

# **Deep Learning Techniques for Electrical Load Forecasting**

by

Tolulope Oluwaseun Olugbenga

BSc. in Computer Science Engineering, University of Debrecen, 2018

A Thesis Submitted in Partial Fulfillment  
of the Requirements for the Degree of

**Master of Science in Engineering**

in the Graduate Academic Unit of Electrical and Computer Engineering

Supervisors: Dr. Dawn MacIsaac, Ph.D., Electrical and Computer Engineering  
Dr. Julian Cardenas, Ph.D., Electrical and Computer Engineering

Examining Board: Dr. Julian Meng, Ph.D., Electrical and Computer Engineering  
Dr. Yevgen Biletskiy, Ph.D., Electrical and Computer Engineering  
Dr. Michael Fleming, Ph.D., Computer Science

This thesis is accepted by the  
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

February 2022

© Tolulope Oluwaseun Olugbenga, 2022

## **ABSTRACT**

Load forecasting is critical for power system operators to maintain a safe and efficient network. Load forecasting contributes to the supply-demand balance by ensuring that consumers receive adequate energy. Load aggregators, power marketers, and independent system operators can all benefit from load forecasting. Over-forecasting leads to excess production and waste of resources. An unexpectedly high load results in a power outage. Both scenarios result in inefficient generation scheduling and technical difficulties for the operator. It is not simple to create a forecasting model for a specific power network. Statistical and machine-learning techniques have been used in load forecasting. Deep learning techniques have recently gained popularity due to their improved ability to interpret complex data relationships. The purpose of this study was to compare deep learning forecasting techniques to some conventional forecasting techniques used by utilities to see if deep learning can better meet their needs.

## **DEDICATION**

This thesis is dedicated to my future self; I want him to look back and realize that his anguish, struggle, and late nights were not in vain. I adore you, and I am excited to meet the man you will become.

## **ACKNOWLEDGEMENTS**

This is, without a doubt, one of the most challenging journeys I have ever undertaken. It has shown me how much I can grow and accomplish when I believe in myself and put in the work. It was an eye-opening experience, and I am grateful that I did not give up and instead persevered in my efforts to cross the finish line. When I first arrived at UNB and saw what my colleagues in the lab were working on, I honestly wondered what I had gotten myself into. After hearing from other students about how difficult it is to complete a thesis master's degree, I began to doubt my abilities and became concerned that I would not complete it.

I would like to express my gratitude to my supervisors, Dr. Dawn MacIsaac and Dr. Julian Cardenas; without them, I would not have been able to complete this program. I appreciate your patience and encouraging words, which reminded me that anything is possible if I keep going. I would also like to express my gratitude to my family for always being there and motivating me to complete this program.

To be completely candid, the individual who entered the lab in December 2018 would not have been able to complete this degree. To complete, I needed to improve on both a personal and intellectual level. I want to applaud myself for not throwing in the towel and giving up; I want to commend myself for persevering through difficult times and even when the going became tougher. In a nutshell, this has been an educational experience and a game of physical and cognitive development. I would do it all over again if I had to because I would not be the man I am today without it. As a result, I would like to convey my appreciation to my supervisors and the University of New Brunswick for providing me with this opportunity to learn and develop as a person.

## Table of Contents

ABSTRACT .....	ii
DEDICATION .....	iii
ACKNOWLEDGEMENTS .....	iv
Table of Contents .....	v
List of Tables .....	x
List of Figures .....	xi
List of Abbreviations .....	xv
1 Introduction .....	1
1.1 Objectives .....	3
2 Overview of Load Forecasting .....	5
2.1 Factors That Affect the Load Demand .....	5
2.2 Load Forecasting Horizons .....	7
2.3 The Benchmark Forecasters .....	8
2.3.1 The Seasonal Naive Forecaster (SNF) .....	8
2.3.2 The Multiple Linear Regression Forecaster (MLR) .....	9
2.3.3 The Auto-Regressive Integrated Moving Average Forecaster (ARIMA) .....	10
2.3.4 Artificial Neural Network Short Term Load Forecaster – Generation Three (ANNSTLF-G3) .....	13
2.4 Deep Learning Forecasters .....	16
2.4.1 Recurrent Neural Networks and the Long Short Term Memory Network .....	19
2.4.2 The Convolutional Neural Network (CNN) .....	21
2.5 Peak Load .....	24

