefficients of the linear regression model are unbiased and consistent estimators of the true population coefficients. In our analysis, only two of the four assumptions hold, which is that the conditional distribution of the residuals has a mean of zero and that there is no perfect multicollinearity. However, not all variables are independently and identically distributed. And variables related to water consumption do not have a finite kurtosis, as large outliers were only removed from the water consumption data if it was known to be an error. Therefore, it should be noted that the results presented in this section are not intended to be reliable. When the ordinary least squares assumptions do not hold, the estimated coefficients of the linear regression model may not be accurate.

While conducting the ordinary least squares regression analysis, we consider the features specified in Table 3.16 of Section 3.12. We drop some features from the analysis due to not satisfying the multicollinearity constraint. Multicollinearity occurs when an independent variable is highly correlated to one or more other independent variables [60]. In practice, it is recommended to remove independent variables which are highly correlated with other independent variables, otherwise, regression coefficients may be estimated imprecisely. In most cases, the magnitudes of the regression coefficients will not make sense or will have the wrong sign. For this analysis, we calculate the pairwise correlations for the features in Table 3.16. We find that all water consumption features are highly positively correlated. Due to not satisfying the multicollinearity constraint, we drop all water consumption features except for *Average of water usage at three, two, and one week ago*, as this feature has the highest pairwise correlation with the target variable. Similarly, we find that the features *Daily temperature* and *Average temperature over the last seven days* are highly positively correlated. We drop *Daily temperature* and keep *Average temperature over the last seven days*, as the latter feature has a higher pairwise correlation with the target variable. Equation (1) expresses the ordinary least squares regression for daily water consumption per unit at the dissemination area level. The model is built over all 30 dissemination areas. We obtain an R^2 value of 0.8751 for equation (1). This means that the model explains 87.51% of the variability of the target variable *DailyWaterConsumption*. The adjusted R^2 value is roughly the same.

$$\begin{aligned} DailyWaterConsumption = 10.32 &+ (0.95 \times AvgLast3Weeks) \\ &+ (0.84 \times AvgTempLast7Days) \\ &+ (5.66 \times WeekendHoliday) \\ &+ (1.05 \times Friday) \end{aligned} \tag{1}$$

The variables of equation (1) are as follows: *DailyWaterConsumption* is the daily water consumption per unit at the dissemination area level (litres). *AvgLast3Weeks* is the average daily water usage per unit at three, two, and one week ago (litres). *AvgTempLast7Days* is the average daily temperature over the last seven days (°C). *WeekendHoliday* is set to true if the day is a weekend or a holiday (binary). *Friday* is set to true if the day is a Friday (binary).

Equation (1) shows that as the average daily water usage per unit at three, two, and one week ago increases, daily water consumption per unit will increase at roughly the same rate. An increase in the average daily temperature over the last seven days by one degree celsius will increase daily water consumption per unit by roughly 0.84 litres. This positive relationship is in line with the current literature that suggests that increased temperatures bring about higher water usage [22], [32]. If the day is a weekend or holiday, equation (1) suggests that daily water consumption per unit will increase by roughly 5.66 litres. Increased water usage during the weekend and on holidays is expected. Typically, people are at home more often during the weekends and holidays, which leads to increased water consumption. If the day is a Friday, equation (1) suggests that daily water consumption per unit will increase by roughly 1.05. The positive coefficient on the *Friday* variable is not what we expect, as visually inspecting the data reveals that Friday water consumption tends to be lower on average compared to other days of the week. Therefore, we would expect the coefficient to have a negative sign. This is likely the result of not all ordinary least squares assumptions holding. As stated by Watson and Stock [60], the estimated coefficients of a linear regression model may not be reliable if the ordinary least squares assumptions do not hold. Later, we show that the *Friday* variable has a negative Pearson correlation and a negative Spearman correlation. The assumptions for Pearson and Spearman hold more strongly than with ordinary least squares for this problem.

3.13.2 Pearson and Spearman Correlation Coefficient

Next, we look at the Pearson correlation coefficient (r) and Spearman rank correlation coefficient (ρ) for each feature against the target variable. The Pearson correlation coefficient is a measure of the strength of the linear relationship between two variables. It is subject to assumptions that must hold in the underlying data, as specified in [27]. For the dependent variables in Table 3.16 that we examine in this analysis, most do not hold for three of the four assumptions. Not all variables are continuous measurements, some dependent variables do not appear to have a linear relationship with the independent variable, and none of the variables are approximately normally distributed. However, it has been shown that the Pearson correlation is robust to violations of these assumptions [27], [44]. On the other hand, Spearman's rank correlation coefficient measures the strength of the monotonic relationship between two variables. Unlike the Pearson correlation coefficient, it is subject to a fewer number of assumptions that must hold in the underlying data. We find that the assumptions for Spearman's rank correlation hold for the data in this analysis, except for the binary variables. Ideally, the variables should be continuous or ordinal measurements. We calculate the Pearson and Spearman correlation for the nine features in Table 3.16. Table 3.17 shows the results of the Pearson and Spearman correlations for each feature.

All water consumption features have a high positive correlation with daily water consumption per unit, as shown in the scatterplots depicted in Figure 3.6. Both temperature features have a small positive correlation with daily water consumption per unit. In general, we expect that features related to temperature would have a smaller positive correlation with daily water consumption for multi-family residences compared to single-family residences, as multi-family residences tend to have smaller lots and less landscaping, reducing outdoor water usage. In addition, we find that climate features that were not selected in the feature selection step, such as barometer, wind speed, and rainfall, have very small negative correlations with the target variable. In general, climate features do not appear to have a significant effect on water consumption for multi-family residences. *Weekend or holiday* has a small positive correlation with daily water consumption per unit. We suspect that the correlation is only a small positive correlation since weekend usage is higher than average for only Saturday or Sunday, and not both, upon visual inspection of the data. In Figure 3.7, the boxplots show that weekend or holiday water consumption is generally higher in terms of the 5th, 25th, 50th, 75th, and 95th percentiles, compared to consumption that does not occur on the weekend or holiday. The *Friday* variable has a small negative correlation with daily water consumption per unit. In general, Friday water usage is slightly less than the other days of the week, so this result is expected. The boxplots in Figure 3.7 show that Friday water consumption is generally lower in terms of percentiles, compared to consumption that does not occur on Friday.

Table 3.17: Pearson and Spearman correlation coefficients for features against daily water consumption

Feature	$\boldsymbol{r}$	$\boldsymbol{\rho}$
Water usage three weeks ago	0.89	0.87
Water usage two weeks ago	0.91	0.88
Water usage one week ago	0.91	0.89
Average of water usage at three, two, and one week ago	0.93	0.91
Average of water usage at three, two, and one day ago	0.91	0.91
Daily temperature	0.13	0.11
Average temperature over last seven days	0.13	0.12
Weekend or holiday	0.06	0.08
Friday	−0.03	−0.04

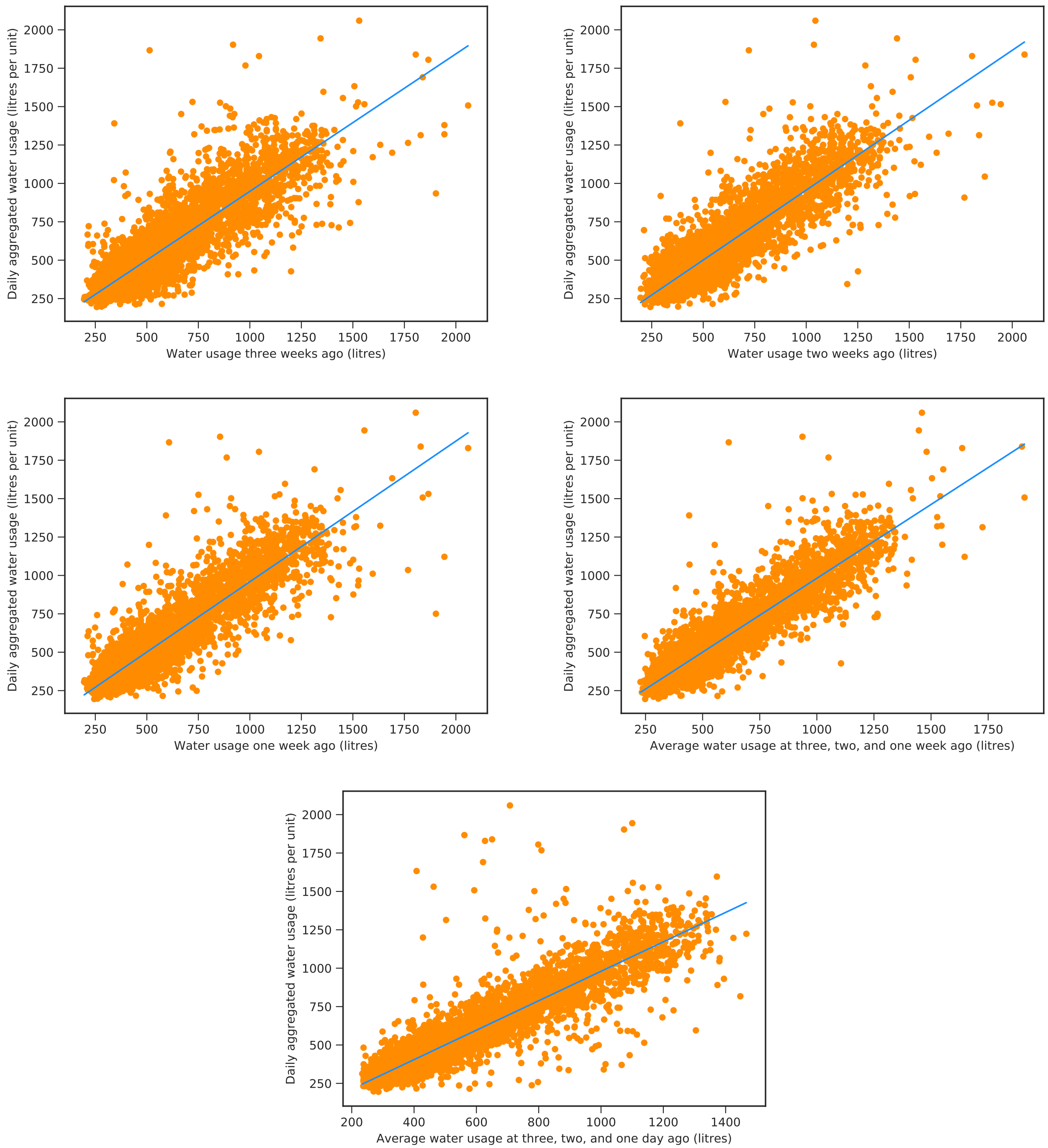


Figure 3.6: Correlation of water features vs. daily water usage

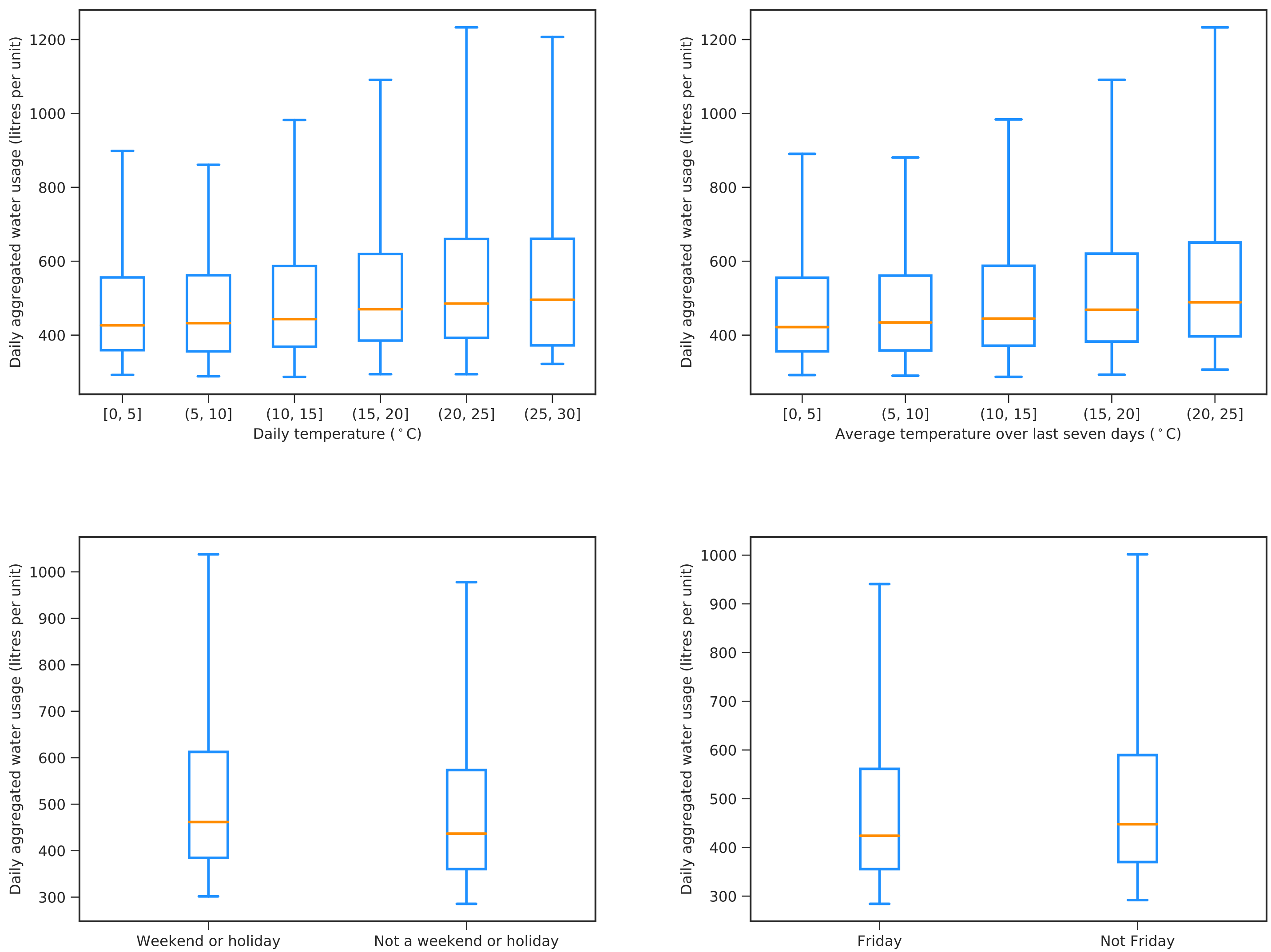


Figure 3.7: Variance of model features vs. daily water usage

3.14 Conclusion and Future Work

In this chapter, we built a machine learning model which predicts daily water consumption for multi-family residences in the city of Abbotsford. The model will be applicable to the city of Abbotsford, but the methodology for building the model can also serve as a template for other cities given the availability of data. Over the short-term, water consumption predictions are useful in the daily operation and management of water supply systems. In particular, short-term water consumption forecasts are used to optimize well, pump, main, and reservoir operations, to properly balance the allocation and distribution of water, to estimate revenue and expenditures in the short-term, and to develop short-term water demand strategies, with the goal of providing reliable and adequate water supply to consumers at reasonable volumes [54], [30]. This work addresses the gap in the research for predicting water consumption for multi-family residences and accounts for the current shift towards denser living spaces in urban areas. In terms of future work, the accuracy of our models can be further improved by collecting and training across more data.

