Probabilistic Forecasts of Day-Ahead Electricity Prices in a Highly Volatile Electricity Market

by

Behrouz Banitalebi

A Thesis submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Asper School of Business
Department of Supply Chain Management
University of Manitoba
Winnipeg

Copyright © 2021 by Behrouz Banitalebi
Abstract

Electricity price forecasting plays an important role in decision-making on bidding strategies of selling and buying electricity. This thesis computes one-day-ahead quantile forecasts of electricity prices in a highly volatile market by applying regression models to a pool of point forecasts. Three data-driven forecasting methods are implemented to generate day-ahead point forecasts of the Ontario market’s electricity prices. In order to generate the three sets of point forecasts, I use: i) the Triple Exponential Smoothing (TES) method, ii) a Neural Network (NN) that combines layers of Convolutional neurons and Gradient Recurrent Units (GRU), iii) an eXtreme Gradient Boosting (XGB) non-linear regression approach. The performance of the three models is compared against a benchmark that considers the forecast of electricity prices as the average price of the same hour and day during the last four weeks. The TES method decreases the Mean Absolute Error (MAE) of the benchmark model from 10.29 to 9.42. The Convolutional GRU (ConvGRU) model and XGB regression also reduce the MAE to 8.20 and 7.06, respectively.

Finally, Quantile Regression Averaging (QRA) is applied to the pool of point forecasts obtained by TES, ConvGRU, and XGB methods to compute day-ahead quantile forecasts of electricity prices. Moreover, the QRA method is further developed in this thesis by employing Gradient Boosting Regression (GBR). It follows from my real data analysis that the GBR method provides more reliable quantiles and tighter prediction intervals with smaller forecasting errors than QRA.

The obtained probabilistic forecasts are used to find the optimal energy procurement plan for a large consumer and the linear programming method is applied to solve the problem. The simulation results indicate that using probabilistic forecasts of electricity prices leads to a more flexible and efficient bidding strategy than using point forecasts. Moreover, the regularized probabilistic forecast of day-ahead electricity demands is computed and used to model power generation units’ scheduling.
Acknowledgements

I am extremely grateful to my supervisors, Dr. Srimantoorao S. Appadoo and Dr. Yuvraj Gajpal, for their invaluable advice and continuous support during my MSc study. Their immense knowledge and plentiful experience have encouraged me in my academic research and daily life. My sincere thanks must also go to my MSc committee members: Dr. Aermbamoorthy Thavaneswaran and Dr. Michael Lukie. They generously gave their time to offer me valuable comments toward improving my work. In particular, Dr. Thavaneswaran helped to shed new light on many of my ideas. My gratitude extends to the Asper school of business for the funding opportunity to undertake my studies at the Department of supply chain management, University of Manitoba.

Finally, I would like to express my appreciation to my parents and my wife. Without their tremendous understanding and encouragement in the past few years, it would be impossible to complete my study.
Dedication

For the love of my life, Forouz, who always stood by my side, no matter how difficult it was.
# Table of Contents

Abstract ................................................................. i
Acknowledgements ...................................................... ii
Dedication ............................................................... iii
List of Figures ............................................................ vi
List of Tables .............................................................. viii
List of Abbreviations .................................................... ix

1 Introduction and Literature Review ................................ 1
  1.1 Electricity Supply Chain ........................................... 1
  1.2 Deregulation of Electricity Supply Chain ......................... 2
  1.3 Various Markets in Electricity Supply Chain ....................... 4
  1.4 Motivation ........................................................ 5
  1.5 Literature Review ................................................. 5
  1.6 Gap in Research ................................................ 9
  1.7 Thesis Outline .................................................. 10

2 Machine Learning Models for Time Series Forecasting ............ 11
  2.1 Lasso Regression ................................................ 11
  2.2 Support Vector Regression ..................................... 13
  2.3 Gradient Boosting Regression ................................... 15
  2.4 Artificial Neural Networks ...................................... 18
  2.5 Gradient Descent ................................................ 19
  2.6 Momentum ....................................................... 21
  2.7 Variable and Adaptive Learning Rates ........................... 22
  2.8 Recurrent Neural Networks ...................................... 25
  2.9 Convolutional Neural Networks .................................. 28
  2.10 Evaluation Metrics ............................................. 29
# Volatility Analysis

3.1 Determinants of the Price Fluctuations .................................. 31
3.2 Volatility of Hourly Electricity Prices .................................... 32
3.3 Structure of Ontario Electricity Market .................................... 34
3.4 Day-Ahead Volatility Forecasts .............................................. 36
3.5 Double Exponential Smoothing ............................................. 36
3.6 Triple Exponential Smoothing ............................................. 37
3.7 Regularized Volatility Forecasts ........................................... 38

# Day-Ahead Forecasts of Electricity Demand

4.1 Load Forecasting Literature ................................................. 43
4.2 Electricity Demand Determinants .......................................... 46
4.3 Method of Analysis .......................................................... 47
4.4 ANN model for Day-Ahead Load Forecasting ............................... 50
4.5 Regression-Based Load Forecasting Methods ............................... 56

# Probabilistic Forecasts of Electricity Prices

5.1 Autocorrelation of Electricity Prices ..................................... 60
5.2 Pre-Processing of HOEP Data .............................................. 63
5.3 ConvGRU HOEP Forecasting Model ........................................ 66
5.4 Benchmark HOEP Forecasting Model ...................................... 68
5.5 Regression Based Forecasting Model ...................................... 70
5.6 Triple Exponential Smoothing ............................................. 71
5.7 Quantile Forecasts of Electricity Prices ................................... 71
5.8 Verification of Probabilistic Forecasts ................................... 76

# Applications of Probabilistic Forecasts in Power Markets

6.1 Power Production Planning .................................................. 81
6.2 Energy Procurement Strategy for Large Consumers .................... 87

# Conclusion and Future Research

7 Conclusion and Future Research ............................................ 95

# References

References ................................................................. 99
List of Figures

1.1 Structure of deregulated electricity supply chain .......................... 3
1.2 Social welfare and market clearing price .............................. 5

2.1 Structure of feed-forward neural network ............................. 18
2.2 Structure of Convolutional neural network ............................. 29

3.1 Effects of load and renewable generations on electricity price .......................... 32
3.2 Effects of time and date on electricity price .......................... 33
3.3 Comparison of Ontario electricity market vs. PJM and Nord Pool markets in terms of volatility ........................................... 34
3.4 Unregularized volatility forecasts of electricity prices .................. 40
3.5 Regularized volatility forecasts of electricity prices .................. 41

4.1 Variation of electricity demand vs. temperature in Ontario .................. 47
4.2 Comparison of electricity consumption during the weekdays versus weekends .................................................. 48
4.3 Impact of time and date on electricity demand ........................... 49
4.4 Dual-phase model for day-ahead load forecasting .......................... 49
4.5 Distribution of 2010-2019 Ontario electricity demand ............... 51
4.6 Data structure for training and testing the ANN model .................. 52
4.7 Training loss and learning rate over the course of model training .......... 54
4.8 Performance of the ANN load forecasting model for a random time period .................................................. 55
4.9 Hourly MAE of ANN load forecasting based on three different recurrent units .................................................. 56
4.10 Hourly MAE of regression methods for load forecasting ............... 57
4.11 Hourly MAE of dual phase load forecasting model ................... 58

5.1 Autocorrelation function of electricity prices .......................... 61
5.2 Training and testing HOEP data for ANN models ....................... 64
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Histograms of original and filtered HOEP data.</td>
<td>65</td>
</tr>
<tr>
<td>5.4</td>
<td>Loss and learning rate of training algorithm for the ANN price forecasting model.</td>
<td>67</td>
</tr>
<tr>
<td>5.5</td>
<td>Illustration of DA HOEP forecasts for a week of 2019.</td>
<td>68</td>
</tr>
<tr>
<td>5.6</td>
<td>MAE of ANN HOEP forecasting models.</td>
<td>69</td>
</tr>
<tr>
<td>5.7</td>
<td>MAE of 3 Different Day-ahead Point Forecasting Models.</td>
<td>71</td>
</tr>
<tr>
<td>5.8</td>
<td>Quantile loss by errors for three different quantile values.</td>
<td>74</td>
</tr>
<tr>
<td>5.9</td>
<td>Total loss per quantile of predictive models.</td>
<td>75</td>
</tr>
<tr>
<td>5.10</td>
<td>Reliability diagram of quantile forecasts obtained from QRA, GBR and NN methods.</td>
<td>77</td>
</tr>
<tr>
<td>5.11</td>
<td>Overall skill assessment of probabilistic forecasts of electricity prices.</td>
<td>78</td>
</tr>
<tr>
<td>5.12</td>
<td>Probabilistic, Point, and Quantile Prediction.</td>
<td>80</td>
</tr>
<tr>
<td>6.1</td>
<td>Power supply and distribution architecture.</td>
<td>82</td>
</tr>
<tr>
<td>6.2</td>
<td>Point and probabilistic forecasts of electricity demand.</td>
<td>85</td>
</tr>
<tr>
<td>6.3</td>
<td>Offers of the large consumer versus market-clearing prices.</td>
<td>89</td>
</tr>
<tr>
<td>6.4</td>
<td>Daily procured energy from day-ahead market.</td>
<td>93</td>
</tr>
<tr>
<td>6.5</td>
<td>Daily procured energy from day-ahead market.</td>
<td>94</td>
</tr>
<tr>
<td>6.6</td>
<td>Daily procured energy from day-ahead market and bilateral contracts.</td>
<td>94</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Descriptive Statistics of 2019 Ontario Electricity Prices

3.2 MAE of Different Volatility Forecasts of Electricity Price

4.1 Meteorological Variables Used in the Analysis

4.2 Descriptive Statistics of Electricity Demand Data

4.3 Initial Values of ANN Forecasting Model for Electricity Demand

4.4 Comparison of the Three Different Recurrent Layers in the ANN Load Forecasting Model

4.5 Evaluation of Regression Methods for Load Forecasting

4.6 Evaluation of XGB Prediction of Electricity Demand in Ontario

5.1 Initial Values ANN Forecasting Model

5.2 Comparison of ANN Forecasting Models Against Benchmarks

5.3 Evaluation of Predictive Density Models

6.1 States of Nature with Probabilities

6.2 Different Scenarios That May Happen During Power Trading

6.3 Payoff Table for Electricity Production in Scenario I

6.4 Expected Profit in Thousands of Dollars for Each Decision

6.5 Simulation Parameters for Optimization Problems

6.6 Average of Daily Procured Electricity Under the Two Scenarios
List of Abbreviations

AdaGrad Adaptive gradient algorithm
Adam Adaptive moment
ANN artificial neural network
CDF cumulative distribution function
CNN Convolutional neural network
ConvGRU Convolutional GRU
CRPS continuous ranked probability score
DES Double exponential smoothing
GBR Gradient Boosting Regression
GRU Gradient Recurrent Units
HOEP Hourly Ontario electricity price
IQR Inter-Quartile Range
Lasso least absolute shrinkage and selection operator
LSTM long short-term memory
MAE Mean Absolute Error
MSE mean squared error
NN Neural Network
PEPF PEPFprobabilistic electricity price forecasting
PI prediction intervals
QRA Quantile Regression Averaging
RMSE root mean squared error
RMSProp Root mean square propagation
RNN Recurrent neural network
std standard deviation
SVR Support vector regression
TES Triple Exponential Smoothing
XGB eXtreme Gradient Boosting
Chapter 1

Introduction and Literature Review

1.1 Electricity Supply Chain

Electricity supply chains are established to provide a continuous flow of electric energy from power plants to final consumers. A typical electricity supply chain consists of power generation plants, transmission lines, distributors, and final customers. Micro-grids and electricity markets are also emerging in the structure of the electricity supply systems. Electricity markets are the auctions that coordinate production, transmission, and distribution of the electrical energy. Micro-grids are small-scale power supply systems established at the customers’ side. Electromechanical generators and solar panels are the most common devices for producing electricity. The electromechanical generators are connected to the turbines that are driven by a source of power. Sources of power can include: kinetic energy such as wind and water, thermal energy produced by combustion or nuclear fission, and geothermal power.

Hydroelectric and thermal power plants are conventional ways of generating electricity. These plants transform mechanical energy into electricity on a large scale. Hydroelectric plants are widely used in the Canadian provinces such as Manitoba, Quebec, and BC due to the abundance of water resources. Thermal power plants are another class that has been mainly established in
Alberta, Saskatchewan, and Ontario. Renewable power plants such as solar and wind plants are also growing in Canada and are getting more popular because of their clean energy production. These plants convert the energy of renewable resources, such as wind or solar, to electricity. The main grid is the part of the electricity supply chain that delivers electricity from power plants to the final customers. Large-scale electricity producers are usually located far from residential areas. As a result, connecting power producers and consumers via the main grid is costly because the capacity of transmission lines is limited, and delivering enough electricity to the consumers’ side is challenging. One solution to this dilemma is establishing micro-grids that produce electric energy at the consumers’ side thereby solving the transmission problem. Moreover, renewable energy resources such as solar and wind have provided opportunities for final customers to install a small power generator to meet their own needs and these micro-grids can also sell electricity to the main grid. By contrast, traditional electricity supply chains have a centralized structure in which power plants were the only power generation source. Therefore, the deployment of micro-grids has decentralized the structure of electricity supply chains in some parts of the world, such as North America, Europe, and Australia.

1.2 Deregulation of Electricity Supply Chain

The idea of deregulation of the electricity supply chain emerged in the 1970’s to promote power generation competition. Over the last few decades, many cities and states in North America, Europe, and Australia have changed the electricity market structure from regulated to deregulated. In the regulated markets, utilities own and manage all electricity supply chain operations, and regulatory/governmental entities determine electricity prices. In other words, from the power production to the meter, the utility has total authority. The utility company owns the whole infrastructure and makes sure that power is generated, sent to the grid, and delivered to customers. This type of market is often considered a monopoly due to its limitations on consumer choices. The benefits of having a monopoly in the market are stable prices and long-term certainty.
Deregulation of the electricity market takes some of the ownership away from the utility. In a deregulated market, the utility controls distribution, maintenance of the grid, and invoicing of the consumer for those services. Also, electricity retailers deliver electricity to the final customers. However, large consumers, such as those who buy electricity more than a certain level, can directly buy electricity from the market. As demonstrated in figure 1.1, the deregulation process breaks the vertically integrated electricity supply system structure and makes a horizontal restructuring of power systems. The generation power plants use different resources to produce electricity where the most dominant resources are oil, natural gas, hydro, nuclear, wind, and solar. In a deregulated market, there is competition between the electric power suppliers and as a result, consumers can choose who to buy their electricity from.

In a deregulated electricity supply chain, an independent system operator manages it and the retailers’ business revolves around products and services that involve electricity supply to residential, industrial, and commercial customers. The retailers set electricity prices for consumers, often referred to as the supply portion of the electricity bill, and consumers benefit by being able to compare the rates and services of other power providers.
1.3 Various Markets in Electricity Supply Chain

There are three important markets in the deregulated electricity supply chains: day-ahead, intra-day, and balancing markets. The day-ahead market is the main place for everyday matching electricity supply and demand. The day-ahead electricity market is responsible for setting electricity prices and scheduling electricity buyers and sellers for the next day. Power plants inform the day-ahead market about their hourly capacity and the price that they are willing to sell their electricity for. Likewise, electricity distributors offer quantities and prices for the electricity they need to buy for the next day. Generation bids and consumption offers are placed every day simultaneously, and no one knows about other bids and offers. A centralized market-clearing algorithm decides about submissions and offers that are retained. Market clearing is the process of matching requests and offers from power producers and distributors, and setting electricity prices. Eventually, the system operator is informed about the trades that occurred.

The centralized algorithm to find the electricity price and schedule the power sellers and buyers is social welfare maximization. The colored area in figure 1.2 indicates social welfare restricted by supply and demand lines. As demonstrated in figure 1.2, consumption offers are ranked in decreasing price order. In contrast, supply offers are ranked in increasing price order. The arrangement defines merit order, and the point where two graphs cross is the equilibrium point between supply and demand and determines electricity market-clearing price. The equilibrium point is that it allows us to maximize social welfare. The intra-day market is a continuous trading platform between day-ahead and real-time operation. This market allows day-ahead market participants to correct their original schedules. For example, if a renewable power plant receives new data on wind power forecasts, it can change its schedule accordingly. Finally, there is a balancing market that is very close to the time of operation. The balancing market can be considered a tool for system operators to ensure that the balance between supply and demand is kept.
1.4 Motivation

Compared to the regulated markets, the deregulated markets give customers more control over decision-making and provide renewable power producers more opportunities to participate in power production. Consequently, fluctuations of electricity prices in deregulated markets are much higher with larger volatility than the regulated markets. As a result, electricity price forecasting plays an important role in the electricity markets. Recently, there has been a growing interest in studying probabilistic electricity price forecasting (PEPF) as it provides more information about future electricity prices than the point forecasts.

1.5 Literature Review

Neural networks and time series analysis methods have been dominant approaches in the PEPF context. In 2003 the research on PEPF began when Zhang et al. [1] derived prediction intervals.
(PI) of electricity prices from a cascade neural network model. Later, Zhang and Luh [2] improved the NN model results by adding a U-D factorization method within a decoupled extended Kalman filter. A hybrid PI forecasting method including bootstrapped neural networks and a GARCH model was developed by Khosravi et al. [3] and was used to derive PEPF for the electricity prices of New York and Australia. The parameters of the GARCH model are estimated by minimizing a PI-based cost function and the authors claim that the hybrid method generates narrow PIs with a large coverage probability. In addition, the authors suggest that the hybrid method generates narrow PIs with a large coverage probability. In a follow-up paper [4], Khosravi et al. applied historical electricity prices of the Australian market to a neural network to calculate point forecast of electricity prices. The authors used delta and bootstrap methods to derive PIs and concluded that while NN prediction errors are relatively large, bootstrap PIs are narrower than delta PIs.