2.6 Performance Metrics .....	26
2.7 The Myth of Finding the One Size Fits All Technique .....	27
3 Investigation.....	28
3.1 Preparation of the Datasets .....	28
3.2 Implementation Specifications for Benchmark Forecasters .....	30
3.2.1 The Seasonal Naïve Forecaster (SNF) .....	31
3.2.2 The Multiple Linear Regression Forecaster (MLR) .....	31
3.2.3 The Seasonal Auto-Regressive Integrated Moving Averages with Exogenous Regressors Forecaster (SARIMAX) .....	32
3.2.4 The Artificial Neural Network Short Term Load Forecaster (ANNSTLF-G3)	33
3.3 Implementation Specifications for the Deep Learning Forecasters .....	35
3.3.1 The Long Short Term Memory Forecaster (LSTM).....	35
3.3.2 The Convolutional Neural Network Forecaster (CNN).....	36
3.4 Performance Analysis .....	36
3.4.1 A Remark on Peak Detection Accuracy Metrics .....	37
3.5 The Performance of Forecasters on the Toronto Dataset.....	38
3.6 The Performance of Forecasters on the Ottawa Dataset.....	40
3.7 The Performance of Forecasters on the Saint John Dataset.....	41
3.8 Conclusion .....	43
4 Comprehensive Evaluation of the Forecasters' Performance.....	44
4.1 The Toronto Dataset .....	44

4.1.1 The Hourly Performance.....	45
4.1.2 The Daily Performance .....	46
4.1.4 Performance During the Seasons .....	49
4.1.5 Comprehensive Analysis Discussion.....	49
4.2 The Ottawa Dataset.....	51
4.2.1 The Hourly Performance.....	51
4.2.2 The Daily Performance .....	53
4.2.3 The Monthly Performance .....	54
4.2.4 Performance During the Seasons .....	56
4.2.5 Comprehensive Analysis Discussion.....	57
4.3 The Saint John Dataset.....	58
4.3.1 The Hourly Performance.....	59
4.3.2 The Daily Performance .....	60
4.3.3 The Monthly Performance .....	62
4.3.4 Performance During the Seasons .....	64
4.3.5 Comprehensive Analysis Discussion.....	65
5 Conclusion .....	67
5.1 Summary .....	67
5.2 Contributions.....	68
5.3 Future Work .....	69

Bibliography .....	71
Appendix A .....	93
1 Determining the SARIMAX Model's Optimal Parameters .....	93
1.1 Statistical Analysis of the Toronto Dataset.....	94
1.2 Statistical Analysis of the Ottawa Dataset.....	97
1.3 Statistical Analysis of the Saint John Dataset.....	100
Appendix B .....	104
1 Metrics for Overall Accuracy .....	104
1.1 The Toronto Dataset's Overall Performance Metrics .....	104
1.2 The Ottawa Dataset's Overall Performance Metrics.....	104
1.3 The Saint John Dataset's Overall Performance Metrics.....	104
2 Other Forecasters' Box Plots of the Error Distribution.....	105
2.1 The Toronto Dataset .....	105
2.1.1 The Hourly Performance.....	105
2.1.2 The Daily Performance .....	107
2.1.3 The Monthly Performance .....	108
2.2 The Ottawa Dataset.....	110
2.2.1 The Hourly Performance.....	110
2.2.2 The Daily Performance .....	111
2.2.3 The Monthly Performance .....	113
2.3 The Saint John Dataset.....	114
2.3.1 The Hourly Performance.....	114