Chapter 4

Urban Planning Models

4.1 Introduction

In this chapter, we introduce a new methodology for building machine learning models to predict daily water consumption for new developments at the dissemination area level. Although the model we build using this new methodology is for the city of Abbotsford, this methodology can also serve as a template for other cities which have the appropriate data at hand. A model to predict water consumption for new developments could be particularly useful to cities which are undergoing rapid growth. Over the next few decades, water infrastructure will expand to accommodate continued growth in urban areas, particularly in developing countries [42]. Predicting water consumption for future developments is useful for determining the appropriate water infrastructure to serve a new development and can be used to determine the general capacity of new water infrastructure [18]. It should be noted that we are building models to predict daily water consumption for new developments which are currently in the planning stage, where properties and water infrastructure to serve these properties are yet to be built.

4.2 Industry Baseline

In this section, we discuss the current industry baselines for predicting water consumption for new housing developments. In the industry, a typical baseline that is used consists of taking an estimate of the population that the utility expects to serve in a new development and multiplying it by the per capita water consumption of existing developments, or some variant of this simple calculation [6], [29], [54]. The water usage for a new development is predicted by using this baseline. Since the current baseline used in industry is very simplistic, there is much room for improvement. The accuracy of predictions can improve by building models of greater complexity, which we will show in this chapter.

Improving the accuracy of water consumption predictions for new developments can potentially save water utilities valuable time and resources. In general, per capita water consumption forecasts have a tendency to consistently overestimate water demand. This has occurred in numerous American cities, such as Seattle, Washington DC, San Diego, and Phoenix [29]. The reasoning behind this overestimation is due to the per capita water consumption calculation accounting for an increasing population in these cities. In reality, actual water usage per capita has declined in these cities. As discussed previously, one of the factors leading to this decline is due to the introduction of denser living spaces in urban areas. Multi-family residences tend to have smaller lots, less landscaping, and fewer pools, which pushes down water consumption. Per capita water consumption forecasts do not take into account land use and housing density. One of the drawbacks of overestimating water consumption is the unnecessary investment in infrastructure, which can cost cities millions of dollars. More critically, underestimating water usage could potentially lead to water supply shortages, higher costs to consumers, and can lead to emergency conservation procedures [29]. In the case of Abbotsford, the city is under a lack of infrastructure and currently enacts water conservation measures in the summer months to reduce peak water demand [64]. These water conservation measures have been in place since 1995 in order to prevent the expansion of water infrastructure in the city, which carries significant costs [16]. Overall, obtaining accurate water consumption predictions will lead to developing water infrastructure which will more appropriately serve the needs of a new development.

In our case, the average and median baselines that we calculate for the city of Abbotsford, which we describe in a later section, has a tendency to widely overestimate or underestimate the per capita water usage of a dissemination area. In contrast, the model we build later in this chapter produces predictions which improve on the error and have less tendency to overestimate or underestimate per capita water consumption.

4.3 Related Work

To our knowledge, models to predict water consumption for new developments have not been attempted before in the existing literature. There is much work on predicting water consumption for existing developments, but none on predicting water consumption for new developments. This is likely due to the difficulty of the problem or the lack of data available. For this problem, as described in more detail later in the chapter, we are limited to climate, property, and demographic features. The model does not use previous water consumption features because this knowledge is not known for new developments. In the related work found in Table 3.2, which looks at forecasting water consumption for existing developments, it has been shown that water consumption features contribute significantly to predictive accuracy compared to climate, property, and demographic features. Previous water consumption values, in the form of statistical values or lagged values, are key features in models which forecast water consumption for existing developments. The difficulty with predicting water consumption for new developments is that we do not have the key water consumption features available to us. Therefore, obtaining accurate predictions will be difficult. In this chapter, the focus will be on improving upon the baselines that are currently used in the industry.

The closest related work pertains to investigating the determinants of water consumption for new apartment properties in Stevenage, UK [18]. Other related work looks at the determinants of water consumption for multi-family residences, as outlined in Table 3.1 of Section 3.2. We are unable to find any work that looks at developing models to predict water consumption for new developments for any type of customer sector, including multi-family residences.

In this chapter, we introduce a new methodology for building machine learning models which predict water consumption for new housing developments. This new methodology is similar to the methodology described earlier in Chapter 3, although there are some notable distinctions, particularly in the feature selection process. We apply this new methodology to build models for the city of Abbotsford, which can also be applied to other cities with the appropriate data. In the following sections, we refer to these machine learning models as urban planning models, as they predict water consumption for new developments. The theme of this thesis focuses on multi-family residences, so we only build models pertaining to multi-family residences. In the future, the same methodology can be applied to other customer sectors as well, such as single-family residences, commercial, and institutional properties. We describe this later in Section 4.13.1.

4.4 Problem Statement

In this chapter, we predict daily water consumption for new multi-family residence developments using water consumption data from the period between September 1, 2012 to August 31, 2013 for the city of Abbotsford. We are using the methodology described in this chapter to build a single machine learning model to predict the daily water consumption per unit at the dissemination area level, where the dissemination area to predict is assumed to be a new development. The inputs of the model are those specified later in Section 4.7. The output of the model is the predicted daily water consumption per unit for a dissemination area under a new development. For each dissemination area under an existing development, we use the daily aggregated water consumption for the period between September 1, 2012 to August 31, 2013 as the target variable. We concatenate the water consumption data for each dissemination area into a single training dataset. Table 4.1 outlines the model built in this chapter. Overall, the main objective of this chapter is to build a machine learning model to predict daily water consumption per unit at the dissemination area level for a new development, while improving over a standard industry baseline. In addition, a new methodology is provided as a guide for building these models, which is discussed later.

Table 4.1: Model characteristics for predicting daily water consumption of multi-family residences in new housing developments

Single or multiple models?	Single
Model input(s)	Selected features from Section 4.7
Model output(s)	Predicted daily water consumption per unit for a dissemination area, where the dissemination area to predict is assumed to be a new development. Per unit water consumption is aggregated at the dissemination area level.
Temporal scale	Daily
Spatial scale	Dissemination area
Train dataset	Target variable: Daily aggregated water consumption data from September 1, 2012 to August 31, 2013 for each dissemination area of existing developments. Data for each dissemination area is concatenated into a single training dataset.
Test dataset	Target variable: Daily aggregated water consumption data from September 1, 2012 to August 31, 2013 for a particular dissemination area assumed to be a new development.

It should be noted that in actuality, we do not have data for actual new developments in the city of Abbotsford. We assume that a particular existing development is a new development in order to properly calculate a test error and evaluate our models, as these require both actual and predicted values. For the test set during cross-validation, we use a dissemination area where we have the actual water consumption and assume that this dissemination area falls under a new development. We obtain the predicted values for the test dissemination area using our model and obtain the test error by comparing the actual and predicted values. In practice, however, we are concerned with predicting the daily water consumption for a new development. The actual water consumption for new developments is unknown and is what we are interested in predicting.

4.5 Data Preprocessing

In this chapter, we use the same datasets as described in Chapter 3, after cleaning and preprocessing the data. We use water consumption data for existing developments in the city of Abbotsford from September 1, 2012 to August 31, 2013, property assessment data from BCAssessment, demographic data from the 2011 National Household Survey, and climate data for the city of Abbotsford.

4.6 Feature Engineering

For the urban planning models we build in this chapter, we start off with the same engineered features as described in Chapter 3. In particular, we are only concerned with calendar, climate, property, and demographic features for training our models, as specified in Table 3.8, 3.9, 3.10, and 3.11, respectively, as previous water consumption data is not known for new housing developments.

Since property and demographic features are only available from a single snapshot in time from when the data was collected and stay relatively fixed throughout the year, property and demographic features are aggregated at the dissemination area level. Each dissemination area will have only one particular value for a property or demographic feature. For example, throughout the period between September 1, 2012 to August 31, 2013, a particular dissemination area will have an average family size of 2.0, while another dissemination area will have an average family size of 2.5. In total, we study 24 dissemination areas, so each property or demographic feature will only have 24 different values, one for each dissemination area. In contrast, the climate features have been collected from September 1, 2012 to August 31, 2013 at a daily rate. Therefore, there will be 365 different values for climate features throughout this time period.

4.7 Feature Selection

For the feature selection step, we take a different approach compared to what was done in Chapter 3, where we used a wrapper method called recursive feature elimination with cross-validation (RFECV) to conduct feature selection. Due to the small size of the dataset and the limited number of values for the property and demographic features, which is limited to one value per dissemination area, feature selection methods such as RFECV do not return consistent results across multiple runs. This is also the case for filter and embedded feature selection methods, such as univariate feature selection, Lasso, and using tree-based feature importances. Since we are unable to obtain additional data to get feature selection methods to return consistent results, we instead look at the current literature which investigates the determinants of water consumption for multi-family residences. We use the significant determinants mentioned in these papers as the features of our urban planning model.

Although there is not much work in the current literature which investigates the determinants of water consumption for multi-family residences, we did find a few relevant papers as outlined in Table 3.1. We looked at each of the papers outlined in Table 3.1 and selected features based on those found to be statistically significant determinants of water consumption (with at least 10% significance) and features which were present in our datasets. Many papers mention a pool and a lot size feature as determinants of water consumption for multi-family residences, but we are unable to use these features in our models since these features are not present in our dataset. The property data which was provided by BCAssessment only included pool and lot size information for half the properties in our multi-family residence dataset. In addition, the pool size feature is not provided in the BCAssessment property assessment tool, although an approximation of the lot size is, in the form of the square footage of a unit excluding outside space. However, we are unable to include a lot size feature as we found the approximation of the lot size to be inconsistent in several cases. For some properties, the lot size value appeared to exclusively measure the indoor unit size, for other properties, the lot size value appeared to be measuring the full lot size of a multiple residence or strata apartment and did not record the lot size per unit. After conducting a thorough literature review, we investigate the following features to use in our urban planning models, as specified in Table 4.2.

Table 4.2: Selected features from literature review

Feature
Daily temperature
Daily rainfall
Average family size
Median household income ($)
Duplex or townhouse %
Average number of bedrooms
Average year built

The following describes the selected features in more detail: In [22], average air temperature was found to be a significant determinant of water consumption for multi-family residences. They found that higher temperatures led to increased water usage. In the study by Fox et al. [18], they look at the architectural type of multi-family residences. It was found that certain architectural types such as detached residences tend to consume more water. We use the *Duplex or townhouse %* feature as a proxy for the architectural type feature. Agthe and Billings [3] and Fox et al. [18] mention the number of bedrooms as a significant determinant of water consumption for multi-family residences. The number of bedrooms tends to serve as a proxy for the occupancy of a household, so a larger number of bedrooms typically results in increased water usage. The year a property is built was found to be a significant determinant of water consumption in [3], [13], and [35]. In general, older properties tend to consume larger amounts of water as these properties are subject to degrading infrastructure, resulting in more water leakages. In addition, water efficient appliances were only introduced during the 1980s and later. Kontokosta and Jain [35] and Wentz and Gober [66] found household size to be a significant determinant of water consumption for multi-family residences. They found that a positive relationship exists between household size and water consumption. We use the *Average family size* feature as a proxy for household size. In addition, median household income was found to be a significant determinant of water consumption in [35]. Here, they find that median household income is negatively related to water consumption. However, in other studies, they find income to be positively related to water consumption or not a significant determinant of water consumption [56], [31]. Finally, rainfall is commonly mentioned as a determinant of water consumption [19], although the analysis in [22] does not find it to be a significant determinant of water consumption.

After investigating the features in Table 4.2 in more detail, we drop the features which do not contribute to improving the accuracy of our models. In particular, we drop the features *Daily rainfall*, *Average year built*, and *Median household income*. The Pearson and Spearman correlation coefficient for these features are nearly zero, which means these features have close to no linear or monotonic relationship with the target variable: daily water consumption per unit. This result is expected as rainfall is not a significant feature for multi-family residences, as multi-family residences tend to have smaller lots and less landscaping compared to single-family residences. The average year built for multi-family households in Abbotsford is clustered around and before the 1980s, before water saving appliances were introduced, which explains the low Pearson and Spearman coefficients for the *Average year built* feature. Based on its low Pearson and Spearman coefficients with the target variable, *Median household income* is not a useful feature for predicting water consumption for multi-family residences. In addition, we find that performance does not improve (it stays roughly the same) when the features *Daily rainfall*, *Average year built*, and *Median household income* are included in a linear regression model. This leaves four features which we use in our urban planning models, as specified in Table 4.3.