The general training algorithms such as backpropagation could be inefficient and trap NN in the local optima, resulting in high forecasting error. To overcome this issue, Chen et al. [5] improved NN point forecasts' accuracy by applying an extreme learning machine to train a NN. They also used the bootstrap method to calculate the PIs of future electricity prices of the Australian electricity market. In related work, [6], the extreme learning machine is used in a hybrid structure to compute point forecasts of half-hourly Australian electricity prices and bootstrap-based NN calculated PIs. Other types of machine learning methods such as support vector machines have also been used to calculate point forecast, as well as PIs of electricity prices by Zhao et al. [7].

A Pareto optimal approach combined with an extreme learning machine and non-dominated storing genetic algorithm is proposed to construct PIs of electricity prices [8]. The Pareto optimal PIs are computed by solving an optimization problem that aims to find PIs with the best reliability that maximizes empirical coverage probability and the best sharpness that minimizes average width. Prediction intervals of electricity prices of Ontario and Australian markets are obtained using a NN, which is fed by preprocessed data [9]. A wavelet transform decomposes the historical
electricity prices data to improve the model accuracy. The NN is also trained by an extreme learning machine and an improved clonal selection algorithm. Eventually, the bootstrap technique is used to compute PIs.

A novel method was developed by Chai et al. [10] where a neural network was trained and generated the point forecasts of electricity prices. In addition, an ensemble of point forecasts created by the NN is then applied to a block of ensemble model output to compute PEPF. The sharpness and reliability of the PEPF were evaluated using a continuous ranked probability score. The results indicated that the Logistic distribution is more robust than the normal distribution for capturing future electricity prices’ uncertainty.

Applications of time series analysis methods in PEPF were proposed for the first time in 2006 by Weron [11]. The performance of different models such as AR, ARX (X denotes the exogenous variable such as electricity load), threshold AR (TAR), TARX, GARCH in PEPF was compared in [12] where TAR/TARX outperformed other methods. Further, parametric and semiparametric time series models were used to calculate PIs of electricity price in California and the Nordic market by Waren and Misiorek [13]. Their analyses indicate that semiparametric models have better performance than parametric ones, particularly when electricity load is considered an exogenous variable. Another semiparametric model is used in [14] to forecast the density of electricity prices by employing a time-adaptive quantile regression model for generating 5-95 percent quantiles and an exponential distribution for the tails. Compared with the GARCH model, the semiparametric model generates more reliable estimates of quantiles.

In order to forecast the distribution of electricity price, the class of Generalized Additive Models for Location, Scale, and Shape (GAMLSS) was implemented by Srilandi [15]. After testing the GAMLSS approach on California and the Italian electricity market, Srilandi concludes that GAMLSS is a flexible alternative to linear and nonlinear stochastic models. In another attempt to compute electricity prices distribution, a vector autoregressive model of order 1 with exogenous variables and skew-t-distributed disturbance was developed in [16]. The parameters of the model
were estimated by the Markov Chain Monte Carlo method, and the model was tested on three years of data of hourly Australian electricity prices. The reliability and sharpness of the forecasted distribution were evaluated by computing the continuous ranked probability score.

Stochastic volatility models for PEPF have been studied in [17] where PIs of future electricity prices of PJM market (interconnection of Pennsylvania-New Jersey-Maryland) were computed using a predictive model with double exponential distribution. Moreover, a non-Bayesian autoregressive model was investigated in [13] and evaluated the accuracy of PIs by using unconditional and conditional coverage tests. The results of [13] revealed that the Bayesian model outperforms non-Bayesian ones.

PEPF attracted much attention in 2014 when 581 participants from 61 countries competed in the Global Energy Forecasting Competition (GEFCom2014), which focused on probabilistic forecasts of electricity load and price and renewable energy. The top four contestant teams that obtained the best results published their works in the 2016 special issue of the International Journal of Forecasting. The ranked first team [18] developed a novel method for PEPF based on quantile regression and a generalized additive model (GAM). Compared with some of the statistical and machine learning methods, the GAM method generated more accurate PEPF.

The second winning team [19] extended the quantile regression averaging (QRA) method that was originally proposed in [20] and developed a hybrid model for PEPF. The model was accurate and flexible so that expert’s opinions on future electricity prices could be incorporated into the forecasting process. Dudek’s work [21] which was ranked 3rd in the competition, used a simple multilayer perceptron with a single hidden layer. The system and zonal electricity load, as well as their squared values, were considered as input data. Five sigmoid neurons processed the input data, and a linear neuron generated the point forecasts of electricity prices for the next 24 hours. The model was trained based on the hourly electricity load of 13 previous days. In order to compute PEPF, Dudek assumed a normal distribution for the forecasting errors. Similar to Dudek’s model, the forecasting model of the ranked fourth team [22] obtained PFEP using...
multiple quantile regression, transforming input data by a radial basis function, and solving the quantile regression minimization problem using an alternative direction method of multipliers.

Recently, Hubicka et al. indicate that taking average 24-hour ahead electricity price forecasts of a predictive model over different calibration windows generates more accurate forecasts than choosing the optimal window size [23]. They determined that accuracy can be achieved by averaging over a few selected window lengths. The idea of averaging across various calibration windows is extended by quantile regression averaging method to the point forecasts of day-ahead electricity prices derived by different window sizes [24]. Probabilistic forecasts of electricity prices are computed by QRA, and the results are compared based on the aggregate pinbal score and the test of predictive ability.

1.6 Gap in Research

The literature review on electricity price forecasting shows that QRA is a powerful method for quantile forecasting of electricity prices because the QRA method uses a linear regression method to derive quantile forecasts. To the best of my knowledge, no one has applied non-linear regression methods to improve the performance of QRA. Hence, this thesis aims to apply non-linear regression methods to a pool of point forecasts of electricity prices to compute quantile forecasts of day-ahead electricity prices. In order to make a pool of point forecasts, I use three different methods, such as triple exponential smoothing, deep neural networks, and gradient boosted trees. Finally, quantile regression averaging, gradient boosting regression, and a neural network are applied to the pool of point forecasts to compute quantile forecasts of DA electricity prices.
1.7 Thesis Outline

This thesis is organized into seven chapters in which chapter 2 describes some of the machine learning models for time series analysis. Chapter 3 presents volatility analysis of the Ontario electricity market. Chapter 4 uses machine learning methods to compute day-ahead forecasts of the electricity demand of Ontario. Chapter 5 creates a pool of point forecasts for day-ahead electricity prices and uses non-linear regression methods to compute quantile forecasts of electricity prices. In chapter 6, probabilistic forecasts of electricity prices are used to find optimal decisions for electricity buyers and sellers. Finally, the thesis is concluded in chapter 7.
Chapter 2

Machine Learning Models for Time Series Forecasting

2.1 Lasso Regression

Linear regression models consider a linear relationship between independent variables and the dependent variable. When there is one independent variable, the relationship is a simple line. With higher dimensions, the relationship is defined by a hyperplane. The optimal coefficients of the model minimize the sum of squared errors between the predictions and the actual values of the dependent variable. The obtained coefficients may define the relationship between the variables correctly, but the model’s performance may decline for unseen data. In more detail, the linear model’s estimated coefficients can become large, making the model sensitive to inputs and possibly unstable for the unseen data. One approach to address the stability of regression models is to include additional costs for a model that has large coefficients. A common approach to overcome the sensitivity issue is to penalize a model based on the sum of the absolute coefficient values and is called the L1 penalty. The L1 penalty minimizes the sum of all coefficients. The model is called least absolute shrinkage and selection operator (Lasso) regression and can be
written as

\[ Y = X\beta + \beta_0 + \epsilon, \]

where \( X \) is a matrix and consists of all the independent variables and can be shown as

\[
X = \begin{bmatrix}
x_{11} & \cdots & x_{1m} \\
x_{21} & \cdots & x_{2m} \\
\vdots & \ddots & \vdots \\
x_{n1} & \cdots & x_{nm}
\end{bmatrix}.
\]

(2.1)

In the machine learning context, the independent variables are also called feature data. The \( k^{th} \) row of \( X \) is a sample of feature data and denoted by \( x_k \). For the time series data, \( x_k \) refers to all of the feature data at time \( t = k \). Equation 2.1 presents the \( j^{th} \) feature data at time \( t = k \) by \( x_{kj} \). Moreover, \( j \) and \( k \) have maximum of \( m \) and \( n \), respectively. Artificial intelligence machines use the feature data to predict observations \( Y \) which can be expressed as

\[ Y = [y_1, y_2, \ldots, y_n]^T, \]

where \( T \) denotes the matrix transpose operator. Furthermore, \( \beta_0 \) is the intercept and \( \beta \) is the regression coefficient vector as

\[ \beta = [\beta_1, \ldots, \beta_m]^T, \]

In addition, \( \epsilon \) is a vector of error terms defined as

\[ \epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]^T, \]

and captures all other factors which influence the dependent variable \( Y \) other than the independent variables \( X \). According to [25], the Lasso estimate \( \hat{\beta} \) of the linear regression model
parameters is given by

\[ \hat{\beta} = \arg\min_{\beta} \frac{1}{\lambda_r \sqrt{n}} \sum_{i=1}^{n} \epsilon_i^2 + \sum_{j=1}^{m} |\beta_j|, \]

and \( \lambda_r \) is regularization parameter.

## 2.2 Support Vector Regression

Regression can be implemented by classification methods as well. For example, all the training data can be classified into different groups; each determines expected observations based on all independent variables. Support vector classifier uses cross-validation to find the best margin that can separate different classes. Cross-validation is a way to find out how well the training algorithm is working. For example, 10-fold cross-validation divides the training data into ten separate groups, uses nine of them to train the algorithm, and evaluates its validity on the last group. This process repeats nine more times, and each time a group is selected as a testing set the other ones are considered the training set. This process ensures that the obtained classifier is the optimum one and will work well for unseen data. The basic ideas behind support vector machines are:

- classifying data with relatively low dimensions,
- moving the data to a higher dimension,
- finding a support vector classifier that separates higher dimension data into proper groups.

Support vector machine uses kernel functions in order to find support vector classifiers in higher dimensions. Indeed, Kernel functions only calculate the relationships between every pair of points as if they are in higher dimensions but they do not actually do the transformation. Calculating the high-dimensional relationships without transforming the data to the higher dimensions is called the kernel trick. This trick reduces the amount of computation required for support vector machines by avoiding the math that transforms the data from low to high dimensions.
Support vector regression (SVR) is another type of regression that uses a support vector machine to find the relationship between dependent and independent variables. This regression method defines an acceptable error threshold for the Kernel trick to find a proper hyperplane to fit the data. In contrast to the Lasso regression that focuses on minimizing the differences between predicted values and the actual observations, SVR determines the optimum relationship by minimizing the following statement [26]

$$\frac{1}{2} W^T W + C \left( \sum_{i=1}^{n} (\zeta_i^{(a)} + \zeta_i^{(b)}) \right)$$

Subject to

$$y_i - f(x_i, W) \leq \epsilon + \zeta_i^{(a)}$$
$$f(x_i, W) - y_i \leq \epsilon + \zeta_i^{(b)}$$
$$\zeta_i^{(a)}, \zeta_i^{(b)} \geq 0$$
$$i = 1, 2, \ldots, n$$

The $f(.)$ function is supposed to find the best fit for the data and is described as

$$f(x_i, W) = W^T \phi(x_i) + b,$$

where $b$ is an intercept, $W = \{w_1, \ldots, w_k\}$ is an $k \times 1$ coefficients vector and

$$W^T \phi(x_i) = w_1 \phi_1(x_i) + \cdots + w_k \phi_k(x_i),$$

where $\phi_i(.)$ is the function (such as Kernel function) that increases data dimension to reach a proper fit. In addition, $\epsilon$ indicates the acceptable margin for error, and the distance of data that are spread above or below of the margin is represented by $\zeta_i^{(a)}$ and $\zeta_i^{(b)}$ respectively.
2.3 Gradient Boosting Regression

Gradient boosting regression aims to predict a dependant variable using decision trees. A decision tree is a tree-like model of decisions and their possible consequences. The model has a flowchart-like structure in which

- each internal node represents a test on an attribute,
- each branch represents the outcome of the test,
- each leaf node represents a decision taken after computing all attributes.

The paths from the root to the leaf represent the regression procedure. Each tree uses the independent variables and uses previous trees’ forecasting error to generate a more accurate prediction than previous trees. More specifically, the first attempt of GBR is taking an average of dependant variables as the initial prediction value. Then, GBR calculates the residual of the prediction and builds trees to predict residuals. The obtained residuals then add to the last prediction to provide a new prediction. Predicting the residuals is repeated until no further improvement is obtained.

In order to model the GBR method mathematically, assume \( \{(x_i, y_i)\}_{i=1}^{n_x} \) to be the training data set in which \( x_i \)'s are features and \( y_i \)'s are observations. The goal is to find a predictive function such as \( F(.) \) that computes prediction of \( y_i \) as

\[
\hat{y}_i = F(x_i).
\]

A differentiable loss function \( L(.,.) \) such as

\[
L(y_i, F(x_i)) = \frac{1}{2}(y_i - F(x_i))^2, \quad (2.2)
\]
is needed to calculate the residuals. As described in [27], the first step in implementing GBR is to initialize the model with a constant value

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n_s} L(y_i, \gamma),$$

where $\gamma$ is the initial predicted value and equals the average of all observations. The GBR algorithm makes $M$ trees to compute predicted values for each observation in the second step. Each tree calculates the pseudo-residuals

$$r_{im} = -\left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x) = F_{m-1}(x)},$$

where $i$ is the sample number, and $m$ is the tree that the algorithm is trying to build. It should be mentioned that $r_{im}$ is called pseudo-residuals because it is not the same as residuals from a linear regression. The next task in step two is to build a tree that uses feature data, $x_i$s, to predict pseudo-residuals. The $m^{th}$ tree has a total of $j_m$ leaves. The tree uses the feature data to categorize $r_{im}s$ into the different leaves. The pseudo-residuals falling under each leaf create a terminal region represented by $R_{jm}$ and $j$ is the leaf index. The tree attributes an output value for each leaf as

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma).$$

(2.3)

It can be proved that the optimal value for $\gamma$ that minimizes the right-hand side of the equation 2.3 is the average of the elements in the corresponding terminal regions. Eventually, GBR makes a new prediction for each sample in the data set based on the previous prediction and output of the recent tree

$$F_m(x_i) = F_{m-1}(x_i) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x_i \in R_{jm}),$$

where $\nu$ is the learning rate. A small learning rate reduces the effect that each tree has on the final prediction. The principle of gradient boosting regression for making trees is summarized in
Algorithm 1 Gradient Boosting Regression

1: Initialize model with a constant value: $F_0(x) = \arg\min_\gamma \sum_{i=1}^{n_s} L(y_i, \gamma)$

2: for $m = 1, 2, \ldots, M$ do

3: Compute $r_{im} = -\left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x) = F_{m-1}(x)}$

4: Fit a regression tree to the $r_{im}$ values

5: for $j = 1, 2, \ldots, J_m$ do

6: Create terminal regions $R_{jm}$

7: $\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

8: end for

9: $F_m(x_i) = F_{m-1}(x_i) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x_i \in R_{jm})$

10: end for

11: Return $F_M(x)$

A more sophisticated version of the gradient boosting method is called extreme gradient boosting trees. The XGB regression uses the same loss function used in GBR to build trees by calculating the optimal output values so that

$$\gamma = \arg\min_\gamma \sum_{i=1}^{n_s} L(y_i, F(x_i)) + \frac{1}{2} \lambda \gamma^2,$$

the first part is the loss function, and the second part consists of a regularization term. The goal is to find an output value, $\gamma$, for the leaf that minimizes the whole equation. XGB regression uses the second-order Taylor approximation to minimize the total loss function in equation 2.4 as

$$L(y_i, F(x_i) + \gamma) \approx L(y_i, F(x_i)) + G_i \gamma + \frac{1}{2} H_i \gamma^2,$$

where $G_i$ (Gradient) and $H_i$ (Hessian) are the first and second derivatives of the loss function. It can be proved that the optimal $\gamma$ for a leaf that minimizes the loss function is

$$\gamma = \frac{\sum_{i=1}^{n_s} G_i}{\sum_{i=1}^{n_s} H_i}.$$
Other characteristics that make XGB a powerful tool for regression and classification are presented in the original paper [28].

2.4 Artificial Neural Networks

An artificial neural network (ANN) is a computing unit created artificially to simulate biological neural networks. The name neural network derives from neurons which are the brain cells that extend through the nervous system. Neurons communicate with each other via electrical/chemical signals. These simple interactions form all of human thoughts and actions. Artificial intelligence can be created by simulating and connecting neurons with computers. Figure 2.1 represents a simple brain simulation that fully connects neurons in input, hidden, and output layers. Since the signals go from the left side of the network to the right side, the network is called a feed-forward neural network. Feed-forward neural networks are the most basic ANNs and form a basis for convolutional and recurrent neural networks. The hidden and output layers in figure 2.1 are called dense layers because they are fully connected to their previous layers.