2.3.2 The Daily Performance .....	116
2.3.3 The Monthly Performance .....	117
Curriculum Vitae	

## List of Tables

Table 1 - Formulas for Several Frequently Used Performance Metrics .....	26
Table 2 – The Training and Testing Dataset Ranges Across All Datasets .....	30
Table 3 - The MLR Forecaster's Independent Variables .....	32
Table 4 - The SARIMAX hyperparameters that were used for each dataset.....	32
Table 5 – LSTM Networks Training Options.....	35
Table 6 – CNN Networks Training Options .....	36
Table 7 - Overall MAPE and RMSE for Each Forecaster – Toronto Dataset .....	39
Table 8 - Matrix Analysis of Peak Values and Time Difference – Toronto Dataset.....	40
Table 9 - Overall MAPE and RMSE for Each Forecaster – Ottawa Dataset .....	41
Table 10 - Matrix Analysis of Peak Values and Time Difference – Ottawa Dataset .....	41
Table 11 - Overall MAPE and RMSE for Each Forecaster – Saint John Dataset .....	42
Table 12 - Matrix Analysis of Peak Values and Time Difference – Saint John Dataset ..	42
Table 13 - Seasonal MAPE and RMSE for the Toronto Dataset.....	49
Table 14 - Seasonal MAPE and RMSE for the Ottawa Dataset .....	56
Table 15 - Seasonal MAPE and RMSE for the Saint John Dataset.....	64
Table 16 - The SARIMAX hyperparameters that were used across all datasets .....	93
Table 17 – The Overall Performance Metrics – Toronto Dataset.....	104
Table 18 - The Overall Performance Metrics – Ottawa Dataset.....	104
Table 19 - The Overall Performance Metrics – Saint John Dataset .....	104

## List of Figures

Figure 1 - The Block Diagram of the Third Generation ANNSTLF [37] .....	13
Figure 2 - The Structure of a Simple Feed-forward ANN .....	14
Figure 3 - Unrolled Recurrent Neural Network (RNN) [140] .....	19
Figure 4 - The Block of Long-Term Short-Term Memory [141] .....	20
Figure 5 - A Simple One-Dimensional CNN's Architecture [165].....	22
Figure 6 – Peak Load vs Base Load [168].....	25
Figure 7 – 2019 Average Daily Demand for Loads Across All Datasets .....	29
Figure 8 – The Structure of the BLF and CLF Network.....	34
Figure 9 - Load Demand on March 11, 2019, and CNN Forecast – Toronto Dataset.....	38
Figure 10: (a) Overall Error Distribution for All Forecasters; (b) Actual and Forecasted Load Demand for July 17th-21st - Toronto Dataset .....	39
Figure 11: (a) Overall Error Distribution for All Forecasters; (b) Actual and Forecasted Load Demand for July 17th-21st - Ottawa Dataset .....	40
Figure 12: (a) Overall Error Distribution for All Forecasters; (b) Actual and Forecasted Load Demand for December 17th-21st – Saint John Dataset.....	42
Figure 13: (a) Hourly MAPE for All Forecasters, (b-d) Hourly Error Distributions for CNN, LSTM, and ANN Forecasters – Toronto Dataset .....	45
Figure 14: (a) Daily MAPE for All Forecasters, (b-d) Daily Error Distributions for CNN, LSTM, and ANN Forecasters – Toronto Dataset .....	47
Figure 15: (a) Monthly MAPE for All Forecasters, (b-d) Monthly Error Distributions for CNN, LSTM, and ANN Forecasters – Toronto Dataset.....	48
Figure 16 - Scatter Plot of Load Demand versus Temperature – Toronto Dataset.....	50