Table 4.3: Urban planning model features

Feature
Daily temperature
Average family size
Duplex or townhouse %
Average number of bedrooms

Next, we discuss the practicality of the selected features. The inputs to our urban planning models should be reasonable to obtain. In [15], one of the primary concerns is the practicality and ease of obtaining the features used in water consumption forecasting models. For the *Daily temperature* feature, historical daily temperature can be obtained from a variety of sources. For the features *Duplex or townhouse %* and *Average number of bedrooms*, urban planners will have information regarding the percentage of buildings which are duplexes and townhouses, as well as the average number of bedrooms in a dissemination area. In general, property owners will have a good estimate for the *Average family size* of a dissemination area, as they are aware of the market and the target demographic for their properties. In addition, estimates of average family size can also be obtained by looking at the size of each property as well as neighbourhood characteristics. Some properties may be suited towards couples with children, while others may be more suited towards couples without children.

4.8 Baseline Models

In this section, we define the baseline models used to compare our urban planning models to. Previously, we described the baseline models currently used in the industry, as specified in Section 4.2. A common baseline mentioned in the literature is calculated by taking an estimate of the population that the utility expects to serve and multiplying it by the per capita water consumption [6], [29], [54]. To calculate this baseline, water utilities must estimate the population and the per capita water consumption of the area. Since these simple models are not costly for water utilities to develop, they are typically used as baselines to compare with other methods. Here, we calculate several different baseline models. Some baseline models perform better than others for our data. Later, we will compare the machine learning models that we build to our best baseline model as described in this section. We do not directly compare our machine learning models to the baselines used in [6], [29], and [54] since the baselines in these papers are specific to the cities that they are applied to, have a different spatial and temporal scale, and do not explicitly report performance measures. Therefore, a direct and fair comparison cannot be made with these baselines.

Since we are predicting water consumption at the dissemination area level, we calculate a baseline prediction for each dissemination area. To measure the performance of the baseline model, we calculate the mean absolute error for each dissemination area by comparing the baseline prediction with the actual consumption. Next, we average out the mean absolute errors across all dissemination areas to get a final performance value. The baselines are calculated over the same 24 dissemination areas used to train our machine learning models.

Here, we explain in detail how each of our baseline models is calculated. We note that each baseline calculates the baseline usage for one dissemination area. The first baseline is expressed in equation (1). We obtain the average family size from the 2011 National Household Survey for each dissemination area. The average family size is defined as the average number of occupants per unit across a particular dissemination area for the year 2011. Since we do not have exact population numbers, as used in the common industry baseline, we use the average family size as a proxy. We obtain the per capita average daily water usage from the 2013 AMWSC Water Demand Projections report [57]. This value is defined as the average daily water usage per person in the city of Abbotsford for the years 2011 to 2012.

$$Water\ usage\ per\ unit\ of\ a\ DA = Average\ family\ size\ of\ a\ DA \\ \times\ Average\ daily\ water\ usage\ per\ person \tag{1}$$

The second baseline, expressed in (2), takes into account the vacancy rate in Abbotsford. We obtained the 2012 and 2013 Abbotsford vacancy rates from [43]. We use the vacancy rate to estimate the number of occupied units of a dissemination area by taking the total number of units in a dissemination area and multiplying it by the vacancy rate.

$$Water\ usage\ per\ unit\ of\ a\ DA = \frac{\begin{array}{l}(Average\ family\ size\ of\ a\ DA \\ \times Average\ daily\ water\ usage\ per\ person \\ \times Number\ of\ occupied\ units\ of\ a\ DA)\end{array}}{Total\ number\ of\ units\ in\ a\ DA} \tag{2}$$

The third baseline (3) is calculated for a dissemination area by taking the average daily water usage per unit across all dissemination areas except the dissemination area being predicted for, in the period between September 1, 2012 to August 31, 2013. In the calculation, we do not include water usage values for the dissemination area being predicted for since this information is not known for a new development and including it would result in test leakage. Here, we assume that the dissemination area that we are predicting for is a new development.

Similarly, the fourth baseline (4) is calculated for a dissemination area by taking the median daily water usage per unit across all dissemination areas except the dissemination area being predicted for, during the period between September 1, 2012 to August 31, 2013.

Finally, the fifth baseline (5) is calculated for a dissemination area by taking the median daily water usage per unit for weekdays only, across all dissemination areas except the dissemination area being predicted for, in the period between September 1, 2012 to August 31, 2013.

The performance of each baseline is outlined in Table 4.4. We can see that baseline (5) is the best performing baseline with a test MAE of 164.79. This means that on average, the difference between the actual and predicted usage for a new development at the dissemination area level is 164.79 litres. This makes up 27.83% of the average daily water usage in Abbotsford of 592 litres per day per unit. Baseline (5) is the baseline that we compare our machine learning models to, which we build in the next section.

Table 4.4: Baseline performance

Baseline	Test MAE (litres)
(1)	173.91
(2)	171.78
(3)	174.78
(4)	165.04
(5)	164.79

4.9 Grid Search

In this section, we train various machine learning models to predict water consumption for new developments using the selected features from Section 4.7. With default model parameters, we obtain test results which see a small improvement from the baseline to performance which fares much worse compared to the baseline. To improve performance, we conduct the same grid search approach as specified in Section 3.10 of Chapter 3. To determine the optimal parameters for each model, we perform a series of coarse grid searches followed by a series of fine grid searches. We conduct grid searches until we are able to narrow down to an optimal set of parameters. Table 4.5 reports the optimal parameters for each model and are described in [48]. Table 4.6 shows the results after conducting a grid search.

Table 4.5: Grid searched parameters for urban planning models

Model	Optimal parameters
Linear Regression	-
KNN	n_neighbors: 800 p: 2 weights: "uniform"
LinearSVR	C: 10 epsilon: 0.0001 loss: "epsilon_insensitive"
SVR	C: 1 epsilon: 0.01 gamma: 0.0001 kernel: "rbf"

Neural network with one hidden layer	activation: "logistic" alpha: 0.00001 batch_size: 200 hidden_layer_sizes: (2) learning_rate_init: 0.001 solver: "adam"
Neural network with two hidden layers	activation: "logistic" alpha: 0.001 batch_size: 200 hidden_layer_sizes: (4, 2) learning_rate_init: 0.001 solver: "adam"
Decision Tree	criterion: "mse" max_depth: 100 max_features: None min_samples_leaf: 400 splitter: "best"
Random Forest	criterion: "mse" max_depth: 50 max_features: None min_samples_leaf: 300 n_estimators: 10
AdaBoost	learning_rate: 0.1 loss: "exponential" n_estimators: 50
Gradient Boosting	criterion: "mse" loss: "lad" max_depth: 50 max_features: "sqrt" min_samples_leaf: 1000 n_estimators: 100 subsample: 0.6

Table 4.6: Urban planning model performance using grid searched parameters and baseline performance

Model	Test MAE (litres)
Neural network with two hidden layers	111.48
SVR	129.79
LinearSVR	131.46
Gradient Boosting	135.49
KNN	135.77
Decision Tree	137.11
Random Forest	138.77
Neural network with one hidden layer	139.77
Linear Regression	149.24
AdaBoost	155.92
Baseline (5)	164.79
Baseline (4)	165.04
Baseline (2)	171.78
Baseline (1)	173.91
Baseline (3)	174.78

4.10 Model Performance

After conducting a grid search, the performance of all machine learning models improve. All models have superior performance compared to the best baseline model. For all models, the optimal parameters are robust in that a wide range of values for each parameter still lead to improved performance over the baseline. We obtain the best performance using a neural network with two hidden layers, which sees a 32.35% improvement over baseline (5). Given the difficulty of the problem, a 32.35% improvement over the baseline appears to be a significant improvement. It should be noted that water utilities need to take into account whether the performance improvement warrants the extra cost associated with building more complex models. They need to determine whether the performance improvement over the baseline is worth the time and cost of developing models of higher complexity. Water utilities need the appropriate data and the technical expertise to build models of this complexity.

4.11 Model Predictions

In this section, we plot the actual water usage, predictions from our neural network model, and predictions from the best baseline model from September 1, 2012 to August 31, 2013 for a specific dissemination area. The dissemination area we plot (DAUID: 59090064) has 1014 multi-family units and has fairly consistent usage throughout the year with no large outliers. To easily see the seasonal patterns, the data is ordered starting in January and ending in December. The neural network model is trained over 29 dissemination areas, excluding the dissemination area we are plotting. We use the four selected features and the grid searched parameters as described in the previous sections. We plot the baseline predictions using baseline (5). In Figure 4.1, we see that the neural network predictions are able to more closely match the actual usage compared to the baseline prediction. In addition, the neural network model is able to capture the seasonality of the data, where there tends to be more usage in the summer months compared to the winter months. For this dissemination area, with the neural network model, as with the baseline model, there is a tendency to overestimate daily usage. However, the overestimation is less severe with the neural network model.

We have presented a new methodology for building machine learning models to predict daily water consumption for new developments at the dissemination area level. This methodology has been applied to the city of Abbotsford, but can also serve as a template for other cities which have the appropriate water consumption, climate, property, and demographic data. To predict water consumption for a new development in another city, the analyst would conduct the steps outlined in this chapter: starting with feature engineering, then, selecting features to use by conducting a literature review or using the features selected in this chapter, conducting a grid search to obtain optimal model parameters, comparing model performance to existing baseline models, and finally building and deploying a machine learning model; training on data collected from existing developments in the city. The analyst can go a step further and investigate the features of their machine learning model in more detail, which we conduct in the next section.

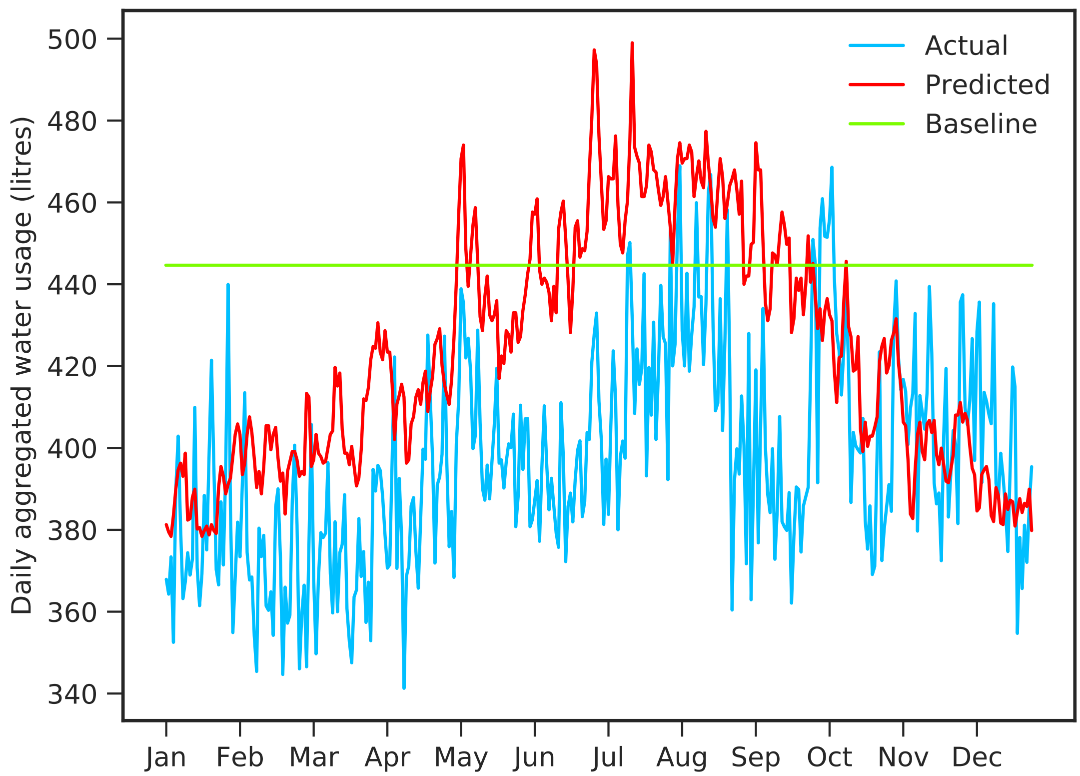


Figure 4.1: Actual vs. predicted water usage using neural network model with selected features for DA 59090064 with baseline prediction

4.12 Explanation

In this section, we calculate Pearson and Spearman correlation coefficients to examine the features of our urban planning models in more detail. It is helpful to examine the features of our models as it gives us a better indication as to why these features are useful for predicting water consumption for new developments.

We calculate the Pearson correlation coefficient (r) and the Spearman rank correlation coefficient (ρ) of each feature against the target variable: daily water consumption per unit. The Pearson correlation coefficient measures the strength of the linear relationship between two variables and is subject to assumptions that must hold on the underlying data [27]. For the data in this analysis, some dependent variables do not appear to have a linear relationship with the independent variable, and no variables are approximately normally distributed. However, it has been shown that the Pearson correlation is robust to violations of its assumptions [27], [44]. The Spearman rank correlation coefficient measures the strength of the monotonic relationship between two variables. It is also subject to assumptions in the underlying data. For the data studied in this analysis, all assumptions for Spearman rank correlation hold. Table 4.7 shows the Pearson and Spearman correlations for each feature.