![Figure 2.1: Structure of feed-forward neural network.](image_url)

Input, hidden, and output layers are the building blocks of ANNs’ structure. Assume an ANN model that takes $\mathbf{x}_i$ as an input vector to compute forecasts of $y_i$. The input layer passes
\( \mathbf{x}_i \) to a hidden layer with one neuron which applies its weights \( W_h \) and bias \( b_h \), as well as an activation function \( f_h(.) \) (such as sigmoid, tanh or Relu) to generates \( h_i \) as the neuron’s output

\[
h_i = f_h(W_h^T \mathbf{x}_i + b_h).
\]

Assume that the output layer has just 1 neuron which receives \( h_i \) and computes \( \hat{y}_i \) using its own weights \( W_o \), bias \( b_o \), and activation function \( f_o(.) \) as

\[
\hat{y}_i = f_o(W_o^T h_i + b_o). \tag{2.7}
\]

The weights and biases of neurons are tuned as the learning process proceeds. The number of neurons and layers in the model is obtained using a trial and error method. Usually, the networks with more than two hidden layers are called deep networks and can capture important features in the data. However, simply connecting neurons and making a network is not enough for creating artificial intelligence. Gradient descent is the method that has been widely used to train ANNs and give them a sense of artificial intelligence.

### 2.5 Gradient Descent

Training algorithms for ANNs seek to find the optimal weights for the neurons that minimize a loss function. The cost function represents the deviation of predicted values from the actual ones. For example, mean squared error (MSE) can be considered as the loss function and is defined as

\[
\mathcal{L}(W) = \frac{1}{n_s} \sum_{i=1}^{n_s} (y_i - \hat{y}_i)^2,
\]

where \( n_s \) is the number of samples in the training data. The general rule of sum for making training data set is to allocate 80% of data for training and the rest of them for testing purposes.

In the training data, there are feature data denoted by \( \mathbf{x}_s \) and the corresponding observation \( \mathbf{y}_s \).
The loss function is a function of neurons’ weights, $W$. The goal is to minimize the loss function $\mathcal{L}(W)$ with respect to the parameter $W$. The general way to approach this problem is to take the derivative of $\mathcal{L}(W)$ with respect to $W$ and set it to zero. Solving the equation provides the neurons’ weights that minimize the loss function. Sometimes the equation is not simply solvable, and it needs to be approximated numerically. The idea behind gradient descent is to compute the derivative of the loss function and then repeatedly take small steps in the direction of the gradient to find a new $W$. At each step, $\mathcal{L}(W)$ decreases provided that the step size is small enough. Eventually, the gradient descent algorithm converges to the minimum. Here, convergence means that the steps become so small that the decrease in loss function becomes insignificant. In other words, neuron weights do not change as the algorithm takes further steps. At that point, the algorithm finds a set of neurons’ weights that minimize the loss function.

**Algorithm 2** Gradient Descent

| Initialize $W_0$ with a random value: |
| for $t = 1, 2, \ldots$, Epoch do |
| $G_t = \nabla \mathcal{L}(W_{t-1})$ |
| $W_t = W_{t-1} - \eta G_t$ |
| end for |
| Return $W_t$ |

Algorithm 2 shows how the gradient descent algorithm works. Epoch and $\eta$ are two important hyperparameters in the algorithm. Epoch is the number of iterations that the algorithm should run the loop and should be high enough so that the loss function converges. Further, the learning rate, $\eta$, must be initialized properly. A large $\eta$ makes the algorithm jump over minima, and a very small $\eta$ either takes too long to converge or gets stuck in an undesirable local minimum. The proper values for the number of epochs and $\eta$ are chosen in a trial and error process. In algorithm 2, $G_t$ is the gradient, $\nabla(.)$, of the loss function that is calculated in each iteration.

When the size of training data is large, computing the loss function for all the data takes much time. One solution to the problem is stochastic gradient descent, which uses a randomly selected batch of the data at every step rather than the entire data. As a result, computing the
derivatives of the loss function takes much less time. The stochastic gradient descent algorithm is shown in algorithm 3.

Algorithm 3 Stochastic Gradient Descent

\[
\begin{align*}
\text{for } t = 1, 2, \ldots, \text{Epoch} & \text{ do} \\
\quad x_s, y_s & = \text{Shuffle}(x_s, y_s) \\
\quad \text{for } j = 1, 2, \ldots, \frac{n_s}{\text{BatchSize}} & \text{ do} \\
\qquad \text{Start} & = j \times \text{BatchSize} \\
\qquad \text{End} & = (j + 1) \times \text{BatchSize} \\
\qquad x^b_s & = x_s[\text{Start} : \text{End}] \\
\qquad y^b_s & = y_s[\text{Start} : \text{End}] \\
\qquad & \text{Apply gradient descent algorithm to } x^b_s \text{ and } y^b_s \text{ and find the optimum } W_t. \\
\text{end for} \\
\text{end for} \\
\text{Return } W_t
\end{align*}
\]

Instead of one loop over the epochs, stochastic gradient descent has two nested loops, one for the epochs and one for the training batches. As mentioned, the idea behind the stochastic gradient descent is to split the training data into batches and to apply gradient descent to each of them. Batch size is usually a small number such as 32, 64, or 128. Accordingly, the number of iterations of the inner loop is \(n_s\) divided by the batch size. On each epoch, the algorithm makes one pass through the whole training data. It is also important to randomize each epoch’s data because running through the same sample with the same order each time could lead to undesirable patterns being learned.

2.6 Momentum

Momentum is a significant improvement over the plain stochastic gradient descent and can improve the learning procedure. By analogy, momentum can be considered as pushing a box on a low friction surface such as ice. The box can move on ice easily without much force because the momentum carries the box the way the box was going before. However, the ice still has some friction, and the box eventually slows down and stops after a while. Momentum in gradient descent
also slows down after a while. To continue with the analogy, we can think of a situation without momentum. For example, the box moves on a high friction surface such as sand as long as a source of force is applied to the box. The situation is like gradient descent without momentum. If we want to move on the error surface, there needs to be a gradient to move in the direction of the gradient. In order to express the analogy mathematically, it was mentioned that the regular gradient descent without momentum is expressed as

\[ W_t = W_{t-1} - \eta G_t. \]  

(2.8)

When there is no gradient, \( G_t = 0 \), there is no movement, and consequently, there is no update for neurons’ weights. In order to model gradient descent with momentum, the \( \eta G_t \) in equation 2.8 can be replaced with a velocity term \( V_t \) such as

\[ V_t = \mu V_{t-1} - \eta G_t, \]

where \( \mu \) is momentum, and the initial value of velocity is zero. Typical values for \( \mu \) are 0.90, 0.95, or 0.99, which indicates that the new velocity has a fraction of the old velocity. The new velocity is used to update the neurons’ weights as

\[ W_t = W_{t-1} + V_t. \]

Momentum helps the training procedure to converge to optimal neurons’ weights much faster than regular gradient descent.

### 2.7 Variable and Adaptive Learning Rates

The learning rate, \( \eta \), appears in the training algorithms as a constant value. However, the learning rate can be a function of time (iteration) denoted by \( \eta(t) \) and is also called learning...
rate scheduling. The simplest rate scheduling is called step decay and it periodically reduces the learning rate by a constant factor (for example, every 100 steps). Another method is exponential decay, in which the learning rate follows an exponential curve. There are other types of learning rate methods, and in all of them, the learning rate decreases with time.

Generally, when a neural network’s weights are initialized randomly, they are very far from the optimal weights. Perhaps, it is better to start with a large learning rate so that the algorithm can take bigger steps towards the goal. Similarly, there is the same motivation in momentum in which gradient descent picks up speed by accumulating past gradients. If the initial value is far from our goal, the gradients should be large. However, when the algorithm gets closer to the goal, the gradient is going to shrink. By definition, the minimum of a function necessitates a gradient of zero. If the algorithm keeps using big steps when it is close to the goal, the algorithm overshoots, and bounces back and forth. Therefore, in order to reduce bouncing around, the algorithm needs to take smaller steps as it approaches the goal.

The variable learning rate can speed up the convergence of the gradient descent algorithm. However, a hyperparameter optimization is still required to determine when to decrease the learning rate. Adaptive learning rate techniques seem to be more efficient than variable learning rate methods. Adaptive gradient algorithm (AdaGrad) is a simple adaptive learning rate method and is based on the fact that the dependence of the loss function on each neuron’s weight is not the same. In other words, the gradient can be steep in one direction and flat in another. So, perhaps it may be beneficial to adapt the learning rate for each neuron’s weight individually based on how much it has changed in the past. The AdaGrad introduces a new variable called cache and is shown by $C_t$. Each neuron of the neural network has its cache; where $C_t$ has the same dimension as $W_t$. The idea behind the cache is that it accumulates the squared gradients as

$$C_t = C_{t-1} + G_t \odot G_t,$$

in which $\odot$ represents element-wise multiplication. Because the squared gradient adds to the
cache at each step, the cache is always positive. If a parameter has a large gradient in the past, its cache will be very large, leading to a very small learning rate. Therefore, the parameter will change slowly in the future. On the other hand, when a parameter has a lot of small gradients in the past, the cache will be small, and its learning rate will remain large. AdaGrad updates weights of neural networks as

$$W_t = W_{t-1} - \frac{\eta}{\sqrt{C_t} + \epsilon} \odot G_t,$$

(2.9)

In other words, each neuron’s weight and its learning rate is updated independently of the others. In addition, $\epsilon$ is a small value such as $10^{-8}$ or $10^{-10}$ and is used to avoid dividing by zero.

The main drawback of AdaGrad is the fact that this method decreases the learning rate too aggressively. The reason that AdaGrad decreases the learning rate too quickly is because the cache is rising too fast. The Root mean square propagation (RMSProp) method is an extension of AdaGrad and restricts the growth of the cache by decreasing it at each update as

$$C_t = \lambda_d \times C_{t-1} + (1 - \lambda_d) \times G_t \odot G_t,$$

(2.10)

where $\lambda_d$ is the decay rate and is a number close to 1 such as 0.99, 0.999, etc. Eventually, RMSProp uses equation 2.9 to update neuron weights.

According to equation 2.10, the cache can be interpreted as the estimation of the expected value of the squared gradient,

$$C_t \approx E(G_t \odot G_t).$$

Adaptive moment (Adam) estimation uses $m_t$ and $s_t$ as the 1st and 2nd moments of $G_t$ in order to find optimal values for learning rates. The exponentially smoothed averages of $m_t$ and $s_t$ can be written as

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) G_t,$$

and

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2) G_t \odot G_t.$$
and

\[ s_t = \beta_2 s_{t-1} + (1 - \beta_2) G_t \odot G_t, \]

in which \( \beta_1 \) and \( \beta_2 \) are coefficients between 0.9 and 0.999. Adam is a modern adaptive learning rate technique that is the default for many deep learning practitioners and employs the same cache mechanism that is used in RMSProp. The exponentially-smoothed average method needs an initial value for \( m_0 \) and \( s_0 \) to calculate updates for \( m_t \) and \( s_t \), respectively. We typically set the initial value to zero and by doing so, \( m_t \) and \( s_t \) are biased toward zero. Adam uses a technique called bias correction to overcome the initialization problem. Adam applies bias correction for exponentially smoothed averages of \( m_t \) and \( s_t \) as

\[ \hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \]

and

\[ \hat{s}_t = \frac{s_t}{1 - \beta_2^t}. \]

Again, \( \epsilon \) is a small number such as \( 10^{-8} \) and is used to avoid division by zero. Eventually, the Adam update of neurons’ weights can be formulated as

\[ W_t = W_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{s}_t} + \epsilon}. \]

2.8 Recurrent Neural Networks

Recurrent neural network (RNN) is another type of ANNs equipped with memory that enables them to apply the previous neuron’s output to compute future outcomes. This particular feature makes RNN a powerful tool for time series forecasting. The simple predictive model described in equation 2.7 can be considered a linear regression model. A non-linear forecasting model can be applied by using a simple RNN model for time series, which is almost the same as ANN. The
hidden state in RNN is a non-linear function of the input. That is

$$h_i = f_h(W_{xh}^T x_i + W_{hh}^T h_{i-1} + b_h),$$

where $W_{xh}^T$ is a vector of weights connecting the input to the RNN neuron, and $W_{hh}^T$ refers to the feedback link from the lagged value of RNN output to its input. At first glance, the simple RNN unit considers all the previous input data to calculate the model output. However, simple RNNs are very vulnerable to the vanishing gradient problem, and they have short memories. In other words, the vanishing gradient problem causes the RNN model to forget what it has learned from lagged values of the input. The most effective way to deal with this problem is to use a more sophisticated unit such as long short-term memory (LSTM) or GRU. GRU is like a simplified version of LSTM with fewer parameters and thus more efficient.

Compared to equation 2.11, the hidden state for a GRU unit is more complicated and depends on update gate vector $z_i$ and reset gate vector $r_i$, which can be written as

$$z_i = S(W_{xz}^T x_i + W_{hz}^T h_{i-1} + b_z),$$

and

$$r_i = S(W_{xr}^T x_i + W_{hr}^T h_{i-1} + b_r),$$

where $S$ is a Sigmoid function and indicates that $z_i$ and $r_i$ vary between 0 and 1. The GRU hidden state $h_i$ is expressed as

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \tanh (W_{xh}^T x_i + W_{hh}^T (r_i \odot h_{i-1} + b_h))$$

where $\odot$ means element-wise multiplication as the two gates ($r_i$ and $z_i$) and the hidden vector have the same dimension. It follows from equation 2.12 that if the reset gate is neglected, $h_i$
turns out to be a weighted sum of the previous hidden state and output of a simple RNN. The role of the reset gate in equation 2.12 is reinforcing the ability of the GRU unit to remember and forget different parts of $h_{i-1}$. LSTM units have almost the same architecture as GRUs but they are different. In addition to the hidden state $h_i$, LSTMs have an extra state, $c_i$, called a cell state. LSTM structure can be considered the connection of some neurons to perform either as gates or generate cell states. The cell state

$$c_i = f_t \odot c_{i-1} + i_t \odot \tanh(W_{xc}^T x_i + W_{hc}^T h_{i-1} + b_c),$$

where $f_t$ is the forget gate given by

$$f_t = S(W_{xf}^T x_i + W_{hf}^T h_{i-1} + b_f),$$

and $i_t$ is input/update gate given as

$$i_t = S(W_{xi}^T x_i + W_{hi}^T h_{i-1} + b_i).$$

Equation 2.13 indicates that LSTM offers the cell state to have the opportunity to remember its old states so that the unit learns from long-term dependencies. Eventually, the output gate $o_i$:

$$o_i = S(W_{xo}^T x_i + W_{ho}^T h_{i-1} + b_o),$$

which is a simple RNN unit (the same as the other 2 LSTM gates) is used to compute the hidden state through a simple transformation on the cell state

$$h_i = o_i \odot \tanh c_i.$$
2.9 Convolutional Neural Networks

Recurrent neural networks are great tools for time series forecasting, particularly for those series with high autocorrelation. It may seem reasonable to use a large amount historical data in order to compute forecasts of future values. For example, using 168 hours (one week) of hourly electricity price data to compute day ahead forecasts of this variable. However, the performance of the recurrent neurons gets worst as the dimension of the input data increases. Therefore, a pre-processing step must extract the essential features from input data and map them into a low dimensional space.

In time series forecasting, applying smoothing techniques to the data improves the predictive power of the models. For example, when dealing with noisy measurements, a simple smoothing method such as a moving average can cancel out noise from data to some extent. Similarly, when there are serial dependencies in the data, a weighted moving average approach is required to extract features. Convolutional neural network (CNN) is a type of feedforward neural network used vastly in image and natural language processing due to its good performance. Convolutional neural networks can smooth the input data and extract features from this data so that the following layers focus on more essential data with much less dimension. For instance, a convolution layer that receives 1-dimensional input data with a length of 168 may divide it into seven subsets where each has 24 elements. Eventually, by training the neural network, CNN maps the $(168 \times 1)$ input data to a $(7 \times 1)$ output.

Figure 2.2 shows how a Convolutional neural network works. The input data, which is a time series with length $L$, goes through some filters. The number of filters, $n_f$, is arbitrary and each filter consists of $l$ neurons with weights, biases, and an activation function. The input data is divided into some chunks named kernels where each has a length of $l$. In order to create a feature map, all of the kernels go through a filter. By changing the neurons’ weights and biases, CNN makes new filters and several feature maps. Eventually, a long input vector with length $L$ is changed to $n_f$ smaller vector each has size $l$. The feature maps are useful input data for...
2.10 Evaluation Metrics

Mean absolute error (MAE) and root mean squared error (RMSE) are used as evaluation criteria to examine the error level resulting from forecasting models. Consider $y_i$ in the testing data set with $n_t$ samples, and $\hat{y}_i$ the forecasts of the future observation. Then

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

and

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_t} (y_i - \hat{y}_i)^2}{n_t}}.$$

The R squared ($R^2$) criteria is calculated to determine what portion of the variance in data is predictable from the forecasting results. The best possible score from these criteria is 1.0, and negative values are also possible, as

$$R^2 = 1 - \frac{\sum_{i=1}^{n_t} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_t} (y_i - \bar{y})^2},$$

where $\bar{y}$ indicates the average of $y_i$. 