Figure 17: (a) Hourly MAPE for All Forecasters, (b-d) Hourly Error Distributions for CNN, LSTM, and ANN Forecasters – Ottawa Dataset.....	52
Figure 18: (a) Daily MAPE for All Forecasters, (b-d) Daily Error Distributions for CNN, LSTM, and ANN Forecasters – Ottawa Dataset.....	54
Figure 19: (a) Monthly MAPE for All Forecasters, (b-d) Monthly Error Distributions for CNN, LSTM, and ANN Forecasters – Ottawa Dataset .....	55
Figure 20 - Scatter Plot of Load Demand versus Temperature – Ottawa Dataset.....	57
Figure 21: (a) Hourly MAPE for All Forecasters, (b-d) Hourly Error Distributions for CNN, LSTM, and ANN Forecasters – Saint John Dataset .....	59
Figure 22: (a) Daily MAPE for All Forecasters, (b-d) Daily Error Distributions for CNN, LSTM, and ANN Forecasters – Saint John Dataset .....	61
Figure 23 - Monthly MAPE for Each Forecaster – Saint John Dataset.....	62
Figure 24 - Monthly Error Distribution for CNN, LSTM, ANN, and SARIMAX Forecasters – Saint John Dataset.....	63
Figure 25 - Scatter Plot of Load Demand versus Temperature – Saint John Dataset.....	65
Figure 26 – Excerpt from the Toronto Dataset .....	94
Figure 27 – Plot of the Initial Auto Correlation – Toronto Dataset.....	95
Figure 28 – ACF Plot Following Seasonal Differencing – Toronto Dataset.....	95
Figure 29 – ACF Plot After Seasonal and Non-Seasonal Differencing – Toronto Dataset .....	96
Figure 30 - PACF Plot After Seasonal and Non-Seasonal Differencing – Toronto Dataset .....	97
Figure 31 - Excerpt from the Ottawa Dataset .....	98

Figure 32 - Plot of the Initial Auto Correlation – Ottawa Dataset.....	98
Figure 33 - ACF Plot Following Seasonal Differencing – Ottawa Dataset.....	99
Figure 34 - ACF Plot After Seasonal and Non-Seasonal Differencing – Ottawa Dataset	99
Figure 35 - PACF Plot After Seasonal and Non-Seasonal Differencing – Ottawa Dataset	
.....	100
Figure 36 - Excerpt from the Saint John Dataset.....	101
Figure 37 - Plot of the Initial Auto Correlation – Saint John Dataset.....	102
Figure 38 - ACF Plot Following Seasonal Differencing – Saint John Dataset.....	102
Figure 39 - ACF Plot After Seasonal and Non-Seasonal Differencing – Saint John Dataset	
.....	103
Figure 40 - PACF Plot After Seasonal and Non-Seasonal Differencing – Saint John Dataset	
.....	103
Figure 41 - Hourly Error Distribution for the MLR Forecaster – Toronto Dataset.....	105
Figure 42 - Hourly Error Distribution for the SARIMAX Forecaster – Toronto Dataset	
.....	106
Figure 43 - Hourly Error Distribution for the SNF Forecaster – Toronto Dataset .....	106
Figure 44 - Daily Error Distribution for the MLR Forecaster – Toronto Dataset .....	107
Figure 45 - Daily Error Distribution for the SARIMAX Forecaster – Toronto Dataset	107
Figure 46 - Daily Error Distribution for the SNF Forecaster – Toronto Dataset.....	108
Figure 47 - Monthly Error Distribution for MLR Forecaster– Toronto Dataset .....	108
Figure 48 - Monthly Error Distribution for SARIMAX Forecaster– Toronto Dataset...	109
Figure 49 - Monthly Error Distribution for SNF Forecaster– Toronto Dataset.....	109
Figure 50 - Hourly Error Distribution for the MLR Forecaster – Ottawa Dataset .....	110