Table 4.7: Pearson and Spearman correlation coefficients for features against daily water consumption

Feature	r	ρ
Daily temperature	0.13	0.11
Average family size	0.35	0.49
Duplex or townhouse %	0.25	0.25
Average number of bedrooms	0.57	0.56

As shown in Table 4.7, *Daily temperature* has a low positive correlation with daily water consumption per unit. In general, this is an expected result since multi-family residences tend to have smaller lots and less landscaping, leading to less outdoor water usage which is highly tied to variations in temperature. In Chapter 3, Figure 3.7 shows the boxplots for *Daily temperature*, where each boxplot shows the 5th, 25th, 50th, 75th, and 95th percentiles of daily water consumption for a particular temperature range. *Average family size* has a positive correlation with daily water consumption per unit, as shown in the boxplots of Figure 4.2. This result is expected as larger family sizes tend to consume more water. *Duplex or townhouse %* has a positive relationship with daily water consumption per unit. Since duplexes and townhouses in Abbotsford tend to have larger lots and more landscaping compared to multiple residences and strata apartments, this positive correlation is expected. Figure 4.3 shows the boxplots for *Duplex or townhouse %*. In addition, we found that *Multiple residence or strata apartment %* has a negative correlation with daily water consumption per unit. There is a positive correlation between *Average number of bedrooms* and daily water consumption per unit. We can see this positive correlation in the boxplots depicted in Figure 4.4. This result is expected as the average number of bedrooms tends to serve as a proxy for family size. In addition, we found that duplexes and townhouses had a slightly greater number of bedrooms on average, compared to multiple residences and strata apartments. As mentioned previously, duplexes and townhouses tend to have greater water usage due to larger lots and more landscaping.

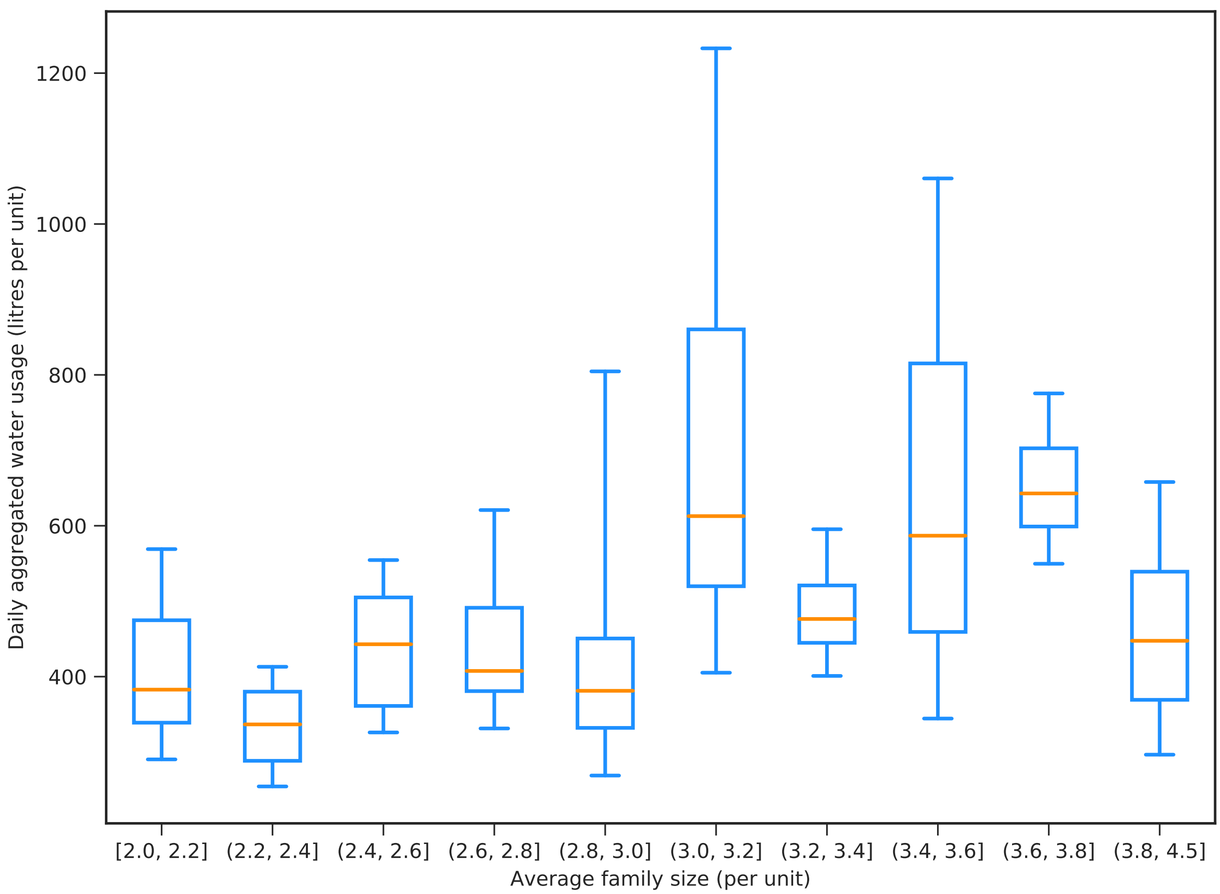


Figure 4.2: Variance of average family size vs. daily water usage

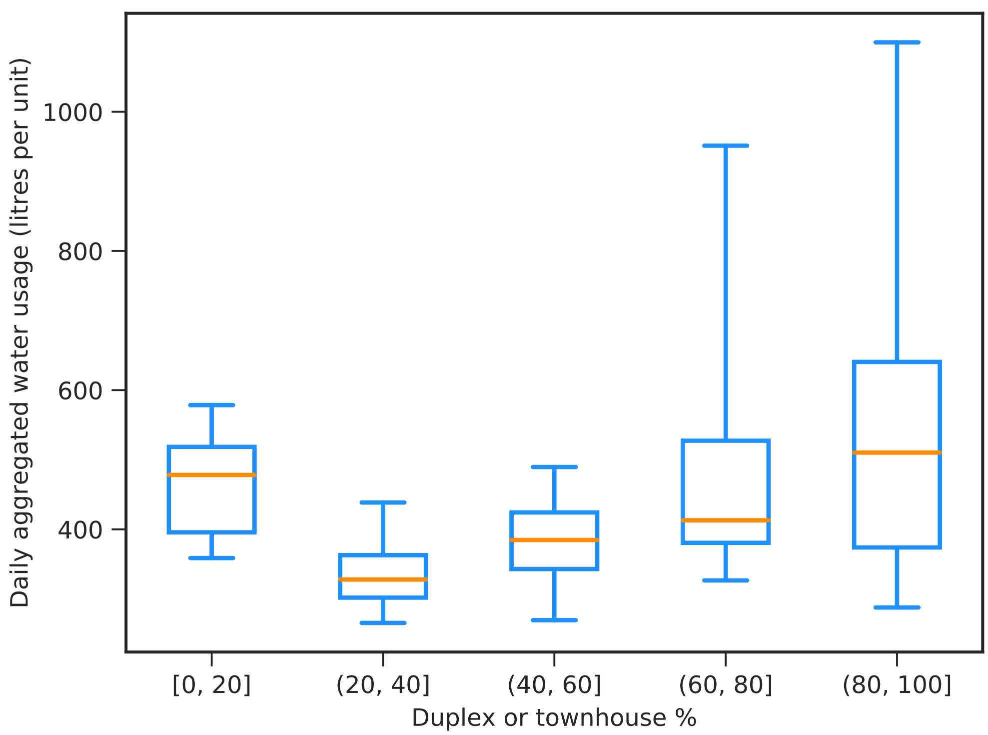


Figure 4.3: Variance of duplex or townhouse % vs. daily water usage

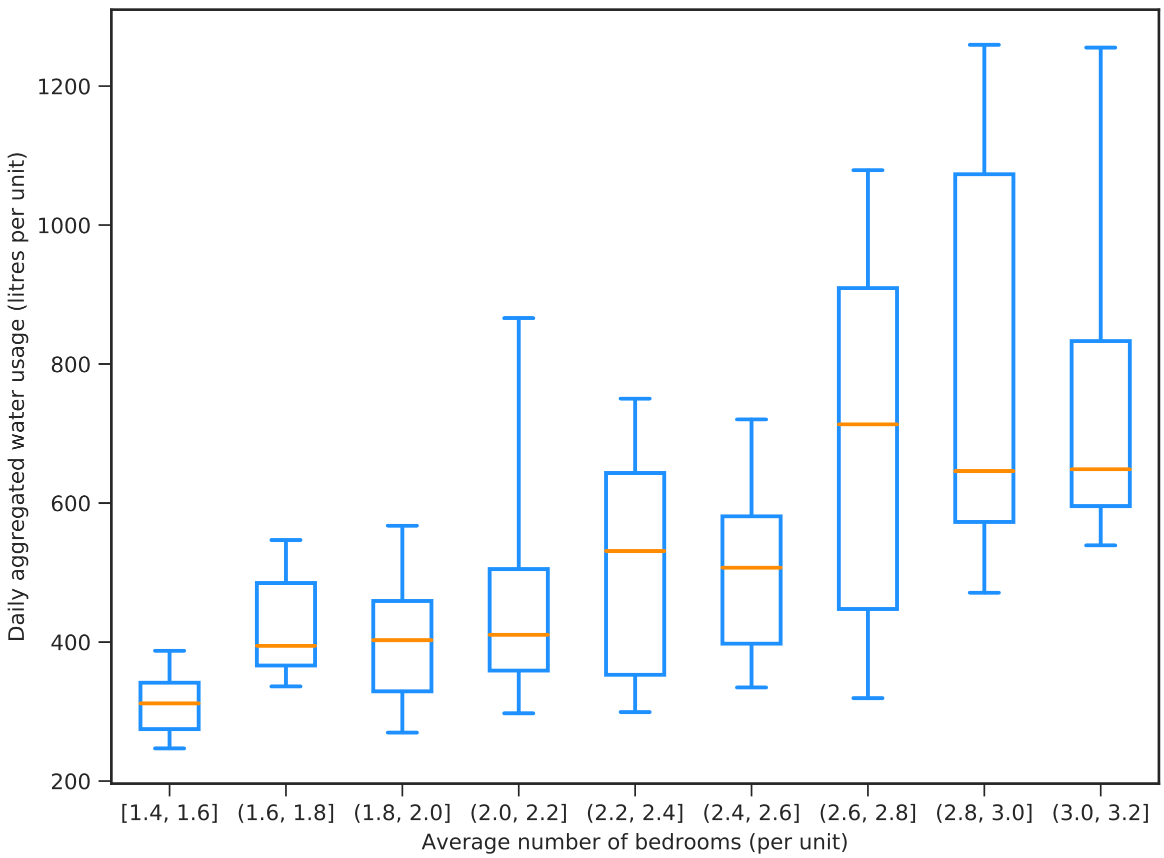


Figure 4.4: Variance of average number of bedrooms vs. daily water usage

4.13 Conclusion and Future Work

In this section, we discuss how to obtain total water consumption predictions for new developments, summarize the contributions of this chapter, and discuss future steps.

4.13.1 Customer Sectors

The following discusses how total water consumption predictions for a new development can be obtained, taking into account all customer sectors in a new development. Since a new development can contain a mix of different customer sectors, such as single-family residences, multi-family residences, commercial, and institutional, a model will need to be built for each sector in order to get an estimate for the total water consumption of a new development. In this chapter, we have shown a new methodology for building a model to predict water consumption for multi-family residences in a new development and are able to achieve improved performance compared to industry baselines. To build models for other customer sectors, the same methodology should also be applicable. To obtain an estimate for the total water consumption of a new development, the forecasts for each customer sector model should be aggregated. Aggregating the forecasts of distinct customer sectors to obtain a total water consumption forecast is a common practice in the industry [6].

Billings and Jones [6] and Heberger et al. [29] discuss the reasoning behind separating water demand customers when building models. The alternative is combining customer sectors into a single water demand model, as is the case with simple per capita water demand models. In general, the accuracy of models is often improved when water demand customers are separated into roughly homogeneous groups according to water use patterns. This is because certain sectors may follow particular patterns of water use, such as the diurnal pattern that is commonly seen in single-family and multi-family residences, where water consumption is heaviest in the mornings and evenings. This pattern is vastly different from the water consumption patterns typically found in the commercial or institutional sector. Water usage levels can also be different for each customer sector. In addition, separating customers allows each customer sector to be analyzed individually. Utility programs and policies will typically require an analyst in the field to examine each customer sector in more detail. For example, the determinants of water consumption for a particular customer sector can be obtained by analyzing the water consumption model for that particular customer sector.

4.13.2 Summary

In this section, we developed a new methodology for building machine learning models to predict water consumption for new developments. This work is particularly important as we are currently seeing rapid growth and development in urban areas. Obtaining accurate predictions of water consumption in new developments informs planners of the appropriate water infrastructure that should be developed in a new area. The current models used in the industry are simple baseline models which tend to grossly overestimate or underestimate water consumption for new developments, leading to costly investments in unnecessary infrastructure or to a lack of water infrastructure, respectively. Using the methodology developed in this chapter, we were able to build a machine learning model with a 32.35% improvement over the best performing baseline model.

In terms of future work, the performance of our machine learning models is likely to improve with additional data. Features which were found not to be useful in the model may also become useful as more data is gathered. Given more time and resources to collect data, the performance of our models may improve with additional features such as pool presence, lot size, and landscaping presence, which have been shown to be significant determinants of water consumption for multi-family residences. At the moment, we do not have this data as obtaining it would be too costly and time consuming. The potential for performance improvement does not seem to warrant the extra cost of obtaining additional data.