Figure 2.2: Structure of Convolutional neural network.

recurrant units because they are low-dimensional vectors that carry important data.
Chapter 3

Volatility Analysis

Electricity price volatility refers to the degree of variation of the electricity price over time. In a deregulated market, the price of electricity depends on buyers’ and suppliers’ decisions at each hour. Here, the failure of power units or congestion in transmission lines results in unexpected electricity price fluctuations. Overall, electricity supply and demand seem to have a substantial impact on electricity prices. In order to validate this statement, I chose the 2019 hourly Ontario electricity price (HOEP) and demand data. Table 3.1 presents some descriptive statistics of the data, including mean and standard deviation (std), minimum and maximum of HOEP, as well as three important quartiles. It seems that while most electricity prices vary between 3.00 and 15.00 CAD/MWh, according to the HOEP data, price spikes cause relatively high volatility (standard deviation) of the data. I use the Inter-Quartile Range (IQR) to define the threshold and separate regular prices from the spikes. IQR is a descriptive statistic and equals the difference between upper ($Q_3$) and lower ($Q_1$) quartiles. The spike prices are defined as the samples in the data set that fall below $Q_1 - 1.7 \times IQR$ or above $Q_3 + 1.7 \times IQR$. Based on the definition, 1.15% of the samples in the 2019 HOEP data set are the spike prices.
Table 3.1: Descriptive Statistics of 2019 Ontario Electricity Prices

<table>
<thead>
<tr>
<th>Mean</th>
<th>std</th>
<th>Min</th>
<th>Q1</th>
<th>Median</th>
<th>Q3</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.72</td>
<td>23.55</td>
<td>-59.29</td>
<td>3.02</td>
<td>14.36</td>
<td>25.56</td>
<td>1028.52</td>
</tr>
</tbody>
</table>

3.1 Determinants of the Price Fluctuations

The data analysis of the Ontario market’s electricity prices suggests a correlation of almost 0.40 between hourly Ontario electricity price and demand. The relationship between electricity prices and demand is illustrated in figure 3.1. Moreover, the relationship between wind power generation and electricity prices has a correlation of −0.07 for the 2019 data. At first glance, it appears that the impacts of renewable power plants such as wind power units on electricity prices are negligible. However, it should be noted that almost 80% of the spike prices occur when the total output of wind power units is less than 1500 MW. It could be interpreted as when wind power plants are scheduled to produce a certain amount of energy, but unexpected incidents happen. In fact, the renewable units are scheduled based on day-ahead weather forecasts. If unpredictable weather changes occur and renewable plants’ electricity production goes above or below what is expected, spike prices will occur.

The impact of date and time on electricity prices shown in figure 3.2 is a consequence of the relationship between electricity price and demand. figure 3.2 provides a sense about the temporal patterns that exist in the 2010-2019 HOEP data set. The time series plots with different scales (hourly/weekly/monthly) show that electricity has a higher price in Winter and Summer months than in other seasons. Also, electricity costs most at the beginning of the week (Monday) and it drops toward the end of the week with the lowest price on Sundays. Within the day, the electricity prices have two peaks in the morning and evening. The impact of time of the day on electricity prices could decrease the validity of volatility analysis. Dividing the hourly time series of the electricity price into 24 subsets is proposed as a solution by Zareipour et al. [29]. They calculated hourly volatility of electricity returns for the Ontario, PJM, and New England Markets and concluded that Ontario was the most volatile market in 2007 among the other two.
markets. In the following, hourly volatility of the Ontario market’s electricity prices is calculated to determine if the results [29] about the Ontario market are still valid.

### 3.2 Volatility of Hourly Electricity Prices

In order to compute the volatility of hourly electricity returns, 2019 hourly electricity prices are divided into 24 series with each has 365 elements. Mathematically, the subsets are shown by $p_{t,h}$ in which $t \in [1, 365]$ and $h \in [1, 24]$. The subsets are used to calculate the sign correlation of the electricity prices time series. The idea of sign correlation of time series is proposed in [30], and for the electricity prices is defined as

$$\hat{\rho}_h = \text{Corr}(p_{t,h} - \bar{p}_h, \text{sign}(p_{t,h} - \bar{p}_h)).$$  \hspace{1cm} (3.1)
In equation 3.1, $\hat{\rho}_h$ and $\bar{p}_h$ are sample sign correlation and mean of electricity prices at hour $h$, and Corr(.) refers to the correlation function. It can be shown that if the electricity prices series has a symmetric distribution with finite mean $\bar{p}_h$ and variance $\sigma^2_{p_h}$ then the mean observed hourly volatility, $\bar{s}_h$, is

$$\bar{s}_h = \frac{1}{365} \sum_{t=1}^{365} \frac{|p_{t,h} - \bar{p}_h|}{\hat{\rho}_h}.$$ 

Figure 3.3 demonstrates the mean hourly volatility of electricity prices of Ontario, PJM, and Nord Pool markets. Overall, the hourly volatility of Ontario’s electricity prices is much higher than in the other two markets. Hence, it can be concluded that the Ontario electricity market has high volatility compared to PJM and Nordpool electricity markets.
3.3 Structure of Ontario Electricity Market

One of the main reasons that the Ontario electricity market has higher volatility than PJM and NordPool is the difference in the markets’ designs. Ontario has a one-settlement electricity market, while the other two markets have two-settlement structures. As already explained, electricity buyers and suppliers participate in the day-ahead market and place their bids and offers. The gate closure of the day-ahead market is around noon, and after that, no offer or bid is accepted. A couple of hours later, the day-ahead market operator obtains the electricity price, $\lambda_s$, and schedules buyers and suppliers for 24 hours, starting at 12:00 am.

The intra-day market starts working as soon as buyers and suppliers find out the day-ahead market’s outcomes. In the intra-day market, buyers and suppliers can adjust their bids and offers as they are made based on forecasts and these bids can be inaccurate. For example, assume that

![Hourly Average Volatility of Electricity Prices](image)

Figure 3.3: Comparison of Ontario electricity market vs. PJM and Nord Pool markets in terms of volatility.
a wind power generator is scheduled for 60 MWh for the upcoming operation period. Based on new weather forecasts, the generator realizes that its generation will be 40 MWh. As a result, the power generator has to buy 20 MWh of electricity from another supplier. The suppliers can compensate for the deficit supply by making contracts in the intra-day market. However, the supplier may buy electricity for a price higher than $\lambda_s$.

Day-ahead and intra-day markets are two financial markets that are managed by an authority called the market operator. The market operator sets the electricity prices and schedules the buyers and suppliers but has no authority to force them to produce or purchase power. The system operator is the authority who penalizes market participants if they cannot make their commitments. The system operator manages the balancing market, which makes the balance between supply and demand. Those suppliers that get scheduled in the day-ahead market have to provide the amount they have already been assigned for. The surplus/deficit power will be sold/bought for a different price, denoted by $\lambda_b$. The electricity price in the balancing market, $\lambda_b$, determines the electricity market’s design to be one or two-settlement.

In a one-settlement situation, $\lambda_b > \lambda_c$, which indicates that suppliers sell their surplus production for a higher price. Some large-scale suppliers voluntarily reduce their production so that other suppliers can sell all of their production. Moreover, suppliers will be penalized based on balancing market price if they have lack of production. It can be expected that suppliers produce more power than they offered in the day-ahead market to increase their profit in the one-settlement markets. As a result, the fact that $\lambda_b$ is greater than $\lambda_s$ can make unexpected changes to electricity prices and increase the volatility.

On the other hand, the two-settlement design has a different approach in managing the balancing market. In the two-settlement markets, the surplus power is sold on $\lambda_s$. Also, the lack of production is penalized based on balancing market price, which is greater than $\lambda_s$. Therefore, electricity price has less volatility than it has in one-settlement markets. The electricity market in Ontario has a one-settlement design, while PJM and Nordpool are two-settlement markets.
It seems that Ontario has decided to keep its market to have a one-settlement design so that renewable power plants expand in this province.

3.4 Day-Ahead Volatility Forecasts

According to figure 3.3, electricity prices vary from hour to hour. Therefore, the day-ahead volatility forecasts of electricity prices can be considered a feature for training the machine learning-based electricity price forecasting models. In section 3.2, the observed hourly volatility of just one year is calculated. Here, 2010-2018 HOEP data are used to obtain day-ahead volatility forecasts of 2019 HOEPs. The whole data set is divided into 24 subsets; each has $N_o$ samples. Exponentially smoothing methods are applied to the observed hourly volatility for all the subsets in order to compute one-step-ahead volatility forecasts of electricity prices denoted by $\hat{\sigma}_{t+1,h}$. The exponentially smoothing methods can extract some patterns from the data. The observed hourly volatility of electricity prices:

$$s_{t,h} = \frac{|p_{t,h} - \bar{p}_h|}{\hat{\rho}_h}$$

is applied to double and triple exponential smoothing methods to compute $\hat{\sigma}_{t+1,h}$.

3.5 Double Exponential Smoothing

Double exponential smoothing (DES) method can be applied to non-stationary data sets as long as there exists a linear regression model for them as

$$s_{t,h} - s_h = \beta_{0,h} + \beta_{1,h}t + \epsilon_{t,h},$$

Where $s_h$ is the sample standard deviation of $p_{t,h}$ based on the first 30 observations, and $\epsilon_{t,h}$ is white noise. DES provides one-step-ahead forecasts of each subset via three main steps: initialization, forecasting, and updating.
Step I. Initialization

\[ S_{0,h}^1 = \hat{\beta}_{0,h} - \frac{1 - \alpha_h}{\alpha_h} \hat{\beta}_{1,h}, \]
\[ S_{0,h}^2 = \hat{\beta}_{0,h} - 2 \frac{1 - \alpha_h}{\alpha_h} \hat{\beta}_{1,h}. \]

Step II. Forecasting \((t \in [0, N_o])\)

\[ S_{t,h} = (2 + \frac{\alpha_h}{1 - \alpha_h})S_{t,h}^1 - (1 + \frac{\alpha_h}{1 - \alpha_h})S_{t,h}^2. \]

Step III. Updating \((t \in [1, N_o])\)

\[ S_{t,h}^1 = \alpha_h S_{t-1,h} + (1 - \alpha_h)S_{t-1,h}^1, \]
\[ S_{t,h}^2 = \alpha_h S_{t-1,h}^1 + (1 - \alpha_h)S_{t-1,h}^2. \]

In these steps, \(S_{t,h}\) is the smoothed value and the forecast of volatility at time \(t + 1\) for each subset. The tuning parameter is \(\alpha_h \in (0, 1)\), and its optimal value can be obtained by minimizing the one-step-ahead forecast error sum of squares

\[ \text{SSE}(\alpha_h) = \sum_{t=1}^{N_o} (s_{t,h} - s_h - S_{t-1,h})^2. \]  

Finally, the optimum value of \(\alpha_h\) is used in the all 3 steps to compute \(\hat{\sigma}_{t,h}\).

### 3.6 Triple Exponential Smoothing

Triple exponential smoothing is another forecasting method that can capture trends and seasonality in the time series. This method considers a quadratic model for the data as

\[ s_{t,h} - s_h = \beta_{0,h} + \beta_{1,h} t + \beta_{2,h} \frac{t^2}{2} + \epsilon_{t,h}. \]
The same as DES forecasting, the TES smoothed statistic can be obtained through the following steps:

**Step I. Initialization:**

\[
S^1_{0,h} = \hat{\beta}_{0,h} - \frac{\omega_h}{\alpha_h} \hat{\beta}_{1,h} + \frac{\omega_h (2 - \alpha_h)}{2 \alpha_h^2} \hat{\beta}_{2,h},
\]

\[
S^2_{0,h} = \hat{\beta}_{0,h} - \frac{2 \omega_h}{\alpha_h} \hat{\beta}_{1,h} + \frac{2 \omega_h (3 - 2 \alpha_h)}{2 \alpha_h^2} \hat{\beta}_{2,h},
\]

\[
S^3_{0,h} = \hat{\beta}_{0,h} - \frac{3 \omega_h}{\alpha_h} \hat{\beta}_{1,h} + \frac{3 \omega_h (4 - 3 \alpha_h)}{2 \alpha_h^2} \hat{\beta}_{2,h},
\]

where \( \omega_h = 1 - \alpha_h \).

**Step II. One-step-ahead forecast of volatility (\( t \in [0, N_o] \))**

\[
S_{t,h} = (3 + \frac{\alpha_h (6 - 5 \alpha_h)}{2 \omega_h^2} + \frac{\alpha_h^2}{\omega_h^2}) S^1_{t,h} \\
-(3 + \frac{\alpha_h (5 - 4 \alpha_h)}{\omega_h^2} + \frac{\alpha_h^2}{\omega_h^2}) S^2_{t,h} \\
+(1 + \frac{\alpha_h (4 - 3 \alpha_h)}{2 \omega_h^2} - \frac{\alpha_h^2}{2 \omega_h^2}) S^3_{t,h}
\]

**Step III. Updating (\( t \in [1, N_o] \))**

\[
S^1_{t+1,h} = (1 - \omega_h) S^1_{t+1,h} + \omega_h S^1_{t,h}
\]

\[
S^2_{t+1,h} = (1 - \omega_h) S^2_{t+1,h} + \omega_h S^2_{t,h}
\]

\[
S^3_{t+1,h} = (1 - \omega_h) S^3_{t+1,h} + \omega_h S^3_{t,h}
\]

The optimum \( \alpha_h \) that minimizes sum of squared errors in equation 3.2, will use to calculate \( \hat{\sigma}_{t,h} \).

### 3.7 Regularized Volatility Forecasts

In order to improve the results of the volatility forecasting, Elastic Net regularization is applied to the smoothed values generated by DES and TES methods. The idea of regularized forecasting
is proposed by [31] in which the explicit form of regularized volatility forecasting based on Elastic Net penalty is given as

$$S_{t,h}^{EN} = \frac{\text{sign}(S_{t,h} - s_h)(|S_{t,h} - s_h| - \omega_h \lambda_h)_+}{1 + (1 - \omega_h)\lambda_h} + s_h,$$

where the tuning parameters $\omega_h$ and $\lambda_h$ vary between 0 and 1. The optimum values for the tuning parameters are obtained by minimizing the sum of squares of forecasting errors as follow:

$$\omega_{h}^{opt}, \lambda_{h}^{opt} = \arg\min_{\omega, \lambda} \sum_{t=1}^{N_a} (\sigma_{t,h} - S_{t-1,h}^{EN}).$$

The regularized volatility forecasts can be computed by using the optimal tuning parameters. Note that, Elastic Net is used as regularization method. However, this includes the Lasso and Ridge regularization methods by selecting the tuning parameters in equation 3.3 properly. In more details, Lasso and Ridge regularization are obtained by setting $\lambda_h$ and $\omega_h$ to 1 in equation 3.3, respectively. The average mean absolute error of day-ahead volatility forecasts is computed using DES, TES, and their combination with Elastic Net regularization. Figure 3.4 presents the results of volatility forecasts of day-ahead electricity prices using double and triple exponential smoothing methods. There are two peaks in the MAE of volatility forecasts; one at 8:00 in the morning and the other one at 8:00 in the evening. In most of the other hours, MAE fluctuates around 11. Figure 3.5 demonstrates the impact of applying Elastic Net regularization on DES and TES volatility forecasts. It can be concluded that applying the Elastic Net regularization method does not reduce the maximum MAE, which occurs at 8:00 in the morning and evening. However, MAE at other hours is decreased to some extent. Table 3.2 indicates that while DES and TES perform similarly, Elastic Net improves their results by decreasing MAE of volatility forecasts to almost 11.
PFEP-HOEP

3.7 Regularized Volatility Forecasts

Figure 3.4: Unregularized volatility forecasts of electricity prices.

Table 3.2: MAE of Different Volatility Forecasts of Electricity Price

<table>
<thead>
<tr>
<th></th>
<th>Regularized DES</th>
<th>Regularized TES</th>
<th>DES</th>
<th>TES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Absolute Error</td>
<td>11.10</td>
<td>11.07</td>
<td>11.98</td>
<td>11.96</td>
</tr>
</tbody>
</table>
Figure 3.5: Regularized volatility forecasts of electricity prices.
Chapter 4

Day-Ahead Forecasts of Electricity Demand

As previously demonstrated in the figure 3.1, variations in electricity demand affect the electricity prices. Therefore, by computing day-ahead forecasts of electricity demand as an explanatory variable, multivariate forecasting models can be implemented for PEPF. Prediction of the load is a supervised learning process in which a regression method is applied to a set of features such as meteorological variables to predict the dependable variable. In this chapter, a predictive model for day-ahead electricity demand is proposed that consists of hourly weather data (including seasonal variation as well) and historical load data. The common forms of analysis for this purpose are regression based methods. In the proposed model, performance of different regression models such as support vector, least absolute shrinkage and selection operator (LASSO) and gradient boosting decision trees are evaluated for day-ahead load forecasting. In order to improve the forecast accuracy (smaller mean absolute error) of regression methods, a dual phase forecasting method is proposed. In the first phase, a neural network consists of convolution and recurrent neurons that take the historical load data and outputs electricity load forecasts. In the second phase, the results of the first phase forecasting are fed into the regression based methods to obtain
4.1 Load Forecasting Literature

Load forecasting is an important process for electricity markets. An accurate forecast for supply and demand helps the electricity market schedule suppliers to meet the whole demand for the next day. Further, electricity supply and demand have direct impact on electricity price. For instance, a shortage in production increases electricity prices, while overproduction may result in negative prices. An accurate forecasts of supply and demand can be used to control price fluctuations. Historical load data are the most useful data for load forecasting. However, they seem inadequate to capture all variation of electricity demand. Unexpected weather events can change electricity demand suddenly. Weather forecasts seem to have a strong influence on the accuracy of load forecasting. For instance, the weather temperature is a key determinant of electricity demand in Canadian cities that experience cold winters. Similarly, warm summers drive up the electricity demand for cooling purposes in Canadian cities. Other meteorological variables such as solar irradiance can also affect electricity demand. For example, photovoltaic power plants are growing in the electricity supply system and final customers can establish solar power generators to produce electricity. If there is a need for extra electricity, it will be bought from the grid. Solar energy seems to be a more sustainable energy resource than fossil fuel because there is no air pollution in solar power plants’ operation. However, solar energy has increased uncertainty in electricity demand because of its stochastic behavior.

Load forecasting can be divided into long and short-term categories. In the long-term analyses, electricity demand is predicted for upcoming years to determine if there is a need for more capacity. The short-term forecasting focuses on electricity demand for the next day or next hour to schedule the suppliers properly. In both categories, the demand forecasts highly depend on important determinants of electricity demand. Therefore, a fundamental step in the forecasting procedure is finding explanatory variables that can explain electricity demand variation ade-
For electricity demand forecasting, various classes of explanatory variables have been considered in the literature. For example, a comprehensive model proposed in [32] includes economic and non-economic variables to forecast annual residential electricity demand in 10 years. The model considered electricity price for residential consumers, income per capita, gas price, price of selected electric appliances, weather temperature, number of residential units, size of households, type of residential area (urban or rural), and time.

Weather temperature, humidity, solar radiation, and population were used to model peak electricity demand in Saudi Arabia in [33]. The results of [33] showed significant variations of maximum electricity demand in warm climate residential areas by accounting for the associated variables. Different models for peak electricity demand forecasting in hot and humid environments were compared in [34]. The comparison also demonstrated that per capita income is another important determinant of electricity consumption in long-term modeling. In addition, temperature and humidity were considered as predictors in some of the past studies [35], and [36]. It turned out that the variations in electricity consumption in the UK are better described by temperature rather than humidity. However, the relationship between temperature and electricity consumption can be affected by factors such as seasonal and geographical variables [37].

Non-linearity of temperature-demand relationship was investigated in [38] by classifying the days of the year into heating and cooling degree days. They defined a threshold for average daily temperature, which classified days of the year into cooling and heating days. It was shown in [39] that non-climatic variables such as electricity price have a substantial effect on the commercial sector’s electricity demand. However, those variables do not affect electricity demand in residential areas significantly in short-term modeling. In order to improve the accuracy of load forecasting, temperature, wind speed, humidity, cloud coverage, and the calendar effect considered in [40] as determinants of monthly electricity demand in Italy. They concluded that temperature is the most important variable for forecasting electricity demand. A combination of all meteorological variables and the calendar effect resulted in a lower mean absolute percentage error.
Seasonality is another determinant of electricity demand that has been addressed in the literature [41], [42] and [43]. Each day of the week impacts electricity demand in residential areas [42]. For instance, the difference between electricity demand during weekdays and weekends was reported in [43]. Splitting the historical load data into subsets according to days of the week can improve the accuracy of forecasts [41]. In [44], an important variable “Time of the day” was considered for hourly load forecasting.

As the literature review indicates, one of the key predictors of electricity demand in residential areas is temperature. However, due to the deployment of solar panels in residential areas, other weather variables such as cloud cover fraction and solar irradiation also play an important role in electricity demand. Thus, I consider a comprehensive model for short-term load forecast consisting of several meteorological variables described in table 4.1.

Table 4.1: Meteorological Variables Used in the Analysis

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Density (AD)</td>
<td>kilogram per cubic meter (kg/m(^3))</td>
<td>Mass per unit volume of Earth’s atmosphere</td>
</tr>
<tr>
<td>Cloud cover (CC)</td>
<td>fraction [0,1]</td>
<td>The portion of the sky that is covered in clouds</td>
</tr>
<tr>
<td>Irradiance Surface (IS)</td>
<td>Watt per square meter (W/m(^2))</td>
<td>The total amount of energy received on the Earth’s surface either directly from the sun</td>
</tr>
<tr>
<td>Irradiance Toa (IT)</td>
<td>Watt per square meter (W/m(^2))</td>
<td>Amount of received energy from sunlight per area at the top of the atmosphere</td>
</tr>
<tr>
<td>Precipitation (P)</td>
<td>millimeter per hour (mm/hour)</td>
<td>Amount of water in any form falling down to earth in an hour</td>
</tr>
<tr>
<td>Snowfall (SF)</td>
<td>millimeter per hour (mm/hour)</td>
<td>The amount of snow fall in mm during an hour</td>
</tr>
<tr>
<td>Snow Depth (SD)</td>
<td>centimeter (cm)</td>
<td>The total amount of snow on the surface of an area</td>
</tr>
<tr>
<td>Temperature (T)</td>
<td>degree Celsius (°C)</td>
<td>A measure that indicates how hot or cold the air is</td>
</tr>
</tbody>
</table>

Calendar and time effects are also added to the model to have more accurate load forecasting results. A proper forecasting method is required to capture the non-linear relationship between
electricity demand and other variables. Regression analysis-based methods are the common way of load forecasting [45]. It has been shown that Lasso regression can attain accurate forecasts when applied to temperature and calendar or time effect [45]. Although traditional methods perform well in load forecasting, machine learning methods such as neural networks and support vector regression have slightly better performance.

### 4.2 Electricity Demand Determinants

As mentioned earlier, electricity demand depends on several variables. Weather data have always been considered important determinants of future load in short-and long-term modeling. Figure 4.1 indicates the effect of air temperature on electricity load in 2016. It follows from figure 4.1 that there is a non-linear relationship between these two variables so that load is high for both colder and warmer weather conditions. The possible reason may be the use of electricity in Ontario for heating and cooling of both space and water. Therefore, it can be expected that electricity demand reaches its minimum when the temperature is neither cold nor hot.

The calendar effect, which considers the effect of weekends and holidays on electricity demand, was already proposed in [40]. Figure 4.2 represents the contour plot for electricity demand in January 2019. As presented in this figure, electricity demand during the weekdays is high and shown by bright colors. The color turns dark for weekends and indicates that electricity demand is lower on weekends.

As illustrated in figure 4.3, time of the day is also another important determinant of electricity demand. During breakfast and dinner, customers need more electricity for cooking, watching TV, and indoor lighting. It follows from the analysis that time and weather data are appropriate variables to explain electricity demand variation in the short term.

I used archival data on meteorological variables presented in table 4.1, and electricity demand in Ontario. The only province in Canada that has these data publicly available is Ontario. The electricity production and distribution in Ontario is managed by a regulatory company named
Independent Electricity System Operator (IESO). Hourly electricity demand in Ontario from 2002 until present is available from IESO. Also, hourly information for variables in table 4.1 for Ontario from 1985 until present are available online from the government of Canada website. Finally, the sample data consists of meteorological data and electricity demand in Ontario from 2010-2019. The relationship between variables in table 4.1 is also evaluated. In most cases, the correlation between variables is almost zero. However, there is a strong positive correlation between IS-IT (0.94) and SD-AD (0.83). There is a strong negative correlation between T-SD (-0.84) and T-AD (-0.98).

4.3 Method of Analysis

The proposed dual-phase model for day-ahead forecasts of electricity demand is described in figure 4.4. First, a neural network including Convolutional, recurrent and dense layers receives
the historical electricity demand. Here, the input of the neural network can come from the past
168 hours of data and the output of the neural network is the forecast of electricity demand for
the next 24 hours. In the second phase, a regression model receives the meteorological variables,
time/date effect, and historical load data as well as day-ahead forecasts of electricity demand
from the neural network model. Finally, the regression model generates day-ahead forecasts of
electricity demand.

The proposed model for day-ahead load forecasting can be expressed as follows

$$\hat{y}_t = f(W_t, TC, Y),$$  \hspace{1cm} (4.1)

where $W_t$ is an $8 \times 1$ vector that includes forecasts of meteorological data for time $t$ and is defined
as

$$W_t = [P_t, T_t, IS_t, IT_t, SF_t, SD_t, CC_t, AD_t].$$  \hspace{1cm} (4.2)
Figure 4.3: Impact of time and date on electricity demand.

Figure 4.4: Dual-phase model for day-ahead load forecasting.

TC is a $4 \times 1$ vector that indicates time and calendar effect on electricity demand and is defined as

$$TC = [ToD, DoW, WoW, NH].$$

The elements of this vector refer to the characteristics of the time and the day that electricity
demand needs to be forecasted. ToD is an integer variable that can vary between 1 and 24 and indicates the hour of the day. DoW shows the day of the week by an integer number between 1 and 7, where 1 refers to Sunday, 2 to Monday, and so forth. Furthermore, WoW is a binary variable that shows if the day is a weekend or weekday. Also, NH is another binary variable that determines if the day is a national holiday or not. $Y$ is a $5 \times 1$ vector that contains information about electricity demand at specific times

$$Y = [Y_{PH}, Y_{AL}, Y_{PD}, Y_{PW}, Y_{DA}].$$  \hspace{1cm} (4.4)

$Y_{PH}$ is electricity demand for the previous hour, and $Y_{AL}$ is the average load in the last 24 hours. $Y_{PD}$ and $Y_{PW}$ are electricity demand for the previous day and week (at a specific time), respectively. The last element of the vector, $Y_{DA}$, is the day-ahead load forecasts obtained by the neural network.

### 4.4 ANN model for Day-Ahead Load Forecasting

The first phase of the load forecasting model is a neural network that takes historical electricity demand and computes day-ahead forecasts of electricity demand. The first step in time series forecasting with ANN models is preprocessing the input data. Neural networks are sensitive to the scale of input data because weights and biases of convolution and recurrent neurons are small numbers between 0 and 1. The scaling process is a simple min-max transformation that maps each element of electricity demand data, $d_i$, to $d_i^s$ as

$$d_i^s = \frac{d_i - d_{\text{min}}}{d_{\text{max}} - d_{\text{min}}},$$  \hspace{1cm} (4.5)

where $d_{\text{max}}$ and $d_{\text{min}}$ are the maximum and minimum values of the electricity demand data, respectively. Table 4.2 provides some descriptive statistics of the Ontario electricity demand data. According to the table, the total number of samples in the data set is 87648, with a mean
of 15791.85 Mega Watt (MW) and a standard deviation of 2397.12. The first and third quartiles of data are also presented by $Q_1$ and $Q_3$. It can be concluded from the descriptive statistics that the Ontario electricity demand data spread around their mean value almost evenly. A histogram of the samples can clarify the distribution of the data. Figure 4.5 indicates the histogram of electricity demand data. It follows from the histogram that the whole data spread around some median value, and there is no concentration of data around outliers. If there exist outliers, they need to be removed because they deteriorate any attempt for scaling the data. It should be mentioned that the 2019 electricity demand data is used for testing the performance of the ANN forecasting model. Therefore, a copy of 2019 data remains untouched for testing the model.

<table>
<thead>
<tr>
<th>Table 4.2: Descriptive Statistics of Electricity Demand Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>87648</td>
</tr>
</tbody>
</table>

Figure 4.5: Distribution of 2010-2019 Ontario electricity demand.
against unseen data. The model seeks to use electricity demand data from the past 168 hours and forecasts the next 24 hours. Figure 4.6 demonstrates a sense of the data structure. The blue part of the graph is a sample for training the ANN model and contains load data of 168 hours. The last part of the graph plotted in red represents the day-ahead forecasts of electricity demand to be computed by the ANN model. Therefore, the ANN model receives input data of shape $(168 \times 1)$ and generates day-ahead forecasts with the shape of $(24 \times 1)$. Accordingly, the training input (2010-2018 Ontario electricity demand data), which is a $(78888 \times 1)$ vector, is restructured to a tensor with a shape of $(3280, 168, 1)$. Moreover, the training output is another tensor with a shape of $(3280, 24, 1)$.

![Lagged Data Structure](image)

**Figure 4.6: Data structure for training and testing the ANN model.**

Recurrent neural networks can capture the temporal patterns in the data and are commonly used in time series forecasting. However, a recurrent neural network easily gets confused when loaded with high dimension input data such as a $(168 \times 1)$ vector. As demonstrated in figure 4.4, a convolution layer comes before the recurrent layer to decrease the dimension of input data. In
more detail, the convolution layer extracts the important features from the \((168 \times 1)\) input vector, produces a \((7 \times 1)\) vector, and passes it to the recurrent layer. The building block of the recurrent layer can be any of Simple RNN, LSTM, or GRU neurons. After the recurrent layer, a drop-out layer masks 10 percent of the recurrent layer neurons in each training iteration. Adding the drop-out layer to the model protects it from overfitting. Eventually, a dense layer generates day-ahead forecasts of electricity demand. The parameters of the ANN model are set as provided in Table 4.3. In addition, Table 4.4 presents the performance of ANN load forecasting model for each of Simple RNN, LSTM and GRU as recurrent units. In the training step, the weights and biases of neurons in the ANN model are tuned based on 2015-2018 Ontario electricity demand data. The 2019 data are considered as testing data to see how the model performs with unseen data. One technique that improves the training step is to take out 20 percent of the training set (2015-2018 electricity demand data) as validation data. The performance of the training algorithm is evaluated in each iteration using the validation data. Overfitting is the situation in which the validation loss is greater than the training loss. Overfitting occurs commonly in deep neural networks because they have many parameters and can fit the training data perfectly but perform poorly for unseen data. Masking a portion of neurons in each iteration of the training algorithm is a solution for overfitting, although it may cause underfitting. In more detail, underfitting appears when the

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution Layer Filters</td>
<td>256</td>
</tr>
<tr>
<td>Convolution Layer Kernel Size</td>
<td>24</td>
</tr>
<tr>
<td>Numbers of Recurrent Units</td>
<td>256</td>
</tr>
<tr>
<td>Dropout Rate</td>
<td>0.10</td>
</tr>
<tr>
<td>Number of Dense Layer Units</td>
<td>24</td>
</tr>
<tr>
<td>Number of Lagged Observation</td>
<td>168</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Batch Size</td>
<td>80</td>
</tr>
<tr>
<td>Epoch</td>
<td>200</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Loss Function</td>
<td>Mean Squared Error</td>
</tr>
</tbody>
</table>
number of epochs is small, resulting in an early stopping of the training algorithm. Figure 4.7 demonstrates that the ANN model is appropriately trained so that training and validation loss are converged to a constant value. The figure also implies that the training algorithm takes bigger steps at the early iterations and uses small steps as it reaches the optimal point. Day-ahead forecasts of the electricity demand of Ontario for a part of the testing data is indicated in figure 4.8. The solid blue line represents the actual electricity demand, and other lines result from the ANN load forecasting models. All of the forecasting results follow the trend in actual observation and it can be concluded from figure 4.8 that all three ANN models perform well for load forecasting during the night time. However, there are some deviations between the actual and forecast load for the daytime. Figure 4.9 compares the accuracy of load forecasting of ANN models, consisting of convolutional, recurrent, and dense layers. Three different recurrent units, namely GRU, LSTM, and Simple RNN, are used in the model. Figure 4.9 indicates the average MAE of load forecasts for 2019 Ontario electricity demand data in an hourly resolution. As mentioned, from 12 am until 6 am that electricity demand is low, and the forecast errors of ANN models are small. MAE of models keeps rising until 4 pm and then bend downward. Figure 4.9
indicates that using GRU as recurrent neurons in the ANN layer results in the least MAE in each hour. Table 4.4 also illustrates that overall, the ConvGRU model has the least MAE and RMSE as well as the highest value for $R^2$. Hence, in the second phase of the load forecasting model, the ConvGRU model results, along with other explanatory variables, will apply to regression methods.

Table 4.4: Comparison of the Three Different Recurrent Layers in the ANN Load Forecasting Model

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE</th>
<th>RMSE</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvRNN</td>
<td>496.37</td>
<td>666.24</td>
<td>0.91</td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>496.83</td>
<td>660.59</td>
<td>0.91</td>
</tr>
<tr>
<td>ConvGRU</td>
<td>457.51</td>
<td>632.65</td>
<td>0.92</td>
</tr>
</tbody>
</table>
As mentioned, the second phase of the load forecasting model uses weather data as explanatory variables for electricity demand. The weather data can be downloaded in hourly resolution from Canada’s government website (climate.weather.gc.ca). Moreover, regression methods take time/date variables such as an hour of the day, day of the week, and day of the year. The weather and electricity demand data from 2015 until 2018 are considered the training data set and the performance of the model is tested on 2019 data. Figure 4.10 demonstrates the performance of the second phase without receiving day-ahead forecasts of load from the ANN model. In more detail, figure 4.10 refers to the situation that regression methods that use historical load, weather, and time/calendar data to predict electricity demand. It follows from the analysis that GBR and XGB perform almost the same and provide much more accurate forecasts than other regression methods. Table 4.5 provides a more comprehensive comparison of regression methods.
by computing MAE, RMSE, and $R^2$ values. As figure 4.10 implies, GBR and XGB are better alternatives than SVR and Lasso regression to be used in the load forecasting model. However, further improvement will be achieved if ANN day-ahead load forecasts are added to GBR and XGB explanatory variables.

Figure 4.10: Hourly MAE of regression methods for load forecasting.