Chapter 5

Deep Learning Models

5.1 Introduction

In this chapter, we investigate deep learning models such as recurrent neural networks and convolutional neural networks to predict daily water consumption for multi-family residences at the dissemination area level. The primary advantage of deep learning models is its capability of learning features automatically without the need for a machine learning practitioner to engineer features and perform a feature selection step [5]. In order to do feature engineering, a machine learning practitioner is required to have domain knowledge of the dataset in order to build the most effective features. For feature selection, there are several methods to choose from, ranging from simple techniques such as filter methods to more computationally intensive methods such as wrapper methods. In Chapter 3, we conducted a feature engineering step followed by a feature selection step. These two steps took a considerable amount of time to complete in the model building process. The feature engineering step required a reasonable amount of domain knowledge, which was obtained by studying the data carefully and gaining the background knowledge necessary to get an idea of what features might potentially be useful. A thorough literature review was also conducted to get an idea of the usefulness of features. We used a wrapper method called recursive feature elimination for the feature selection step. This step also took a substantial amount of time since running recursive feature elimination is computationally intensive, particularly for large feature sets such as ours. Overall, conducting feature engineering and feature selection can take a significant amount of time. In this chapter, we investigate the use of deep learning models which are capable of learning data representations to predict water consumption for multi-family residences. The objective is to determine whether we can obtain comparable or improved performance with deep learning models compared to traditional learning techniques which require a feature engineering and selection step. Obtaining comparable or improved performance over traditional machine learning techniques may save water utilities valuable time and resources put towards the model building process, while still achieving a comparable level of performance.

5.2 Terminology

Here, we discuss the terminology used in this chapter. Deep learning models are based on artificial neural networks and are a subset of machine learning. In contrast to traditional machine learning models such as decision trees, which use an algorithmic approach to process and learn from data, deep learning models are capable of learning data representations which eliminate the need for manual feature engineering. Recurrent neural networks and convolutional neural networks are specific types of artificial neural networks and are what we refer to as deep learning models in this thesis. We also examine the use of gated recurrent units and long short-term memory units in recurrent neural networks and refer to these as LSTM and GRU networks. We refer to decision trees, tree-based ensembles, and support vector machines as traditional machine learning approaches, as these methods are not capable of learning features from data. It should be noted that the RNN, GRU, LSTM, and CNN models developed in this chapter are considered to have a shallow architecture, as this architecture is what facilitated the best performance. For simplicity, we still refer to these models as deep learning models as they are capable of learning data representations and can be extended to deeper architectures as more data becomes available.

5.3 Related Work

In the current literature, there has been much effort in engineering the best features for predicting water consumption and in selecting these features to obtain the best performance [2], [1], [23], [30], [45], [63]. In these papers, they use feed-forward neural networks and several variations of artificial neural networks for predicting water consumption. We found this work to be the closest work in the current literature to what we are trying to accomplish in this chapter. The difference is that these particular models still require a feature engineering and a feature selection step, which are both time consuming tasks.

In the related literature, the authors put a significant amount of effort into engineering and selecting features. The work by Adamowski [2] goes into great depth building and comparing several models, each with different features. The purpose behind training many models was to determine the best features to use. In [1], [62], and [63], the authors conduct a thorough feature selection process. They use previous water consumption data and climate variables as features. These features were selected based on a literature review and were found to be the features which influence water consumption. The models are built by iteratively adding features and keeping only the features which improve model performance. In Ghiassi et al. [23], several combinations of lag features for water consumption data are used as model features. The models are trained and compared and the features of the best performing model are outlined. Herrera et al. [30] and Odan et al. [45] perform a literature review to select the appropriate features for water consumption forecasting models. Overall, these methods for selecting features can be both time consuming and costly to water utilities during the model building process.

In the paper by Ghalehkhondabi et al. [21] which reviews the most recent methods for forecasting water consumption in the period between 2005 to 2015, it is noted that no work has been done on using recurrent neural network or convolutional neural network models to conduct water consumption forecasting. On the other hand, recurrent neural networks and convolutional neural networks have been used widely over the last two decades in various fields, such as the energy and financial sectors. Liu et al. [38] and Zhu et al. [69] use time series data to predict wind power output using recurrent neural networks and convolutional neural networks, respectively. In [68], [40], and [28], the authors use recurrent neural networks to forecast electrical load using time series data. In the financial sector, recurrent neural networks have been used widely for time series problems such as stock price prediction, forecasting foreign exchange rates, and determining the future price of commodities [4], [8], [34], [36], [61], [53]. Overall, the results of these papers show that deep learning models such as recurrent neural networks and convolutional neural networks are able to perform well using time series data in these respective fields.

Recurrent neural networks and convolutional neural networks have been shown to be useful in predicting future values for financial time series data since stock prices and exchange rates are noisy, nonlinear, and volatile. Similarly, with electrical load forecasting, the data tends to be nonlinear, non-stationary, and nonseasonal. For this type of data, deep learning models are especially suited since these models are capable of learning nonlinear dependencies in the data. In particular, [34] and [68] show that recurrent neural networks are able to achieve comparable or superior performance compared to traditional forecasting models, such as support vector regression, feed-forward neural networks, and ARIMA. In general, it has been shown that recurrent neural networks can be practically applied to predicting future values in a time series [20]. Similarly, although convolutional neural networks are often used for image recognition problems, Pal and Prakash [47] show that CNNs can be applied to time series forecasting.

Donkor et al. [15] mentions the practicality of water consumption models with regards to the features used in models. In Coomes et al. [12], an ordinary least squares regression analysis is conducted to investigate the determinants of water use. In this study, many features are found to be significant in determining water use. However, it may not be practical for water utilities to obtain many of these features, as the features are obtained from a wide variety of data sources, including household survey data, real estate assessment data, and water billing data. It is also noted that in past work, water consumption forecasting models have included features which can be difficult to calculate, due to being very specific, such as in the case of the work by Goodchild [24]. In this work, features such as *water content in top 0.15m of soil* and *rain minus evapotranspiration* are used. Since recurrent neural networks and convolutional neural networks are capable of learning features, analysts who are building water consumption models do not need to worry about obtaining every relevant data source or engineering the most effective features.

Overall, eliminating the feature engineering and feature selection step can enable water utilities to save a significant amount of time and potentially reduce costs towards the resources they put towards building water consumption forecasting models. This can be achieved by training deep learning models, which are capable of learning features from data.

Ideally, we would like these deep learning models to have comparable or improved performance compared to machine learning models which have been built using a traditional approach with a feature engineering and selection step, such as in Chapter 3. What we gain is a model which is quicker to build and can be built by a practitioner who does not necessarily have a vast amount of domain knowledge of the dataset. In the following sections, we show the methodology for building these models and report the results. To our knowledge, building deep learning models to predict water consumption has not yet been attempted in the existing literature.

5.4 Problem Statement

Here, we describe the problem definition in more detail. In this chapter, we predict daily water consumption for multi-family residences using water consumption data from the city of Abbotsford for the period between September 1, 2012 to August 31, 2013. We build multiple models, one for each dissemination area, to predict the daily water consumption per unit for a particular dissemination area. The inputs to the model are fixed sized windows of previous daily water consumption as specified in Section 5.6. The output of interest is the predicted daily water consumption per unit for a dissemination area, where per unit water consumption is aggregated at the dissemination area level. Table 5.1 outlines the model built in this chapter.

Table 5.1: Deep learning model characteristics for predicting daily water consumption of multi-family residences at the dissemination area level

Single or multiple models?	Multiple, one for each dissemination area
Model input(s)	Fixed sized input windows of previous daily water consumption as described in Section 5.6
Model output(s)	Predicted daily water consumption per unit for a dissemination area, where per unit water consumption is aggregated at the dissemination area level. Elaborated in Section 5.7
Temporal scale	Daily
Spatial scale	Dissemination area
Train dataset	Daily aggregated water consumption data from September 1, 2012 to June 30, 2013 for a particular dissemination area.
Validation dataset	Daily aggregated water consumption data from July 1, 2013 to July 31, 2013 for a particular dissemination area.
Test dataset	Daily aggregated water consumption data from August 1, 2013 to August 31, 2013 for a particular dissemination area.

The reasoning behind why we train multiple models, one for each dissemination area, is due to the nature of recurrent neural networks, GRU, and LSTM. These particular models are capable of remembering previous patterns in the data. We train a model for each dissemination area as we only want the model to remember water consumption patterns of the dissemination area being predicted for and not the patterns of other dissemination areas as some tend to have completely different usage patterns or a different average water usage. For example, some dissemination areas have a much more seasonal pattern than others, with higher usage in the summer months. And not all dissemination areas have an average water usage within the same range as each other.

5.5 Data Preprocessing

In this chapter, we use water consumption data for the city of Abbotsford from September 1, 2012 to August 31, 2013 obtained after the cleaning and preprocessing steps as described in Chapter 3. We train a model for each dissemination area and split the data into a train, validation, and test set. Each dissemination area has daily water consumption data for the period beginning on September 1, 2012 and ending on August 31, 2013. We set the training dataset from September 1, 2012 to June 30, 2013, the validation dataset from July 1, 2013 to July 31, 2013, and the test dataset from August 1, 2013 to August 31, 2013. The dataset was split in this way as it follows the true chronological order of the data. It is also possible to separate the data in other ways, such as setting January as the validation set and February as the test set. We have verified that using different months as the validation and test set give us similar results to the results obtained when using July and August as the validation and test sets respectively.

5.6 Model Inputs

In this section, we explain the inputs to our deep learning models. For recurrent neural networks, GRU, LSTM, and convolutional neural networks, the inputs are past daily water consumption from the last 15 days, thus a window size of 15, to predict the current daily water consumption as the output of the model. We train recurrent neural networks, GRU, and LSTM using non-overlapping windows of past daily water consumption data. We found that training these models with overlapping windows would result in models which overfit on the training dataset and lead to predictions which do not generalize well to new data. For convolutional neural networks, unlike recurrent neural networks, GRU, and LSTM, we found better performance on the validation dataset by training convolutional neural networks on overlapping windows of past daily water consumption data, without overfitting the training dataset. Figure 5.1 provides an example of overlapping vs. non-overlapping input windows of size four. The window size was determined based on the best performance on the validation dataset. In this case, a window size of 15 yielded the best performance on the validation set for most models. It should be noted that the choice of window size was found to be robust. Window sizes between the 10 to 20 range also worked well. We also note that the window size directly corresponds to the size (the number of time steps) of the truncated unrolled network from which backpropagation through time is applied for models based on recurrent neural networks.

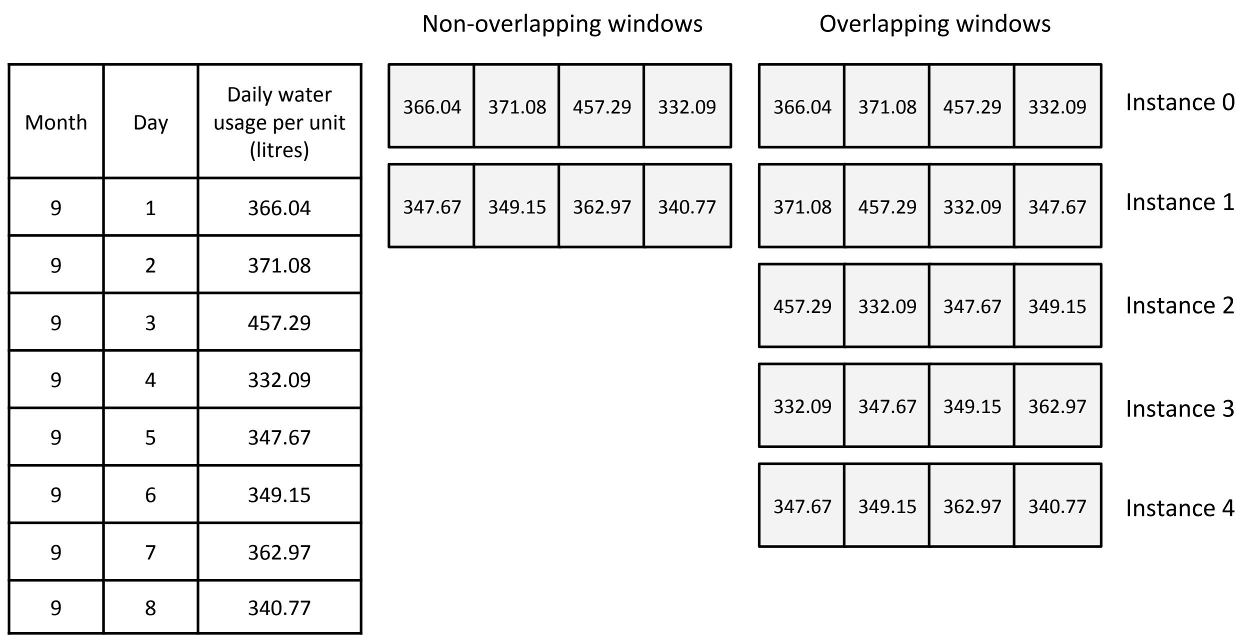


Figure 5.1: Overlapping vs. non-overlapping input windows for deep learning models

5.7 Model Outputs

The output of our RNN, GRU, and LSTM models take on the form of fixed sized windows and occur one time step ahead of its corresponding input window. The output we are interested in is the last value of the output window, as this is the predicted next value of the input sequence. This is also elaborated in Section 2.2.4. For CNNs, the output is a single value which is the predicted next value of the input sequence.

5.8 Regularization

To prevent overfitting, we use early stopping [52] as a regularization technique for recurrent neural networks, GRU, LSTM, and convolutional neural networks. To perform early stopping, we train the model on the training dataset and keep track of the validation performance. When the validation performance starts to decline, we take note of the number of epochs. This process is repeated until we obtain an average number of epochs. We train our model to this number of epochs to prevent overfitting.