<table>
<thead>
<tr>
<th>Regression Method</th>
<th>MAE</th>
<th>RMSE</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lasso</td>
<td>627.14</td>
<td>811.72</td>
<td>0.81</td>
</tr>
<tr>
<td>Support Vector</td>
<td>595.77</td>
<td>753.89</td>
<td>0.87</td>
</tr>
<tr>
<td>Gradient Boosting</td>
<td>416.27</td>
<td>570.90</td>
<td>0.933</td>
</tr>
<tr>
<td>Extreme Gradient Boosting</td>
<td>415.99</td>
<td>570.46</td>
<td>0.932</td>
</tr>
</tbody>
</table>

Figure 4.11 presents hourly MAE of dual-phase load forecasting models that use different regression methods. As expected, GBR and XGB outperform the other two methods by providing load forecasts with smaller MAEs in each hour. Further, the dual-stage model successfully
manages to employ ANNs and regression methods to provide more accurate forecasts of electricity demand. For instance, the ANN-based models such as ConvGRU and ConvLSTM have better performance than regression methods in forecasting the electricity demand from midnight until 6 am. In contrast, regression methods are more accurate than ANN models from 6:00 am until midnight. The dual-stage model takes advantage of historical electricity demand data and weather and date/time data. As a result, day-ahead forecasts of electricity demand from the dual-phase model are more accurate than each phase. The MAE and RMSE of the dual-phase load forecasting models are provided in table 4.6. Overall, the combination of the ConvGRU model and XGB regression provides the most accurate load forecasts among other studied methods in this chapter. Recall from table 4.2, the mean of electricity demand data is 15791.85 MW. The MAE and RMSE of ConvGRU-XGB are 2.15 and 2.94 percent of the mean value, respectively. The small values of MAE and RMSE and the high value of $R^2$ indicate that the ConvGRU-XGB model can properly compute one-day-ahead forecasts of electricity demand.

Figure 4.11: Hourly MAE of dual phase load forecasting model.
Table 4.6: Evaluation of XGB Prediction of Electricity Demand in Ontario.

<table>
<thead>
<tr>
<th>Dual-Phase Model</th>
<th>MAE</th>
<th>RMSE</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvGRU-Lasso</td>
<td>451.97</td>
<td>621.84</td>
<td>0.921</td>
</tr>
<tr>
<td>ConvGRU-SVR</td>
<td>451.11</td>
<td>579.24</td>
<td>0.928</td>
</tr>
<tr>
<td>ConvGRU-GBR</td>
<td>343.58</td>
<td>476.81</td>
<td>0.954</td>
</tr>
<tr>
<td>ConvGRU-XGB</td>
<td>339.16</td>
<td>471.35</td>
<td>0.955</td>
</tr>
</tbody>
</table>
Chapter 5

Probabilistic Forecasts of Electricity Prices

This chapter seeks to compute probabilistic forecasts of day-ahead electricity prices of Ontario. The fundamental step in probabilistic forecasting is to derive point forecasts of the HOEP data. Chapter 4 describes ConvRNN model and regression methods as two different approaches for computing point forecasts of electricity demand. Similarly, day-ahead forecasts of electricity prices are computed in this chapter by implementing ANNs and regression methods. In addition, the triple exponential smoothing method is used to provide another set of day-ahead point forecasts of electricity prices. Later, linear and non-linear regression methods will be applied to the pool of point forecasts to generate quantile forecasts for electricity prices.

5.1 Autocorrelation of Electricity Prices

The ANN model for electricity price forecasting aims to use historical electricity prices to compute one-day-ahead forecasts of electricity prices. Before implementing the ANN model, the autocorrelation of electricity prices needs to be evaluated. Autocorrelation shows the correlation between a given time series and a lagged copy of itself over successive time intervals. In other words,
autocorrelation measures the relationship between a variable’s current value and its historical values. The autocorrelation of electricity prices of Ontario for the training data set is presented in figure 5.1. The vertical axis is the autocorrelation coefficient, $R_l$, which is defined as

$$R_l = \frac{C_l}{C_0}$$

where $C_l$ is the autocovariance function

$$C_l = \frac{1}{n_s} \sum_{i=1}^{n_s-l} (y_i - \bar{y})(y_{i+l} - \bar{y}).$$

Figure 5.1: Autocorrelation function of electricity prices.
As mentioned before, \( n_s \) is the number of observations in the training data, and \( \bar{y} \) denotes the mean of observations. Also, \( C_0 \) is the variance of electricity prices calculated as

\[
C_0 = \frac{1}{n_s} \sum_{i=1}^{n_s} (y_i - \bar{y})^2.
\]

If the autocorrelation is zero, there is no pattern in the time series, and the values are generated randomly. On the other hand, when there is a pattern in the series, then one or more of the autocorrelations will be significantly non-zero. Computing correlation coefficient to evaluate a relationship between two variables tells us half of the story because the correlation represents the degree of relationship in one sample. If another sample is selected, the correlation coefficient may be a different value. The goal is to determine whether the conclusion obtained from the sample can be generalized to the population. To that end, a statistical significance test needs to be conducted. The significance test determines whether the conclusion from the sample is expected to be true for the population and can be conducted through a hypothesis test. Hypothesis testing is the main part of statistical inference, which is concerned with making inferences about a population based on the analysis on a sample.

Assume that there is a sample of electricity prices of size \( n_s \). The sample autocorrelation for lag \( l \) is \( R_l \), and the population autocorrelation coefficient for the same lag value is \( P_l \). The goal is to make an inference about \( P_l \) based on \( R_l \). The hypothesis test reveals that whether \( P_l \) is almost zero or significantly different from zero. Suppose the test shows that \( P_l \) is almost zero. In that case, there is insufficient statistical evidence that the autocorrelation of the population is significant. In other words, the autocorrelation happened on account of random coincidences, and it does not appear in the entire population.

The hypothesis test starts by defining null and alternative hypotheses. Here, the alternative hypothesis is that the population has significant autocorrelation. Further, the null hypothesis indicates that there is no significant autocorrelation in the population. The student’s T-test (also called T-test) can be used to determine whether the autocorrelation observed in the sample can
be generalized to the population. The value of the T-test is calculated as

\[ T = \frac{R_l \times \sqrt{n_s - 2}}{\sqrt{1 - R_l^2}}. \]

If the t-value is high enough, the autocorrelation will be repeatable for the population as well. P-value is used to determine if the t-value is high enough. The p-value can be interpreted as the probability of obtaining test results (or even more extreme) from the population if the null hypothesis is correct. For example, a p-value of five percent means that there is only five percent probability that the autocorrelation of the sample occurred out of chance. When the p-value is small enough, one can reject the null hypothesis confidently. In most cases, a threshold of 0.05 or lower specifies the significance level. Using t-value and degrees of freedom which equals \(n_s - 2\), the p-value can be obtained by using a t-value table or programming software such as Python. The autocorrelation plot in figure 5.1 has a thin blue-colored area around the x-axis that demonstrates the confidence intervals. By default, this is set to a 95% confidence interval, suggesting that the correlation values outside of this area are very likely a correlation and not a statistical anomaly. Therefore, it follows from figure 5.1 that there is a significant correlation in the hourly Ontario electricity prices data. Recurrent neural networks can extract the pattern in the time series. Hence, the next section aims to use historical HOEP data and recurrent neural networks to provide day-ahead forecasts of electricity prices.

## 5.2 Pre-Processing of HOEP Data

As mentioned earlier, there are serial dependencies in the HOEP data and can be captured by recurrent neural networks. The goal is to train ANN models based on 2010-2018 HOEP data and test the models on 2019 data. Figure 5.2 demonstrates the training and testing HOEP data. Since recurrent neural networks use weights and biases between 0 and 1, the input data must be scaled first. It follows from figure 5.2 that there are some spike prices in the data. Spike prices...
Figure 5.2: Training and testing HOEP data for ANN models.

are defined as prices higher than $Q_3 + 1.7 \times IQR$ or lower than $Q_1 - 1.7 \times IQR$. Data analysis shows that almost 2.5 percent of the whole HOEP data are spike prices with the maximum and minimum of 1822.95 and -138.79 CAD/MWh. The negative price implies that electricity supply exceeds the demand, and eventually, suppliers have to pay their customers for selling the electricity. The majority of electricity prices vary between 10.00-33.00 CAD/MWh. The outliers are unfavorable and almost cancel out the effect of scaling. In order to solve this problem, the negative and positive spikes are set to $Q_1 - 1.7 \times IQR$ and $Q_3 + 1.7 \times IQR$, respectively.

Figure 5.3 illustrates the histogram of the HOEP data before and after removing the outliers. Figure 5.3 describes in more detail why outliers deteriorate any attempt to scale the data. Before removing the outliers, electricity prices are mainly concentrated around some median values. However, the histogram indicates that the data are much more spread out after removing the outliers. The right-hand side histogram of figure 5.3 demonstrates that replacing positive and negative spike prices with upper and lower thresholds makes the distribution of electricity prices...
Figure 5.3: Histograms of original and filtered HOEP data.

much wider than it used to be. Now the HOEP data are ready to be scaled. The scaled HOEP time series, \( p_t^s \), is obtained through a relatively simple transformation

\[
p_t^s = \frac{p_t - p_{\text{min}}}{p_{\text{max}} - p_{\text{min}}},
\]

in which \( p_t \) is the original HOEP data with minimum and maximum values \( p_{\text{min}} \) and \( p_{\text{max}} \), respectively.

Another essential task in preparing data is to change the structure of the data set. The original training data is a 78888 \( \times \) 1 column vector, but the convolutional and recurrent layers expect 3-dimensional input data such as (Samples, Time Steps, Features). Here, each sample refers to electricity prices data for 168 hours. Time steps refer to the number of historical electricity prices that the model uses to forecast the next 24 hourly prices. Moreover, ”Features” denotes the number of feature data that the model uses in order to compute day-ahead forecasts of HOEPs. As the model just uses the historical HOEPs, the number of features will be equal to one. For instance, using the last seven days (168 hours) of prices to forecast future 24 hours
would require the training input and output data to have (3280, 168, 1) and (3280, 24) dimensions, respectively.

## 5.3 ConvGRU HOEP Forecasting Model

The ConvGRU HOEP forecasting model consists of different layers. First, a convolution layer receives sample data from the input layer and extracts the important features from the data. The next layer uses one of the recurrent units such as simple RNN, GRU, or LSTM. As a matter of fact, the performance of recurrent neurons gets worse as the dimension of their input data increases. The convolutional layer can extract the main features of data and generate a new series with much less dimension. As a result, the neurons in the recurrent layer (come after convolution layer) receive low dimension data. The recurrent layer’s output goes to a fully connected dense layer to generate the day-ahead forecasts of electricity prices. The training data set is HOEPs from 2010 to 2018, 20 percent of which are considered as validation data. The validation data set is a part of training data that the model uses to check whether the training algorithm tunes the hyperparameters correctly in each iteration. Validation data can decrease the chance of having over or underfitting in the training process.

In order to avoid overfitting, a dropout layer is added after the recurrent layer to mask 10 percent of neurons in each training. The performance of the model is tested on an out-of-sample 2019 HOEP data. The model is implemented using Keras library and Tensorflow backend in Python. The parameters of the model are presented in table 5.1.

Figure 5.4 provides some features of the training process of the ConvGRU electricity price forecasting model. The convergence of training and validation loss curves indicates that the training proceeds successfully without having any problem with overfitting or underfitting. The training algorithm also decreases the learning rate after almost 50 epochs to ensure that training loss is not going down any further. Figure 5.5 illustrates the performance of the ANN price forecasting models against unseen data. In fact, all the 2019 HOEP data applied to the ANN
Table 5.1: Initial Values ANN Forecasting Model

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution Layer Filters</td>
<td>256</td>
</tr>
<tr>
<td>Convolution Layer Kernel Size</td>
<td>24</td>
</tr>
<tr>
<td>Numbers of Recurrent Units</td>
<td>256</td>
</tr>
<tr>
<td>Dropout Rate</td>
<td>0.10</td>
</tr>
<tr>
<td>Number of Dense Layer Units</td>
<td>24</td>
</tr>
<tr>
<td>Number of Lagged Observation</td>
<td>168</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Batch Size</td>
<td>80</td>
</tr>
<tr>
<td>Epoch</td>
<td>200</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Loss Function</td>
<td>Mean Squared Error</td>
</tr>
</tbody>
</table>

Figure 5.4: Loss and learning rate of training algorithm for the ANN price forecasting model.

models as testing data, but figure 5.5 plots the forecasting results just for one arbitrary week in May 2019. As mentioned, the ANN models combine convolutional and recurrent layers of neurons to compute day-ahead forecasts of HOEPs. It follows from figure 5.5 that the ANN forecasts can follow the upward and downward trends in the actual HOEP data in most cases. There are some sudden drops and rises in HOEP data that ANN forecasts are not able to follow. Some changes in electricity price data are unpredictable because they depend on human behavior. Figure 5.6
demonstrates how accurate the ANN models forecast electricity prices in each hour. MAE varies from hour to hour, and most of the time ConvGRU model provides the most accurate (the least MAE) forecasts of HOEP. Moreover, there are two peaks in the MAE graph that indicate electricity prices are more volatile at 8:00 in the mornings and 7:00 in the evenings.

5.4 Benchmark HOEP Forecasting Model

In order to determine how well the ANN models perform in computing day-ahead forecasts of electricity prices, benchmark models are used for comparison. The ANN models can be very complex and take a considerable amount of time to train the neurons and generate the results. However, ANNs are not necessarily very efficient in time series forecasting and can beat simple benchmark models. Benchmarks are simple models that perform forecasting based on the best guess. Four benchmark models are defined that consider the forecast of electricity prices for each
The hour of the next day is as follows:

- **B1**: price of the same hour of the previous day,
- **B2**: price of the same hour and day of previous week,
- **B3**: average price of the same hour and day during last 2 weeks
- **B4**: average price of the same hour and day during last 4 weeks

Table 5.2 presents a comparison of the performance of ANN-based forecasting models against benchmarks. It reveals that among the benchmarks, B4 has the most accurate forecasts. All of the ANN-based models significantly outperform B4, and the ConvGRU model results in the most accurate predictions.
5.5 Regression Based Forecasting Model

Up to now, day-ahead forecasts of electricity prices are computed by a ConvGUR model that uses electricity prices from 168 hours to forecast the following 24 prices. However, electricity price forecasting can take advantage of other features such as electricity load and price volatility. In this section, GBR and XGB regression methods are assigned to predict HOEPs using a set of explanatory variables from 2015 until 2018 that includes

- Historical HOEP for the last 168 hours,
- Day-ahead forecasts of electricity price volatility, and load,
- Time/date related variables such as time of the day, day of the week, and day of the year.

At first glance, regression methods receive many input data because of 168 hours of historical HOEP data. However, the gradient boosting regression methods have the unique feature that they perform better when receiving high-dimensional input data. Therefore, there are no concerns that GBR and XGB regression methods get confused when receiving many input data. The regressors are trained using the 2015-2018 features data set and are tuned in a trial and error process. Data analysis indicates that the XGB model outperforms GBR by decreasing the ConvGRU model’s MAE from 8.20 to 7.06. It should be mentioned that using the day-ahead volatility forecasts as a feature improves the XGB performance by 15.79 percent. In other words, without day-ahead volatility forecasts, the XGB regressor results in an MAE of 7.24.
5.6 Triple Exponential Smoothing

So far, two sets of day-ahead HEOP forecasts are obtained from GBR and ConvGRU model. The third method considers the same idea that is used for volatility forecasting. In more detail, the 2010-2019 HOEP data are divided into 24 subsets so that each subset contains electricity prices for a particular hour. Then, the triple exponential smoothing method described in section 3.6 computes day-ahead forecasts of 2019 HOEP data. Performances of the three forecasts are compared in figure 5.7 in terms of hourly MAE.

![Hourly MAE of Point forecasting Methods](image)

Figure 5.7: MAE of 3 Different Day-ahead Point Forecasting Models.

5.7 Quantile Forecasts of Electricity Prices

In order to obtain probabilistic forecasts of electricity prices, quantiles are used to describe the distribution of HOEP data. The best-known quantile is the median which is the 50% quantile. The standard ordinary least squares method looks for the relationship between independent vari-
variables and the conditional mean of the dependent variable. On the other hand, quantile regression methods find the relationship between dependent variables $z_t$ and quantiles of dependent variable $p_t$. Compared to the point forecasting methods, quantile forecasts provide a more comprehensive explanation of the future values of the dependent variable.

The most common way to generate probabilistic forecasts of electricity prices is to compute the PIs of point forecasts of electricity prices. The point forecast for electricity price at time $t$, i.e., $\hat{p}_t$, and the forecasting error

$$\epsilon_t = p_t - \hat{p}_t,$$

are the two main elements for computing PIs. The center of PIs is set to $\hat{p}_t$ and cumulative distribution function (CDF) of $\epsilon_t$, $F_{\epsilon_t}$, is used to derive lower and upper bounds of PIs. For instance, 5% and 95% quantiles of $\epsilon_t$ are required to compute 90% PI of electricity price. The idea of computing PIs can be extended further to calculate 99 quantiles ($q = 1\%, \ldots, q = 99\%$) as

$$F_{\epsilon_t}^{-1}(q) = \hat{p}_t + F_{\epsilon_t}^{-1}(q),$$

and obtain a discretization of the electricity price distribution. Different methods have been proposed to calculate the distribution of the error term. QRA is a method to derive quantiles of the electricity price distribution. Initially, the QRA method considers point forecasts of electricity prices as explanatory variables for the quantiles of the electricity price distribution and the number of point forecasts can be chosen arbitrarily. During the GEFCom2014, some of the exogenous variables such as hourly electricity load/price forecasts are added to the explanatory variables, which eventually improved the QRA method’s results.

Assume that the relationship between electricity price at time $t$ and its corresponding point forecasts is defined as

$$p_t = f(z_t) + \epsilon_t$$

where $z_t$ is a vector of explanatory variables including point forecasts, and $f(\cdot)$ is a transformation
function [20]. The ordinary least square method minimizes $\sum_i \varepsilon_i^2$ with the sum of squares of the forecasting errors. In addition, least absolute-deviation regression minimizes $\sum_i |\varepsilon_i|$ in order to compute the median. Eventually, the quantile forecasts are obtained by minimizing the following loss function

$$
L = \sum_{i \in I_U} q |\varepsilon_i| + \sum_{i \in I_O} (1 - q) |\varepsilon_i|,
$$

where $I_U$ and $I_O$ are under and over predictions, respectively. The summation gives the asymmetric penalties $q|\varepsilon_i|$ for under predictions and $(1 - q)|\varepsilon_i|$ for over predictions. In more detail, the goal of quantile regression is to find the optimum $f(.)$ that minimizes the loss function

$$
\mathcal{L}(p_t, q) = \sum_{t: p_t \geq f(z_t)} q |p_t - f(z_t)| + \sum_{t: p_t \leq f(z_t)} (1 - q) |p_t - f(z_t)|
$$

which can also be written as

$$
\mathcal{L}(p_t, q) = \sum_{t=1}^{n_s} [1_{p_t \geq f(z_t)} q |p_t - f(z_t)| + 1_{p_t \leq f(z_t)} (1 - q) |p_t - f(z_t)|]
$$

where $\mathbf{1}$ is an indicator function. Eventually, the loss function can be written in a simpler form as

$$
\mathcal{L}(p_t, q) = \sum_{t=1}^{n_s} \rho_q(p_t - f(z_t)),
$$

where $\rho_q$ is called the check function and is defined as

$$
\rho_q(u) = (q - 1_{u<0})u.
$$

In order to clarify the topic in more detail, suppose that a prediction has been made for a single point with an actual value of zero. The data has already been scaled, and the predictions vary between $-1$ and $+1$. As a result, the errors also range from $-1$ to $+1$. Figure 5.8 illustrates how the check function varies with the error and three different quantiles. The blue lines indicate the
median regression, which eventually provides the 50% quantile. In order to find the median, the check function is symmetric around zero, which means that the function weighs the underestimates equally to the overestimates. It follows from equation 5.1 that the quantile regression loss function treats the positive and negative errors in the same way for $q = 0.5$. That is, the underestimates and overestimates are penalized by a 0.5 coefficient which makes a symmetric shape for the check function.

The red line represents the check function for the 10% quantile. The graph has an asymmetric shape because the penalties that are assigned by the check function are lower for negative errors and higher for positive errors. The 10% quantile implies a 10 percent chance that the actual observation is below the predicted value. Therefore, it makes sense to assign less penalty to underestimates than to overestimates. Additionally, the green line shows the 90% quantile, which is the reverse pattern from the 10% quantile.

The quantile regression method models $f(.)$ as a linear function and the coefficients of the
linear function are obtained by the linear programming optimization method. However, machine learning methods can be applied to model \( f(.) \) as a non-linear function. Each machine learning method has a training algorithm that seeks to minimize a loss function. When the loss function is set to mean squared errors, the machine learning method provides the expected future value of a dependent variable such as electricity price or demand. Additionally, if the training algorithm minimizes the check function, machine learning methods generate quantile forecasts of the dependent variables. I used QRA and GBR methods and a fully connected NN model to minimize the loss function and to eventually compute the quantile forecasts of day-ahead electricity prices. Figure 5.9 demonstrates the total quantile loss for each of the three methods. It follows from the figure that when \( q \leq 0.5 \), QRA minimizes the loss function better than the other two methods. However, as \( q \) increases, the GBR method demonstrates better performance than the other methods.

Figure 5.9: Total loss per quantile of predictive models.
5.8 Verification of Probabilistic Forecasts

One of the criteria used to compare the results of the three models in quantile forecasts is reliability. Reliability determines how the quantile forecasts \( \hat{q}_t^{(\alpha)} \) at the \( \alpha \) nominal level match with empirical findings. For instance, when \( \alpha = 0.9 \), 90 percent of the HOEP data are expected to be less than \( \hat{q}_t^{(0.9)} \). The empirical level of the quantile forecasts is given by

\[
a_{\alpha} = \frac{\sum_{i=1}^{n_t} 1(p_i < \hat{q}_i^{(\alpha)})}{n_t}.
\]

A set of empirical levels of quantile forecasts can be obtained for different values of \( \alpha \) in \([0, 1]\) and the reliability assessment can be summarized in reliability diagrams. Figure 5.10 is a reliability diagram in which empirical levels of quantile forecasts are reported versus their nominal levels for QRA, GBR, and NN methods. Red squares show the ideal situation. The closest line to the red squares indicates the most reliable quantile forecasts. A metric is required to determine which line has the least deviation from the diagonal line. I define Reliability Deviation as the distance between the empirical quantiles from the diagonal line. Reliability deviation, \( \mathcal{E} \), can be computed as

\[
\mathcal{E} = \sqrt{\sum_{\alpha=0.01}^{0.99} (a_{\alpha} - \alpha)^2}.
\]

Among various quantile forecasters, the most accurate one has the least reliability deviation.

Prediction interval estimates represent an interval in which a future observation will lie with a specified probability, given all the past observations. For example, a 95% prediction interval of 10 to 30 dollars per megawatt-hour for the electricity prices reveals that future electricity prices will fall into that range 95% of the time. There is a 5% chance that electricity prices will lie out of the interval. As quantile forecasts of electricity prices have already been computed, deriving prediction intervals is an easy task. For instance, the 90% interval is the area above and below of the 5% and 95% quantiles, respectively. In order to determine how tight the prediction intervals
are, I used the sharpness metric. The sharpness of interval forecasts is obtained by calculating their average over the evaluation period

\[ s_\delta = \frac{\sum_{t=1}^{n_t} \hat{q}_t^{(\alpha_u)} - \hat{q}_t^{(\alpha_l)}}{n_t}, \]

where \( \alpha_u \) and \( \alpha_l \) are the upper and lower bounds, and \( \delta \) is the difference between them. The smallest sharpness value indicates that the corresponding quantile forecasts method has the tightest PIs.

In order to evaluate the overall performance for a probabilistic forecast model, I compute the most common skill score for predictive densities known as continuous ranked probability score (CRPS). The CRPS generalizes MAE to the case of probabilistic forecasts. For a given predictive
density of electricity prices, $\hat{F}_{pt}(x)$ and corresponding observation $p_t$

$$\text{CPRS}(\hat{F}_{pt}(x), p_t) = \int_{-\infty}^{\infty} (\hat{F}_{pt}(x) - 1_{(p_t \leq x)})^2 dx.$$  \hspace{1cm} (5.2)

Figure 5.11 illustrates the overall assessment of probabilistic forecasts. The dotted line is the discretization forecasts of electricity prices’ CDF for a particular hour. The CDF is obtained by computing 99 quantiles. The dashed line indicates the actual observation, which is equal to 0.45. The horizontal axis shows the scaled electricity prices range between 0 and 1. The blue-colored area indicates the deviation of the obtained probabilistic forecast from the perfect forecasts. The smaller the area of the colored region, the more accurate the probabilistic forecasts. The CRPS metric determines how close is the dotted line to the dashed line. However, the integral in equation 5.2 encompasses numerical difficulties. A more convenient way to compute CPRS as

![Figure 5.11: Overall skill assessment of probabilistic forecasts of electricity prices.](image)
The loss function so that
\[
\text{CPRS}(\hat{F}_p(x), p_i) = \int_0^1 L(p_i, q) dq.
\]

Finally, the CRPS score value is given by taking the average for each of the predictive densities in the testing data set as
\[
\text{CPRS} = \frac{1}{n_t} \sum_{i=1}^{n_t} \text{CPRS}(\hat{F}_p(x), p_i).
\]

Table 5.3 expresses the results of probabilistic forecasts evaluation. The pool of point forecasts of day-ahead hourly Ontario electricity prices is used to compute quantile forecasts. Linear regression analysis methods such as QRA have been among the most common ways of computing quantile forecasts of electricity prices. I use non-linear regression methods such as GBR and a fully connected neural network for probabilistic forecasting of electricity prices. Table 5.3 compares performances of the 3 methods. It follows from the data analysis that the GBR method has the least score in all metrics. That is, GBR quantile forecasts cover more observations than other methods while it has the tightest PIs. In addition, the CPRS for GBR indicates that GBR quantile forecasts are more accurate than the other two methods.

<table>
<thead>
<tr>
<th>Model</th>
<th>Reliability Deviation</th>
<th>Sharpness</th>
<th>CPRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantile Regression Averaging</td>
<td>0.09</td>
<td>11.30</td>
<td>1226.4</td>
</tr>
<tr>
<td>Neural Network</td>
<td>0.23</td>
<td>11.17</td>
<td>1270.2</td>
</tr>
<tr>
<td>Gradient Boosting Regression</td>
<td>0.04</td>
<td>11.12</td>
<td>1182.6</td>
</tr>
</tbody>
</table>

The results of point and probabilistic forecasts of hourly electricity prices for a random period in 2019 are presented in figure 5.12 and the red squares are the actual electricity prices. The solid blue line shows the day-ahead point forecasts obtained from the XGB regressor. The point forecasts follow the trends in actual prices, although there are uncertainties in the forecasting results. The prediction intervals are obtained by GBR and plotted as the gray-colored area. The darker gray colors denote smaller prediction intervals. The lightest and darkest gray colors indi-
cate 90% and 10% prediction intervals. As illustrated in the figure, the prediction intervals give more information about future electricity prices and compensate for point forecasts' uncertainty to some extent.

Figure 5.12: Probabilistic, Point, and Quantile Prediction.
In order to show some of the applications of point and probabilistic forecasts in the electricity supply chain, two types of operations are studied in this chapter. First, the power production of an electricity generator is modeled as an optimization problem. In addition, the amount of power supply that maximizes the power unit’s profit is obtained by using probabilistic forecasts of electricity load. Also, the energy procurement of a large consumer is modeled as another optimization problem, and two methods are proposed for solving the problem. The first method uses the point forecasts of electricity prices. The second method applies machine learning algorithms to find a strategy that maximizes the consumer’s daily procured power.

6.1 Power Production Planning

The structure of electricity supply and distribution is presented in figure 6.1. Power generation units are connected to the distribution centers to deliver the required electricity. Electricity is produced based on demand forecasts, with forecasting errors. As a result, a surplus or lack of electricity at the production time is expected. The connection to other electricity markets enables
power plants to handle the situation and retains the supply-demand condition in a balanced state. However, selling electricity to other markets or buying from them has some consequences. The electricity that is bought from other markets encompass transmission cost and has a higher price than local electricity. Similarly, the surplus electricity that is being sent to other markets may not have any profit. The congestion in transmission lines increases transmission costs and makes the situation even worse. Therefore, a proper schedule of power generation units is necessary to avoid any extra cost in production.

Consider a power plant with $N$ power generating units, and the cost of electricity production from the $i^{th}$ generator is $\lambda_i^G$. There are also $M$ offers of electricity at the quantity of $D_j$, $j = 1, \ldots, M$ and the prices of $\lambda_j^D$ from electricity distributors. Therefore, power plants need to decide on the quantity and price of their production that maximizes their profit:

$$\text{Maximize } \sum_{j=1}^{M} \lambda_j^D D_j - \sum_{i=1}^{N} \sum_{j=1}^{M} S_{ij} (T_{ij} + \lambda_i^G) \quad (6.1)$$

![Figure 6.1: Power supply and distribution architecture.](image)
subject to

\[
\begin{align*}
\sum_{i=1}^{N} \sum_{j=1}^{M} S_{ij} &= \sum_{j=1}^{M} D_j, \\
\sum_{i=1}^{N} S_{ij} &= D_j, \\%
\sum_{j=1}^{M} S_{ij} &\leq C_j, \quad 1 \leq j \leq M, \\
S_{ij} &\geq 0.
\end{align*}
\]

(6.2)

\(S_{ij}\) is the electricity produced by unit \(i\) for distribution center \(j\), and \(T_{ij}\) is the transmission cost associated with that trade. The revenue from power production as shown in equation 6.1 is a function of \(S_{ij}\) and the optimization problem aims to find proper \(S_{ij}\) for \(i \in [1, N]\) and \(j \in [1, M]\). Equation 6.2 represents the constraints of the problem, and \(C_i\) is the maximum capacity of the \(i^{th}\) generation unit. The problem stated in equation 6.1 can be solved by using the linear programming method.

A linear programming problem is defined as maximizing or minimizing a linear objective function subject to linear constraints. The constraints may be equality or inequality. The general form of a linear programming problem can be expressed as

\[
\begin{align*}
\text{Max } c^T x, \quad \text{Subject to :}
\end{align*}
\]

\[
\begin{align*}
Ax &= b \\
Cx &\leq d \\
x &\in X
\end{align*}
\]

(6.3)

Where \(c^T x\) represents the linear objective function, in which \(c\) is the coefficient vector. Moreover, \(Ax = b\) is the set of equalities that describe the system (e.g., production rates), and \(Cx \leq d\) is the set of inequalities that define the specifications or constraints for feasible plans and schedules. Besides, \(x \in X\) represents the corresponding solution space \(X\) for the continuous decision variable \(x\).
The revenue function and the constraints can be more complicated than the model we present here. For instance, each generation unit needs a certain amount of time to start up or shut down. Previous studies considered different assumptions, and as a result, several optimization methods have been applied in this context. Most of the optimization-related problems in the electricity supply system can be categorized under linear and non-linear programming [47], mixed-integer linear and non-linear programming [48], and quadratic programming [49]. Other optimization approaches include stochastic, convolution, regression, Lagrangian relaxation decomposition, and Newton-Raphson constrained linear least-squares programming, have also been applied in the context [50].

In equation 6.1, $D_i$ can be obtained by using point forecasting methods. However, electricity demand is a random variable, and it is more appropriate to consider the effect of demand uncertainty in the optimization problem. The probabilistic forecast of electricity demand is used in the decision analysis. The decision-making process includes two main components: (1) decisions, and (2) actual events that may occur in the future, known as the state of nature. As a matter of fact, at the time of scheduling generation units, the decision-makers in power plants are uncertain of nature. When a power plant detects a rise in electricity demand, it increases electricity production for future hours. If the demand remains high, the power plant will realize an increase in profit. However, if demand takes a downturn, the power plant will lose money. In this case, electricity demand determines the state of nature. Decision analysis helps decision-makers choose a proper schedule for their generation units.

The probabilistic forecasts of electricity demand provide some information about future states of nature to assign probabilities to their occurrence. The expected value criteria are used to calculate the state of nature and determine the best choice. The expected value is obtained by multiplying each outcome of a decision by the probability of its occurrence. The probabilistic forecast of electricity demand can be obtained by applying the same approach explained in chapter 5. Without loss of generality, I assume that there is a single distributor in the model. When
there are multiple distributors, the probabilistic forecasting process must be done several times to compute all the $D_i$s. The hourly Ontario electricity demand is publicly available from the IESO website. Electricity demand data from 2010 till 2019 are used to compute probabilistic forecasts of electricity demand. The results of point forecasts of electricity demand and prediction intervals are provided in figure 6.2.

![Figure 6.2: Point and probabilistic forecasts of electricity demand.](image)

For each hour, quantiles of electricity demand are computed using the GBR method. Then, the obtained quantiles are used to compute the realization of electricity demand for some intervals. For instance, according to the obtained probabilistic demand forecasts, it is expected that on 2019-03-17 at 4:00 pm, the electricity demand varies between 12 and 17 GWh. The whole range is divided into ten equal intervals; each has a 0.5 GWh length. The first and the last intervals are ignored in table 6.1 because they have very low probabilities. Table 6.1 provides the probability of other intervals. As mentioned earlier, surplus and lack of electricity production have consequences for electricity power plants. Additionally, it is assumed that power plants make 1 $/MWh profit.
by selling their electricity in the local market.

Table 6.1: States of Nature with Probabilities

<table>
<thead>
<tr>
<th>Demand (GWh)</th>
<th>(13,13.5)</th>
<th>(13.5,14)</th>
<th>(14,14.5)</th>
<th>(14.5,15)</th>
<th>(15,15.5)</th>
<th>(15.5,16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>0.05</td>
<td>0.14</td>
<td>0.31</td>
<td>0.31</td>
<td>0.14</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Actual electricity supply and demand are unpredictable and are always expected to have deficit or surplus power in production. Selling or buying the necessary power encompasses some costs and it is assumed that selling electricity to other markets costs 80 cents per MWh. When the supply is less than demand, the required power needs to be imported from other markets and this power may be bought from renewable or fossil fuel plants. As renewable power producers have a low carbon footprint, they usually offer lower prices than fossil fuel plants. Unlike renewable power plants, gas and coal plants can increase their power production at relatively high rates. Therefore, when renewable plants have no extra power to sell, the required power needs to be provided by the gas or coal power plants. It is also assumed that buying electricity from renewable and fossil fuel plants costs 45 and 65 cents per MWh, respectively. Transmission of electricity over the lines also increases the additional cost of exporting and importing the electricity. This cost may increase when the transmission lines get congested. Therefore, transmission costs of 40 cents/MWh (60 cents/MWh) for low (high) congested lines are considered. Based on the assumptions, the following four scenarios are presented in table 6.2.