5.9 Grid Search

We conduct grid search on one particular dissemination area, as conducting a grid search for each dissemination area would be too time consuming and impractical. We select a dissemination area (DAUID: 59090374) with generally consistent water usage throughout the year, containing 139 multi-family units. During grid search, models are trained on the train dataset and performance is evaluated on the validation dataset. Compared to other machine learning models, such as SVR, deep learning models have a much larger parameter space and the structure of the model must also be determined. This makes conducting a thorough grid search across all parameters very impractical. Instead, we select parameter values based on recommendations from Goodfellow et al. [25] and LeCun et al. [37], and perform a fine grid search based on these recommendations. For machine learning models with a feature engineering and feature selection step, the same grid search strategy from Chapter 3 is conducted, where we start with a series of coarse grid searches followed by finer grid searches. Table 5.2 reports the optimal parameters found for the models of interest. Table 5.4 shows the test performance for each model. The test performance for each model is obtained by training a model for each dissemination area and averaging out the test MAE for all dissemination areas. In general, we found that deep learning models take longer to train compared to other models such as SVR and tree ensemble models. However, this increase in training time is only in the order of a few minutes. Traditional machine learning techniques took an additional two to three weeks before training due to requiring a feature engineering and feature selection step.

In terms of the model structure for recurrent neural networks, GRU, LSTM, and convolutional neural networks, we found that a shallow network structure performs adequately on the validation dataset. For RNNs, GRU, and LSTM, one layer of recurrent neurons was found to perform well. For CNNs, a shallow architecture as described in Table 5.3 was found to perform well. Due to the small size of the dataset, using a deep network structure would result in overfitting the training dataset and not generalizing well to unseen data. We are, however, able to obtain robust results on a wide configuration of parameters using shallow networks. For example, for CNNs, we are able to get similar results for a reasonable range of different parameter values for the filter size and number of filter maps. Using a different optimizer and activation function from the ones specified in Table 5.2 also gives us comparable results. Similarly, for recurrent neural networks, GRU, and LSTM, varying the number of recurrent neurons, optimizer, and activation function gives us comparable results to using the optimal set of parameters described in Table 5.2.

Table 5.3 provides a description of the structure of our CNN model. The CNN model contains layers in the following order: an input layer where windows of size 1x15 are passed into the model, a convolutional layer where 100 feature maps of size 1x2 are learned using a filter of size 1x14, a max pooling layer which downsamples the feature maps to size 1x1 using a max filter of size 1x2, a flatten layer which flattens the output matrix of the pooling layer, and finally a dense fully connected layer which outputs the model prediction, which is a single value and is the predicted next value of the time series sequence.

Comparing deep learning models, RNNs, GRU, and LSTM were more prone to overfitting compared to CNN models. The training time for RNNs, GRU, and LSTM was longer (in the order of minutes) compared to CNNs.

Table 5.2: Grid searched parameters for deep learning models and LinearSVR

Model	**Optimal parameters**
RNN	activation: "tanh"
	learn_rate: 0.0001
	num_layers: 1
	nodes_per_layer: 100
	optimizer: "adam"
	window_size: 15
GRU	activation: "tanh"
	learn_rate: 0.0001
	num_layers: 1
	nodes_per_layer: 100
	optimizer: "adam"
	window_size: 15
LSTM	activation: "tanh"
	learn_rate: 0.0001
	num_layers: 1
	nodes_per_layer: 100
	optimizer: "adam"
	window_size: 15

CNN	activation: “tanh” filter_length: 14 learning_rate: 0.001 num_convolution_layers: 1 num_filters: 100 num_pooling_layers: 1 optimizer: “adam” pooling_type: “max” window_size: 15
LinearSVR	C: 1 epsilon: 0.0001 loss: “squared_epsilon_insensitive”

Table 5.3: Convolutional neural network architecture

Layer	**Type**	**Filter size**	**Stride**	**Feature map size**	**Number of feature maps**	**Output shape**	**Trainable parameters**	**Activation**	**Optimizer**
In	Input	-	-	-	-	1x15	-	-	-
C1	Convolution	1x14	1	1x2	100	100x2	1500	tanh	adam
S2	Max pooling	1x2	2	1x1	1	100x1	0	-	-
F3	Flatten	-	-	-	-	1x100	0	-	-
D4	Dense fully connected	-	-	-	-	1x1	101	linear	adam

5.10 Model Performance

Table 5.4: Deep learning and machine learning model performance using grid searched parameters

Model	**Average test MAE over all DAs (litres)**	**Feature engineering and selection step?**
RNN	54.83	No
CNN	55.09	No
GRU	57.62	No
LSTM	58.15	No
LinearSVR	55.68	Yes
Linear Regression	56.06	Yes
AdaBoost	57.52	Yes
SVR	57.86	Yes
KNN	59.74	Yes
Neural network with one hidden layer	65.07	Yes
Decision Tree	66.56	Yes
Neural network with two hidden layers	68.04	Yes
Random Forest	69.20	Yes
Gradient Boosting	87.57	Yes

Overall, we find that deep learning models perform comparably to traditional machine learning models. Figure 5.2 shows the test performance for recurrent neural networks, GRU, LSTM, convolutional neural networks, and LinearSVR for each dissemination area. LinearSVR is the best performing model with a feature engineering and feature selection step. The ordering for each model on the x-axis is in ascending order of the test MAE for the 30 dissemination areas. As with the results depicted in Table 5.4, the plots show that these models perform similarly. There is no particular model which appears to outperform all others. The plots for GRU and LSTM give us an indication as to why the test MAE is worse in Table 5.4 for these models compared to the other deep learning models. GRU and LSTM perform significantly worse on one particular dissemination area, which pushes its average test MAE upwards. For the remaining dissemination areas, GRU and LSTM perform similarly compared to the other models. In Figure 5.2, the dissemination areas plotted between 25 and 30 are dissemination areas which have a fewer number of multi-family units. The daily water consumption patterns for these dissemination areas tend to have very seasonal patterns, with more usage in the summer months compared to the winter. There are also significantly more outlying values compared to dissemination areas with a greater number of multi-family units. In general, we expect to have worse performance on these types of dissemination areas as they are less aggregated compared to dissemination areas with a greater number of multi-family units. Section 5.11 describes model accuracy performance on dissemination areas in more detail.

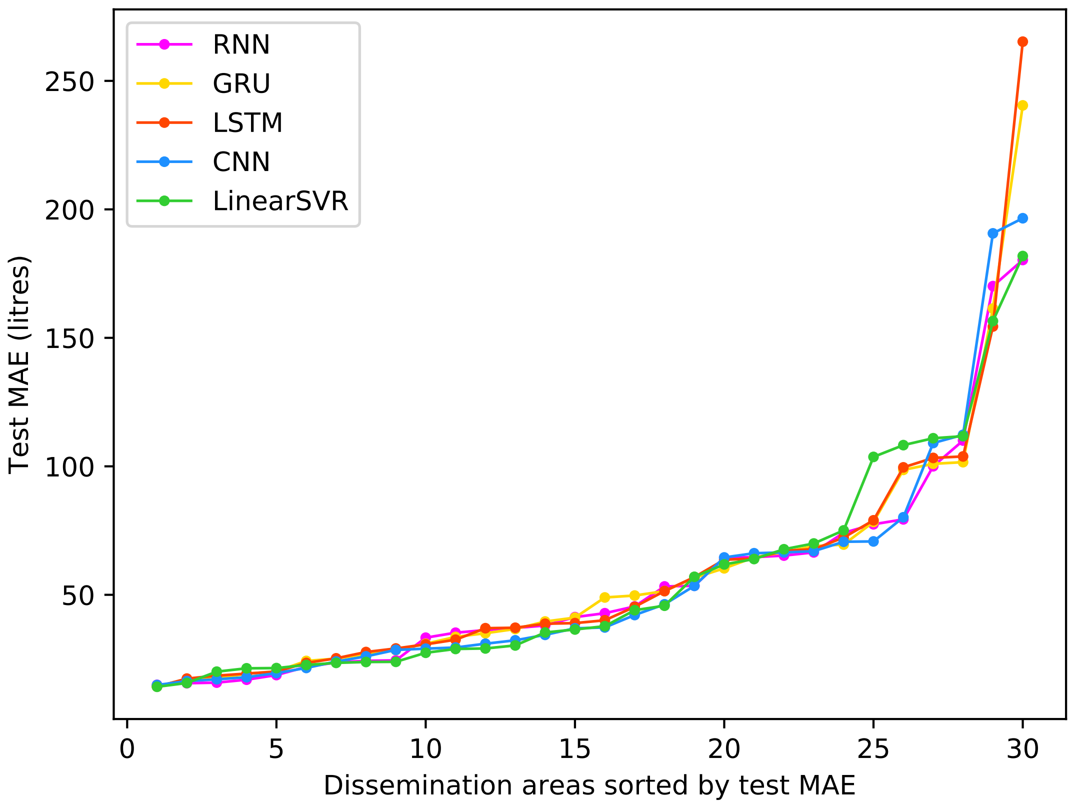


Figure 5.2: Comparing test performance of deep learning models and LinearSVR

5.11 Dissemination Area Performance

Of the 30 dissemination areas in our dataset, there are certain dissemination areas which are more difficult to predict accurately. For the models we have built throughout the entire thesis, we find that dissemination areas which have a higher standard deviation on its daily water consumption from September 1, 2012 to August 31, 2013 are more difficult to predict and give us a higher test error. These dissemination areas have a greater difference on average from the mean daily water usage for the dissemination area. This trend is depicted in Table 5.5, where we show the test MAE compared to the standard deviation for each dissemination area from the RNN model obtained in Chapter 5. In Table 5.5, sorting the dissemination areas by the test MAE shows that as the standard deviation increases, the error also tends to increase. Note that for merged dissemination areas, only the first DAUID is reported in the table. We plot a boxplot of daily water consumption from September 1, 2012 to August 31, 2013 for each dissemination area as shown in Figure 5.3. Each boxplot depicts the 5th, 25th, 50th, 75th, and 95th percentiles of daily water consumption for a dissemination area. Along the x-axis, the dissemination areas are ordered by test MAE, from best to worst performance. We observe that as the performance worsens, the spread of the data increases. For the multi-family residence models in Chapter 3, the urban planning models in Chapter 4, and the remaining deep learning models in Chapter 5, we also observe this trend.

We find that a higher standard deviation on the daily water usage for a dissemination area is caused by a dissemination area having a fewer number of multi-family units. Since these dissemination areas contain a fewer number of multi-family units, the daily aggregated water usage will be less aggregated compared to dissemination areas with a greater number of multi-family units. A dissemination area which is less aggregated will contain significantly more outlying values, which are harder to predict and increases both the test error and the standard deviation. As mentioned in Chapter 3, we retained most outlying values in the data preprocessing step as they were not found to be the result of an error. Overall, some dissemination areas are more difficult to predict accurately due to these dissemination areas having more outliers on average.

Figure 5.4 depicts the daily water usage for dissemination areas 59090068, 59090070, 59090080, and 59090374, respectively, which have a lower standard deviation and test error. These dissemination areas tend to have fairly consistent daily water consumption throughout the year and do not contain large outliers.

In contrast, Figure 5.5 depicts the daily water consumption for dissemination areas 59090051, 59090056, 59090384, and 59090797, respectively. These dissemination areas have a higher standard deviation and test error compared to other dissemination areas in our dataset. We observe that most of these dissemination areas tend to have a much more seasonal pattern, with much higher daily water usage in the summer months which are more difficult to predict accurately. It should be noted that dissemination areas without a seasonal pattern, but contain large outliers also have a larger standard deviation and test error, as is the case with dissemination area 59090056.

Table 5.5: Test MAE and standard deviation for each DA using RNN model

Dissemination area	**Test MAE (litres)**	**Standard deviation (litres)**
59090070	14.96	22.66
59090080	15.59	23.40
59090107	15.82	28.07
59090064	16.96	25.79
59090110	18.77	51.22
59090068	21.95	35.67
59090113	23.66	37.25
59090103	24.30	70.80
59090374	24.47	36.14
59090439	33.31	36.26
59090057	35.25	69.59
59090063	36.27	45.93
59090444	37.05	51.07
59090078	38.05	45.73
59090094	41.32	44.20
59090065	42.83	65.54
59090086	45.47	60.77
59090114	53.26	57.52
59090112	53.51	55.45
59090796	64.39	62.98
59090074	64.65	115.43
59090144	65.26	99.21
59090089	66.50	149.66
59090056	74.11	81.78
59090129	77.44	119.61
59090375	79.35	82.29
59090797	100.02	123.24
59090051	110.03	150.96
59090384	170.12	222.09
59090792	180.30	311.63

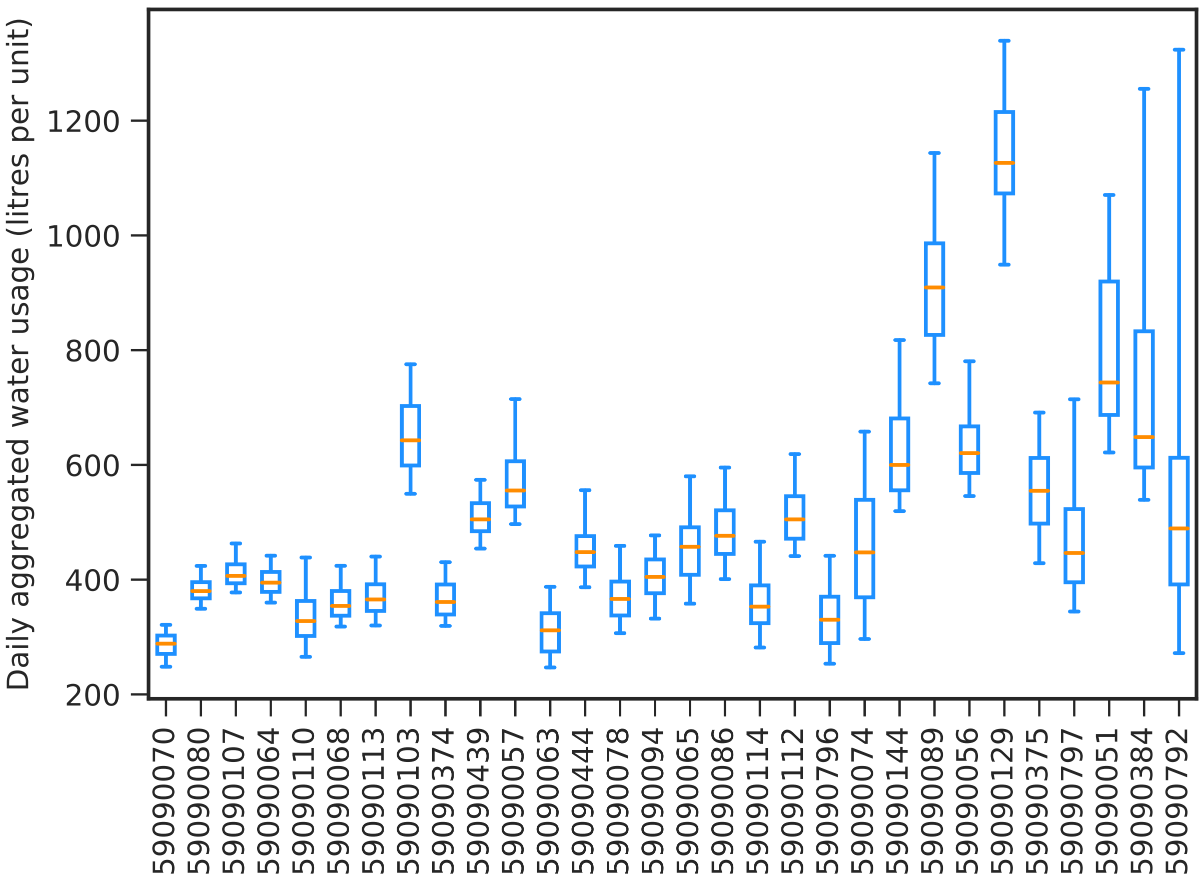
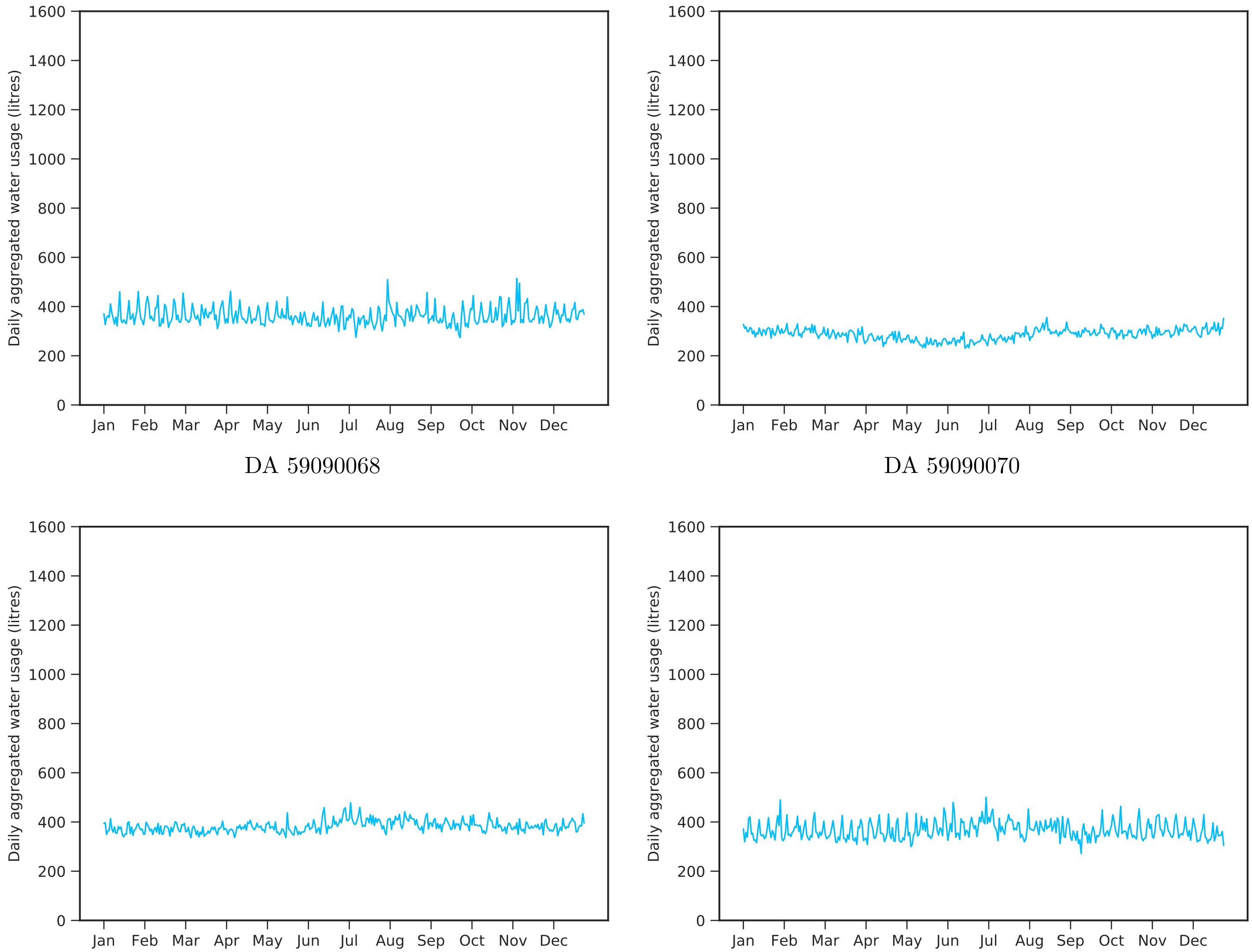


Figure 5.3: Dissemination areas sorted by test MAE in ascending order using RNN model



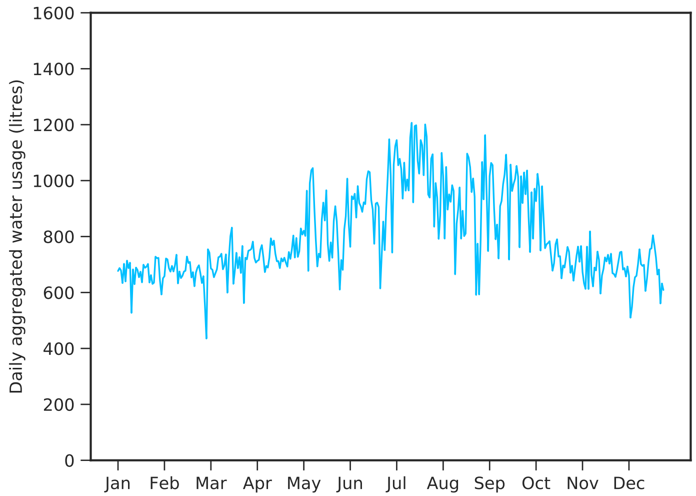
DA 59090068

DA 59090070

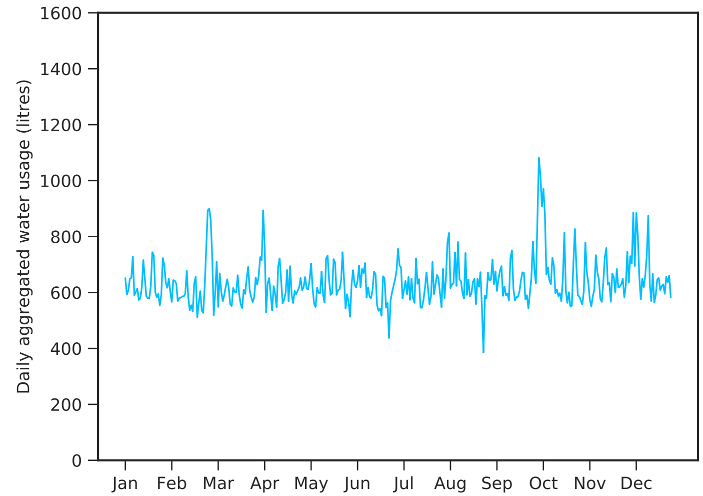
DA 59090080

DA 59090374

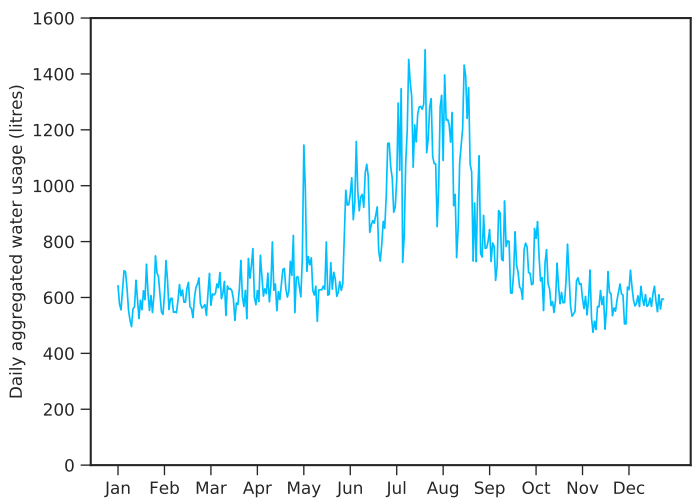
Figure 5.4: Dissemination areas with a low standard deviation on daily water usage



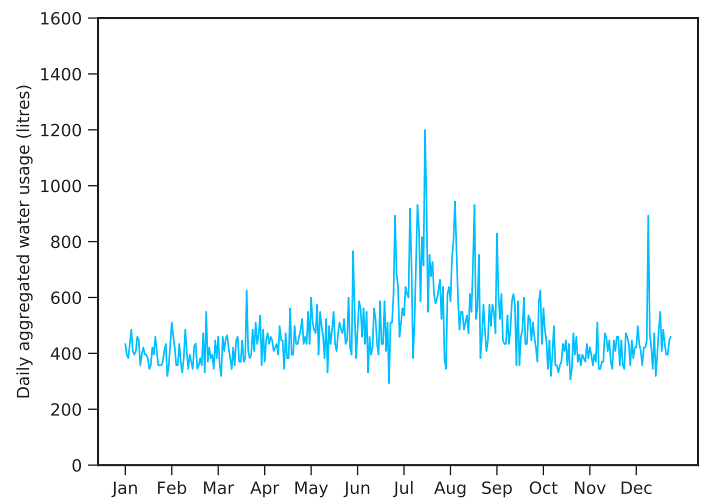
DA 59090051



DA 59090056



DA 59090384



DA 59090797

Figure 5.5: Dissemination areas with a high standard deviation on daily water usage

5.12 Model Predictions

Figure 5.6 shows the predicted daily water consumption compared to the actual daily water consumption for the period between March 1, 2013 to August 31, 2013 using the recurrent neural network model. We plot a particular dissemination area (DAUID: 59090374) which contains 139 multi-family units. LinearSVR, GRU, LSTM, and CNN have visually similar predictions to recurrent neural networks. For the particular dissemination area plotted, as well as for other dissemination areas, daily water consumption is more variable and harder to predict in the summer months, particularly in July and August. This is due to households using more water due to the warmer climate, or using less water due to being away on vacation. As with the models described in Chapter 3, peak usage is difficult to predict. This is likely due to the small size of the dataset.

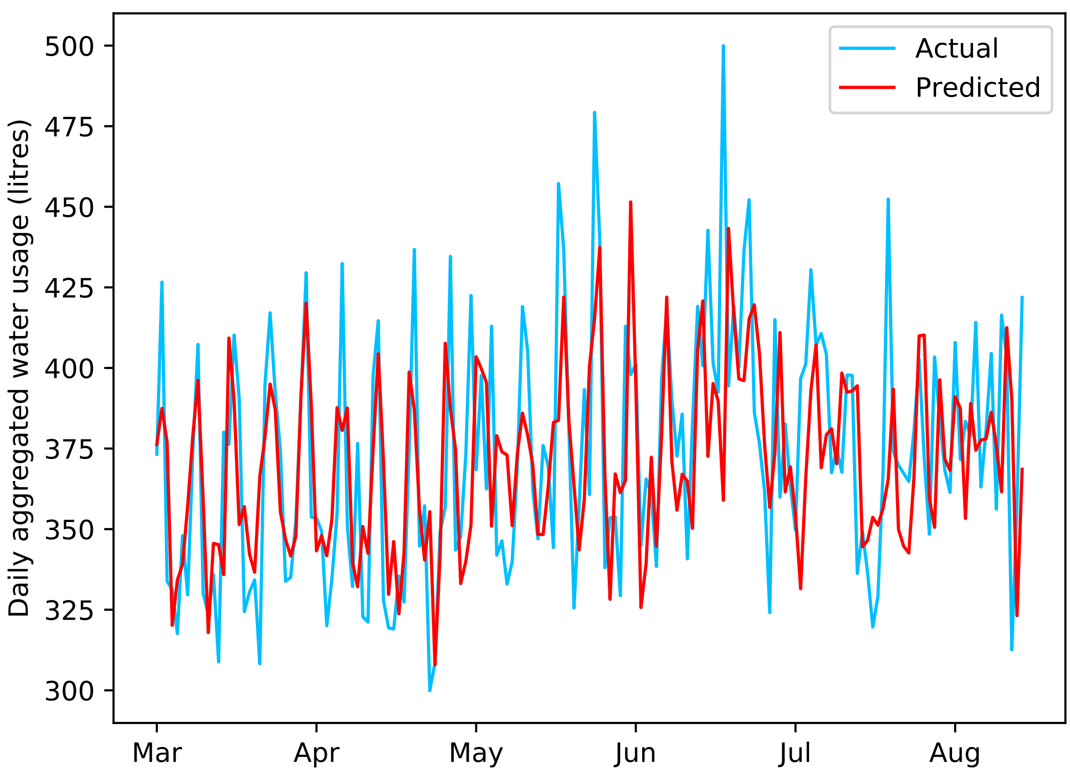


Figure 5.6: Actual vs. predicted water usage using RNN model for DA 59090374

5.13 Conclusion and Future Work

In this chapter, we built deep learning models such as recurrent neural networks and convolutional neural networks to predict daily water consumption for multi-family residences in the city of Abbotsford. We obtained comparable performance using recurrent neural networks and convolutional neural networks compared to LinearSVR, the best performing model which included a feature engineering and selection step. Since deep learning models are capable of learning data representations and therefore do not require a feature engineering or selection step, a considerable amount of time can be saved. For water utilities, deep learning models could present a solution which saves valuable time and money.