Table 6.2: Different Scenarios That May Happen During Power Trading

<table>
<thead>
<tr>
<th>Importing from</th>
<th>Low Congestion in Transmission Lines</th>
<th>High Congestion in Transmission Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renewable Plants</td>
<td>Scenario I</td>
<td>Scenario II</td>
</tr>
<tr>
<td>Fossil Fuel Plants</td>
<td>Scenario III</td>
<td>Scenario IV</td>
</tr>
</tbody>
</table>

The payoff table for each scenario can be obtained by incorporating the assumed production and transmission costs. Table 6.3 shows the outcomes of power trading in scenario I for different
decisions (electricity supply). In other words, table 6.3 demonstrates the profit of power production for a different amount of production. Further, the states of nature in table 6.3 represent the midpoint of the intervals in table 6.1. The payoff tables for other scenarios can be obtained in the same way. The probabilities of states of nature are given in table 6.1 and the anticipated profit of each state is computed and provided in table 6.4. It follows from the table that for all scenarios, producing 14.75 GWh of electricity will maximize the power plant revenue, and the power generation units should be scheduled accordingly.

### Table 6.3: Payoff Table for Electricity Production in Scenario I

<table>
<thead>
<tr>
<th>Decision (GWh)</th>
<th>13.25</th>
<th>13.75</th>
<th>14.25</th>
<th>14.75</th>
<th>15.25</th>
<th>15.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.25</td>
<td>13,250</td>
<td>12,825</td>
<td>12,400</td>
<td>11,975</td>
<td>11,550</td>
<td>11,125</td>
</tr>
<tr>
<td>13.75</td>
<td>12,650</td>
<td>13,750</td>
<td>13,325</td>
<td>12,900</td>
<td>12,475</td>
<td>12,050</td>
</tr>
<tr>
<td>14.25</td>
<td>12,050</td>
<td>13,150</td>
<td>14,250</td>
<td>13,825</td>
<td>13,400</td>
<td>12,975</td>
</tr>
<tr>
<td>14.75</td>
<td>11,450</td>
<td>12,550</td>
<td>13,650</td>
<td>14,750</td>
<td>14,325</td>
<td>13,900</td>
</tr>
<tr>
<td>15.25</td>
<td>10,850</td>
<td>11,950</td>
<td>13,050</td>
<td>14,150</td>
<td>15,250</td>
<td>14,825</td>
</tr>
<tr>
<td>15.75</td>
<td>10,250</td>
<td>11,350</td>
<td>12,450</td>
<td>13,550</td>
<td>14,650</td>
<td>15,750</td>
</tr>
</tbody>
</table>

### Table 6.4: Expected Profit in Thousands of Dollars for Each Decision

<table>
<thead>
<tr>
<th>Decision (GWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario</td>
</tr>
<tr>
<td>II</td>
</tr>
<tr>
<td>III</td>
</tr>
<tr>
<td>IV</td>
</tr>
</tbody>
</table>

### 6.2 Energy Procurement Strategy for Large Consumers

Large consumers such as aluminum and steel production plants have comparatively high electricity consumption levels. Large consumers may own many loads that are distributed throughout the power transmission network. In this section, the energy procurement of a large consumer is
modeled and solved. Operations of large consumers can be in either elastic or inelastic modes. Inelastic mode refers to the situation where the large consumer is insensitive to electricity prices and is willing to buy electricity at any price. For example, a hospital cannot postpone surgery operations to the hours that electricity has low prices. The hospitals need to have a continuous electricity supply. On the other hand, the elastic mode refers to how the large consumer is sensitive to electricity prices. It means that the large consumer works at its maximum capacity when the electricity price is low and decreases power consumption as least as possible when electricity has high prices. For example, a steel company may prefer to increase its production when electricity has low prices. In both modes, the objective of the large consumer is to minimize the cost of buying electricity.

The goal is to develop a procurement strategy for a large consumer that works in elastic mode. One day before the operations, the large consumer needs to plan for buying electricity via the day-ahead electricity market or bilateral contracts. A bilateral contract is an agreement between a power plant and the large consumer for buying electricity at a fixed rate. Based on the day-ahead forecasts of electricity prices, the large consumer decides on buying electricity from the day-ahead market or bilateral contracts. Moreover, suppose some of the consumer’s offers are rejected in the day-ahead market. In that case, the electricity needs to be procured from the intra-day market at higher prices. Accurate day-ahead forecasts of electricity prices may help consumers to apply proper strategies for buying electricity. Figure 6.3 demonstrates offers of the large consumer for a typical day. The consumer offers in figure 6.3 looks relatively close to the market-clearing prices. However, it is scheduled to buy electricity for 14 hours. Furthermore, during the rest of the day, consumer offers are not high enough, and it has to buy electricity from the intra-day market.

A linear programming approach can be used to find an optimum strategy for the energy procurement of a large consumer. Assume that \( p_{d,h} \) is the electricity price at the \( h^{th} \) hour \(( h \in [1,24])\) of day \( d \in [1,365]\), and \( \hat{p}_{d,h} \) is the corresponding forecast computed by the XGB
method. The goal is to determine how much electricity the consumer should offer in the day-ahead market and buy from bilateral contracts that maximize its procured energy. In the problem formulation, $\zeta$ is the threshold that determines whether to buy electricity from the market or through bilateral contracts. Also, $\mathcal{H}^{(m)}_d$ and $\mathcal{H}^{(b)}_d$ are the sets of hours of day $d$ that $\hat{p}_{d,h} < \zeta$, and $\hat{p}_{d,h} \geq \zeta$, respectively. Suppose that $\tau_{d,h}$ represents the offer placed in the market for $h^{th}$ hour. Moreover, there are some restrictions for the procurement strategy as

- **Constraint 1:** The consumer has a limited hourly capacity, and the amount of purchased electricity at each time can be a maximum of $\tau_{h}^{max}$. Also, the amount of purchasing power at each hour should be greater than $\tau_{h}^{min}$.

- **Constraint 2:** The total cost of procured electricity for a whole day (24 hours) has to be at most $M$.

- **Constraint 3:** The difference between power purchased at two consecutive hours cannot
Energy procurement of the large consumer can be formulated as a linear programming (optimization) problem as

\[
\text{Maximize } C_{lp}^{d} = \sum_{h=1}^{24} \tau_{lp}^{d,h},
\]

subject to

\[
\tau_{h}^{\text{min}} \leq \tau_{lp}^{d,h} \leq \tau_{h}^{\text{max}} \quad \text{(Constraint 1)}
\]

\[
\sum_{h \in H_{d}^{(m)}} \tau_{lp}^{d,h} \hat{p}_{d,h} + \sum_{h \in H_{d}^{(b)}} \tau_{lp}^{d,h} \zeta \leq M \quad \text{(Constraint 2)}
\]

\[-\mathcal{R} \leq \tau_{d,h} - \tau_{d,h-1} \leq \mathcal{R}, \quad \forall h > 1 \quad \text{(Constraint 3)}\]

(6.4)

(6.5)

(6.6)

Solving the problem provides a strategy to participate in the market or buying contracts. The superscript \(lp\) indicates that \(\tau_{lp}^{d,h}\) is obtained from a linear programming model. The PuLP library in Python is used to solve the linear programming problem modeled in equation 6.4. After the market operator informs the participants about market-clearing prices, the large consumer’s total procured energy is equal to

\[
K_{lp}^{d} = \sum_{h \in H_{d}^{(m)}} \tau_{lp}^{d,h} 1_{\{p_{d,h} \leq \hat{p}_{d,h}\}} + \sum_{h \in H_{d}^{(b)}} \tau_{lp}^{d,h}.
\]

(6.6)

It follows from equation 6.6 that only those offers are accepted in the day-ahead market that are greater than the market-clearing prices. If the consumer needs to buy more power than \(K_{lp}^{d}\), it has to buy from the intra-day market for a higher price.

In the second approach, the large consumer’s power procurement problem is modeled and solved using the obtained quantile of day-ahead electricity prices. Suppose point forecasts of electricity prices are less than the actual observations. In that case, the strategy’s efficiency obtained from equation 6.4 will decline. The smallest quantile that is greater than the actual
observation is a better option than the point forecasts. Machine learning algorithms can be used to find a set of quantiles that can provide a better energy procurement strategy than point forecasts. The machine learning algorithm can be a neural network or gradient boosting trees.

First, quantile forecasts of day-ahead electricity prices from 2015 till 2019 are used to create a training data set. The machine is trained using 2015-2018 set of \( \{ Q_{d,h}, \hat{q}_{d,h} \} \), where \( Q_{d,h} \) is a \( 100 \times n_s \) matrix; each of its row contains 100 quantiles of electricity prices for a particular hour. Furthermore, \( \hat{q}_{d,h} \) is the smallest quantile in \( Q_{d,h} \) that is the closest to the actual observation at the corresponding hour. The machine tunes its parameters using the testing data, then receives 2019 quantile forecasts of electricity prices \( Q_{d,h} \) to generate \( \hat{q}_{d,h} \). As the machine receives a \( 100 \times 1 \) dimension input, employing gradient boosting trees is more appropriate than neural networks. The XGB method is used to find a set of \( \hat{q}_{d,h} \) for 2019 and to model the energy procurement of the large consumer. The optimization problem is almost the same as equation 6.4 and can be written as

\[
\text{Maximize } C_{xgb}^{d} = \sum_{h=1}^{24} \tau_{xgb}^{d,h}, \tag{6.7}
\]

subject to

\[
\tau_{h}^{min} \leq \tau_{xgb}^{d,h} \leq \tau_{h}^{max} \quad \text{(Constraint 1)}
\]

\[
\sum_{h \in \mathcal{H}_{d}^{(m)}} \tau_{xgb}^{d,h} \hat{q}_{d,h} + \sum_{h \in \mathcal{H}_{d}^{(b)}} \tau_{xgb}^{d,h} \zeta \leq M \quad \text{(Constraint 2)}
\]

\[
-\mathcal{R} \leq \tau_{xgb}^{d,h} - \tau_{xgb}^{d,h-1} \leq \mathcal{R}, \quad \forall h > 1 \quad \text{(Constraint 3)}
\]

where, \( \mathcal{H}_{d}^{(m)} \) and \( \mathcal{H}_{d}^{(b)} \) indicate hours of day \( d \) that \( \hat{q}_{d,h} < \zeta \), and \( \hat{q}_{d,h} \geq \zeta \), respectively. Solving the optimization problem provides the large consumer’s total procured energy as

\[
K_{xgb}^{d} = \sum_{h \in \mathcal{H}_{d}^{(m)}} \tau_{xgb}^{d,h} 1_{\{ p_{d,h} \leq \hat{q}_{d,h} \}} + \sum_{h \in \mathcal{H}_{d}^{(b)}} \tau_{xgb}^{d,h}. \tag{6.9}
\]
The average of daily energy procurement costs for the hourly electricity prices of 2019 is used to compare the results of the two models. The initial values for simulation parameters are provided in table 6.5. The daily procured electricity is computed under two scenarios. Scenario

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>$\tau_h^{\min}$</th>
<th>$\tau_h^{\max}$</th>
<th>$M$</th>
<th>$\zeta$</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value (MW)</td>
<td>15</td>
<td>100</td>
<td>45</td>
<td>1000</td>
<td>20</td>
</tr>
</tbody>
</table>

$I$ assumes that the large consumer buys electricity only from the day-ahead market. And, in scenario $II$, the electricity can be bought from the day-ahead electricity market and bilateral contracts. Figure 6.4 illustrates the daily procured electricity for the second week of October 2019 in scenario $I$. The dashed line with red squares represents an ideal situation when the large consumer has perfect forecasts of day-ahead electricity prices. The ideal situation never happens, and it is plotted to give a sense about the performance of the proposed methods. The dashed line with green triangles denotes the procurement strategy obtained by the XGB point forecasts of electricity price. Also, the dashed line with yellow circles indicates the procurement strategy obtained by the XGB quantiles. It follows from the data analysis that XGB quantiles improve the procurement strategy obtained by the XGB point forecasts of electricity prices.

A comparison of daily procured electricity under the two scenarios for the second week of August 2019 is provided in figures 6.5 and 6.6. As expected, adding bilateral contracts to the sources of power supply improves the performance of procurement strategies. Table 6.6 presents the average of daily procured electricity under the two scenarios. It arises from the analysis that XGB quantiles improved the solution of equation 6.4 for scenarios $I$ and $II$ by 7% and 8.8%, respectively.
Figure 6.4: Daily procured energy from day-ahead market.

Table 6.6: Average of Daily Procured Electricity Under the Two Scenarios

<table>
<thead>
<tr>
<th>Source of Procured Electricity</th>
<th>Actual Prices</th>
<th>XGB Point Forecasts</th>
<th>XGB Quantile Forecasts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day-Ahead Market</td>
<td>657.14</td>
<td>371.22</td>
<td>397.08</td>
</tr>
<tr>
<td>Day-Ahead Market and Bilateral Contracts</td>
<td>692.85</td>
<td>405.84</td>
<td>441.36</td>
</tr>
</tbody>
</table>
Figure 6.5: Daily procured energy from day-ahead market.

Figure 6.6: Daily procured energy from day-ahead market and bilateral contracts.
Chapter 7

Conclusion and Future Research

Electricity price forecasting plays an important role in decision-making on bidding strategies of selling and buying electricity. Recently, there has been a growing interest in studying probabilistic forecasts of electricity prices as it provides more information about future electricity prices than the point forecasts. In this thesis, one day-ahead quantile forecasts of electricity prices in a highly volatile market are computed by applying regression models to a pool of point forecasts. Three electricity markets: Ontario, PJM, and Nord Pool, are compared in terms of the electricity price volatility. Then, three data-driven forecasting methods are implemented to generate day-ahead point forecasts of the Ontario market’s electricity prices, which have the highest volatility amongst the other two ones.

To generate the first set of electricity price point forecasts, I use a relatively simple idea to break the historical hourly electricity price data into 24 subsets. Each subset has electricity prices of a particular hour. The triple exponential smoothing method is applied to each subset and generates point forecasts of the corresponding hour. The TES method decreases the mean absolute error of a benchmark model from 10.29 to 9.42. The benchmark model considers the forecast of electricity prices for each hour of the day-ahead as the average price of the same hour and day during the last four weeks.
The second method consists of a Convolutional layer combined with gradient recurrent units. The neural network gets hourly electricity prices as input and gives the day-ahead point forecasts of electricity prices. It results in an MAE of 8.20. The third method uses the extreme gradient boosting regression approach and obtains another set of day-ahead point forecasts. The XGB method uses explanatory variables such as the day-ahead forecasts of electricity demand and price volatility in addition to the historical electricity prices. The XGB regression approach further decreases the MAE to 7.06.

A dual-phase model for short-term load forecasting is proposed. In the first phase, a neural network consists of convolutional and GRU neurons that is used to generate electricity load forecasts. In the second phase, several regression methods such as Lasso, SVR, GBR, and XGB use the results of phase one along with meteorological variables to compute short-term forecasts of electricity demand. The data analysis shows that the XGB method provides the most accurate (least MAE) forecasts of day-ahead electricity demand. It is shown that the ConvGRU-XGB method outperforms the other methods. The data analysis shows a significant improvement in terms of performance where the maximum mean absolute error decreases from 415.99 to 339.16.

The double and triple exponentially smoothing methods are used to compute day-ahead forecasts of electricity price volatility. Further, the Elastic Net regularization method is used to enhance the forecasting performance of the model. The regularization step improved TES volatility forecasting results by decreasing the mean absolute error from 11.98 to 11.07. Regularized volatilities are used as a feature for day-ahead electricity prices forecasting. Finally, the quantile regression averaging method is applied to the pool of point forecasts obtained by TES, ConvGRU, and XGB methods to compute quantile forecasts of day-ahead electricity prices. Moreover, the QRA method is further developed by employing GBR. It follows from the data analysis that the GBR method provides more reliable quantiles and tighter prediction intervals with smaller forecasting errors than QRA.

The scheduling problem of the power plants is studied as an application of probabilistic
First, day-ahead probabilistic forecasts of electricity demand are obtained. Then quantiles of electricity demand are used in decision analysis to find the optimum amount of power production. Due to the electricity market’s non-linear behavior, the optimum electricity supply is derived under four different scenarios. The scenarios are defined by incorporating the cost of buying electricity from renewable or fossil fuel power plants and the transmission costs. Analysis shows that the optimum amount of supply maximizes the revenue in all of the scenarios.

Energy procurement of a large electricity consumer is also modeled as another optimization problem. The consumer’s objective is to find a bidding strategy that maximizes the daily procured electricity from the day-ahead market and bilateral contracts. Point and probabilistic forecasts of electricity prices are used to model and solve the optimization problem. The XGB method is used to find the smallest quantile, which is greater than the actual electricity price. The linear programming method is applied to the models, and two power procurement strategies are provided. It follows from the simulation results that obtained quantiles improve the bidding strategy of the large consumer.

The data-driven forecasting models that are implemented in this thesis have implications in electricity supply forecasting as well. In particular, renewable power plants can use the forecasting models to compute quantile forecasts of wind or solar power supply. As mentioned, electricity can have negative prices, and therefore there is always some level of risk attached to electricity production. Probabilistic forecasts of electricity prices improve the operation of renewable power plants by minimizing the risk of selling electricity at negative prices. However, the explanatory variables for electricity supply are not the same as what this thesis suggests for the electricity demand and price, and there needs to be extensive research to find proper feature variables.

Future studies may apply other time series forecasting programming packages such as Prophet, which is developed by Facebook. This package is able to capture the trend and seasonality of time series, and as a result, it is a proper tool for long-term forecasts of electricity supply and demand. It is expected that using electricity supply and demand historical data as features improves the
PFEP-HOEP

results of day-ahead electricity price forecasting.
References