One of the disadvantages of deep learning models is that they are not easily interpretable, compared to tree-based methods which can be understood quite easily. In particular, features learned by deep learning models cannot be easily interpreted. In addition, although deep learning models can be used as black boxes in practice, technical expertise is still required to set up deep learning frameworks, train deep learning models, and tune model hyperparameters. If a machine learning practitioner requires a model which offers easy interpretation, it would be more advantageous to use a tree-based model rather than a deep learning model. The primary advantage of using deep learning models for our problem is its capability of learning features from data which results in not requiring a feature engineering or selection step.

In terms of future work, it would be interesting to see the performance we could obtain with bidirectional recurrent neural networks and other deep learning models. Bidirectional recurrent neural networks are particularly interesting since the network will have knowledge of both the past and the future during training. The output of the model is determined given both the past and future context. In addition, as more data is gathered, the performance of deeper network structures may also improve and overfit less. We were limited to shallow network structures in this chapter due to the small size of the dataset causing overfitting issues, but a larger dataset could allow deeper network structures to be investigated in more detail.

Another area of future work which can be investigated is the idea of using transfer learning to obtain improved model performance, particularly when the size of the dataset is small, as is the case in this chapter. In transfer learning, a deep learning model is initially trained on a dataset with similar characteristics as the current dataset and initially achieves a similar task as the current model objective. This initial model is then trained on the dataset at hand and the model is fine-tuned based on the current objective. Typically, the lower layers of the model are preserved and the trainable parameters of the higher layers are tweaked, as lower layers capture low-level features and are more likely to be applicable to the current task compared to higher layers which capture more specific high-level features [20]. In terms of the objective of this chapter, a deep learning model can initially be trained on the other dissemination areas and later on the dissemination area of interest. The deep learning model can then be fine-tuned, at the higher layers, based on the dissemination area of interest. Since the model is trained on more data, there is potential to achieve improved model performance as well as potential to achieve superior performance compared to traditional machine learning techniques. The idea of transfer learning has been shown to work well when applied to image recognition tasks and many other applications which use deep neural networks.

Chapter 6

Conclusion

There are three main contributions to this thesis from which we obtained the following results:

- First, we built machine learning models to predict daily water consumption for existing multi-family households in the city of Abbotsford. From this, we obtained models with accurate predictive accuracy and also investigated the determinants of water consumption for multi-family residences in detail.
- Second, we presented a new methodology to predict daily water consumption for new developments at the dissemination area level. Using this methodology, we obtained a machine learning model which significantly improves over industry baseline models. This new methodology can also be applied to other cities with the appropriate data at hand.
- Third, we built deep learning models which are capable of learning data representations to predict water consumption for existing multi-family residences. This could enable water utilities to save valuable time and resources in the model building process, as deep learning models do not require a feature engineering and selection step. With deep learning models, we obtain comparable results to traditional machine learning techniques.

To our knowledge, these three main research contributions have not yet been attempted in the current literature.

References

[1] Jan Adamowski, Hiu Fung Chan, Shiv O Prasher, Bogdan Ozga-Zielinski, and Anna Sliusarieva. Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in montreal, canada. *Water Resources Research*, 48(1), 2012.

[2] Jan Franklin Adamowski. Peak daily water demand forecast modeling using artificial neural networks. *Journal of Water Resources Planning and Management*, 134(2):119–128, 2008.

[3] Donald E Agthe and R Bruce Billings. Water price influence on apartment complex water use. *Journal of water resources planning and management*, 128(5):366–369, 2002.

[4] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[6] R Bruce Billings and Clive Vaughan Jones. Forecasting urban water demand. 2008.

[7] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In *European Business Intelligence Summer School*, pages 62–77. Springer, 2012.

[8] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*, 2017.

[9] Robert M Bradley. Forecasting domestic water use in rapidly urbanizing areas in asia. *Journal of Environmental Engineering*, 130(4):465–471, 2004.

[10] Statistics Canada. 2011 census dictionary, 2012. 98-301-X.

[11] Graham P Clarke, A Kashti, Adrian McDonald, and Paul Williamson. Estimating small area demand for water: a new methodology. *Water and Environment Journal*, 11(3):186–192, 1997.

[12] Paul Coomes, Tom Rockaway, Josh Rivard, and Barry Kornstein. North america residential water usage trends since 1992. *Water Research Foundation, Denver*, 2010.

[13] Talita Flores Dias, Andreza Kalbusch, and Elisa Henning. Factors influencing water consumption in buildings in southern brazil. *Journal of Cleaner Production*, 184:160–167, 2018.

[14] Elena Domene and David Saurí. Urbanisation and water consumption: Influencing factors in the metropolitan region of barcelona. *Urban Studies*, 43(9):1605–1623, 2006.

[15] Emmanuel A Donkor, Thomas A Mazzuchi, Refik Soyer, and J Alan Roberson. Urban water demand forecasting: review of methods and models. *Journal of Water Resources Planning and Management*, 140(2):146–159, 2012.

[16] Karl Filiatrault. 2013 sprinkling restrictions. Technical Report WSC 21-2013, Abbotsford/Mission Water & Sewer Commission, 2013.

[17] Karl Filiatrault and Tracy Kyle. Smart meter update. Technical Report ENG 45-2011, Abbotsford/Mission Water & Sewer Commission, 2013.

[18] C Fox, Brian S. McIntosh, and Paul Jeffrey. Classifying households for water demand forecasting using physical property characteristics. *Land Use Policy*, 26(3):558–568, 2009.

[19] Thomas M Fullerton Jr and Juan P Cárdenas. Forecasting water demand in phoenix, ariz. *Journal-American Water Works Association*, 108(10):E533–E545, 2016.

[20] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2017.

[21] Iman Ghalehkhondabi, Ehsan Ardjmand, William A Young, and Gary R Weckman. Water demand forecasting: review of soft computing methods. *Environmental monitoring and assessment*, 189(7):313, 2017.

[22] Saeed Ghavidelfar, Asaad Y Shamseldin, and Bruce W Melville. A multi-scale analysis of low-rise apartment water demand through integration of water consumption, land use, and demographic data. *JAWRA Journal of the American Water Resources Association*, 52(5):1056–1067, 2016.

[23] Manoochehr Ghiassi, David K Zimbra, and Hassine Saidane. Urban water demand forecasting with a dynamic artificial neural network model. *Journal of Water Resources Planning and Management*, 134(2):138–146, 2008.

[24] CW Goodchild. Modelling the impact of climate change on domestic water demand. *Water and Environment Journal*, 17(1):8–12, 2003.

[25] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[26] Guenter Hauber-Davidson and E Idris. Smart water metering. *Water*, 33(3):38–41, 2006.

[27] Larry L Havlicek and Nancy L Peterson. Robustness of the pearson correlation against violations of assumptions. *Perceptual and Motor Skills*, 43(3_suppl):1319–1334, 1976.

[28] Wan He. Load forecasting via deep neural networks. *Procedia Computer Science*, 122:308–314, 2017.

[29] Matthew Heberger, Kristina Donnelly, and Heather Cooley. A community guide for evaluating future urban water demand. 2016.

[30] Manuel Herrera, Luís Torgo, Joaquín Izquierdo, and Rafael Pérez-García. Predictive models for forecasting hourly urban water demand. *Journal of hydrology*, 387(1-2):141–150, 2010.

[31] Lily House-Peters, Bethany Pratt, and Heejun Chang. Effects of urban spatial structure, sociodemographics, and climate on residential water consumption in hillsboro, oregon 1. *JAWRA Journal of the American Water Resources Association*, 46(3):461–472, 2010.

[32] Lily A House-Peters and Heejun Chang. Urban water demand modeling: Review of concepts, methods, and organizing principles. *Water Resources Research*, 47(5), 2011.

[33] Google Inc. Google maps. https://www.google.com/maps, online; accessed Jul 2018.

[34] Khurshid M Kiani and Terry L Kastens. Testing forecast accuracy of foreign exchange rates: Predictions from feed forward and various recurrent neural network architectures. *Computational Economics*, 32(4):383–406, 2008.

[35] Constantine E Kontokosta and Rishee K Jain. Modeling the determinants of large-scale building water use: Implications for data-driven urban sustainability policy. *Sustainable Cities and Society*, 18:44–55, 2015.

[36] Chung-Ming Kuan and Tung Liu. Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of applied econometrics*, 10(4):347–364, 1995.

[37] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[38] Ziqiao Liu, Wenzhong Gao, Yih-Huei Wan, and Eduard Muljadi. Wind power plant prediction by using neural networks. In *Energy Conversion Congress and Exposition (ECCE), 2012 IEEE*, pages 3154–3160. IEEE, 2012.

[39] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[40] JK Mandal, AK Sinha, and G Parthasarathy. Application of recurrent neural network for short term load forecasting in electric power system. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 5, pages 2694–2698. IEEE, 1995.

[41] Marshall and Swift. *Residential Cost Handbook*. CoreLogic Inc., 2017.

[42] Robert I McDonald, Katherine Weber, Julie Padowski, Martina Flörke, Christof Schneider, Pamela A Green, Thomas Gleeson, Stephanie Eckman, Bernhard Lehner, Deborah Balk, et al. Water on an urban planet: Urbanization and the reach of urban water infrastructure. *Global Environmental Change*, 27:96–105, 2014.

[43] Canada Mortgage and Housing Corporation. Rental market report. https://www.cmhc-schl.gc.ca/en, online; accessed Jan 2018.

[44] Geoff Norman. Likert scales, levels of measurement and the "laws" of statistics. *Advances in health sciences education*, 15(5):625–632, 2010.

[45] Frederico Keizo Odan and Luisa Fernanda Ribeiro Reis. Hybrid water demand forecasting model associating artificial neural network with fourier series. *Journal of Water Resources Planning and Management*, 138(3):245–256, 2012.

[46] City of Pasadena. City of pasadena general plan: Land use element. Technical report, City of Pasadena, 2016.

[47] Avishek Pal and P.K.S. Prakash. *Practical Time Series Analysis*. Packt Publishing, 2017.

[48] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[49] Valerie Platsko. Smart-meter enabled estimation and prediction of outdoor residential water consumption. Master's thesis, University of Waterloo, 2018.

[50] Valerie Platsko, Peter van Beek, et al. Identification, prediction, and explanation of outdoor residential water consumption using smart meter data. In *WDSA/CCWI Joint Conference Proceedings*, volume 1, 2018.

[51] Austin S Polebitski, Richard N Palmer, and Paul Waddell. Evaluating water demands under climate change and transitions in the urban environment. *Journal of Water Resources Planning and Management*, 137(3):249–257, 2010.

[52] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[53] Chai Quek, Michel Pasquier, and Neha Kumar. A novel recurrent neural network based prediction system for trading. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 2090–2097. IEEE, 2006.

[54] Jean-Daniel Rinaudo. Long-term water demand forecasting. In *Understanding and managing urban water in transition*, pages 239–268. Springer, 2015.

[55] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[56] Joachim Schleich and Thomas Hillenbrand. Determinants of residential water demand in germany. *Ecological economics*, 68(6):1756–1769, 2009.

[57] Brent Schmitt. Amwsc water demand projections. Technical Report WSC 27-2013, Abbotsford/Mission Water & Sewer Commission, 2013.

[58] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[59] Rodney A Stewart, Rachelle Willis, Damien Giurco, Kriengsak Panuwatwanich, and Guillermo Capati. Web-based knowledge management system: linking smart metering to the future of urban water planning. *Australian Planner*, 47(2):66–74, 2010.

[60] James H Stock and Mark W Watson. *Introduction to econometrics: Global edition.*

[61] Paolo Tenti. Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*, 10(6):567–582, 1996.

[62] Mukesh K Tiwari and Jan Adamowski. Urban water demand forecasting and uncertainty assessment using ensemble wavelet-bootstrap-neural network models. *Water Resources Research*, 49(10):6486–6507, 2013.

[63] Mukesh K Tiwari and Jan F Adamowski. Medium-term urban water demand forecasting with limited data using an ensemble wavelet–bootstrap machine-learning approach. *Journal of Water Resources Planning and Management*, 141(2):04014053, 2014.

[64] Amy Wakeford. 2011 water conservation summary. Technical Report WSC 1-2012, Abbotsford/Mission Water & Sewer Commission, 2012.

[65] David J Walker, Enrico Creaco, Lydia S Vamvakeridou-Lyroudia, Raziyeh Farmani, Zoran Kapelan, and Dragan Savić. Forecasting domestic water consumption from smart meter readings using statistical methods and artificial neural networks. 2015.

[66] Elizabeth A Wentz and Patricia Gober. Determinants of small-area water consumption for the city of phoenix, arizona. *Water Resources Management*, 21(11):1849–1863, 2007.

[67] Elizabeth A Wentz, Angela J Wills, Won Kyung Kim, Soe W Myint, Patricia Gober, and Robert C Balling Jr. Factors influencing water consumption in multifamily housing in tempe, arizona. *The Professional Geographer*, 66(3):501–510, 2014.

[68] Jian Zheng, Cencen Xu, Ziang Zhang, and Xiaohua Li. Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In *Information Sciences and Systems (CISS), 2017 51st Annual Conference on*, pages 1–6. IEEE, 2017.

[69] Anwen Zhu, Xiaohui Li, Zhiyong Mo, and Ruaren Wu. Wind power prediction based on a convolutional neural network. In *Circuits, Devices and Systems (ICCDS), 2017 International Conference on*, pages 131–135. IEEE, 2017.