Figure 51 - Hourly Error Distribution for the SARIMAX Forecaster – Ottawa Dataset	110
Figure 52 - Hourly Error Distribution for the SNF Forecaster – Ottawa Dataset.....	111
Figure 53 - Daily Error Distribution for the MLR Forecaster – Ottawa Dataset.....	111
Figure 54 - Daily Error Distribution for the SARIMAX Forecaster – Ottawa Dataset..	112
Figure 55 - Daily Error Distribution for the SNF Forecaster – Ottawa Dataset .....	112
Figure 56 - Monthly Error Distribution for MLR Forecaster – Ottawa Dataset.....	113
Figure 57 - Monthly Error Distribution for SARIMAX Forecaster – Ottawa Dataset...	113
Figure 58 - Monthly Error Distribution for SNF Forecaster – Ottawa Dataset .....	114
Figure 59 - Hourly Error Distribution for the MLR Forecaster – Saint John Dataset....	114
Figure 60 - Hourly Error Distribution for the SARIMAX Forecaster – Saint John Dataset .....	115
Figure 61 - Hourly Error Distribution for the SNF Forecaster – Saint John Dataset .....	115
Figure 62 - Daily Error Distribution for the MLR Forecaster – Saint John Dataset .....	116
Figure 63 - Daily Error Distribution for the SARIMAX Forecaster – Saint John Dataset .....	116
Figure 64 - Daily Error Distribution for the SNF Forecaster – Saint John Dataset.....	117
Figure 65 - Monthly Error Distribution for MLR Forecaster – Saint John Dataset .....	117
Figure 66 - Monthly Error Distribution for SNF Forecaster – Saint John Dataset.....	118

## List of Abbreviations

<b>ANN</b>	Artificial Neural Networks
<b>ANNSTLF</b>	Artificial Neural Networks Short Term Load Forecaster
<b>ARIMA</b>	Auto-Regressive Integrated Moving Average
<b>ACF</b>	Auto Correlation Function
<b>BLF</b>	Base Load Forecaster
<b>CLF</b>	Change in Load Forecaster
<b>CNN</b>	Convolutional Neural Networks
<b>DL</b>	Deep Learning
<b>GRU</b>	Gated Neural Networks
<b>ICDAR</b>	International Conference on Document Analysis and Recognition
<b>ILSVRC</b>	ImageNet Large Scale Visual Recognition Challenge
<b>ISBI</b>	International Symposium on Biomedical Imaging
<b>LSTM</b>	Long Short-Term Memory networks
<b>LTLF</b>	Long Term Load Forecasting
<b>MAE</b>	Mean Absolute Error
<b>MAPE</b>	Mean Absolute Percentage Error
<b>MATLAB</b>	Matrix Laboratory
<b>MBE</b>	Mean Biased Error
<b>MICCAI</b>	Medical Image Computing and Computer-Assisted Intervention Society
<b>ML</b>	Machine Learning
<b>MLP</b>	Multilayer Perceptron
<b>MLR</b>	Multiple Linear Regression

<b>MSE</b>	Mean Squared Error
<b>MTLF</b>	Medium-Term Load Forecasting
<b>MW</b>	Mega-Watt
<b>PACF</b>	Partial Auto Correlation Function
<b>ReLU</b>	Rectified Linear Unit
<b>RLS</b>	Recursive Least Squares
<b>RMSE</b>	Root Mean Square Error
<b>RNN</b>	Recurrent Neural Networks
<b>SARIMAX</b>	Seasonal Auto-Regressive Integrated Moving Averages with Exogenous regressors
<b>SNF</b>	Seasonal Naïve Forecaster
<b>STD</b>	Standard Deviation
<b>STLF</b>	Short Term Load Forecasting
<b>SVM</b>	Support Vector Machine
<b>UNB</b>	University of New Brunswick
<b>VMD</b>	Variational Mode Decomposition

## **1 Introduction**

Load forecasting is an important component of electric utility design, planning, and operation; it has been used in the power industry for over a century [1]–[8]. Load forecasting is a critical building component for power system operators to ensure the network operates continuously and is managed safely and efficiently. The goal of load forecasting is to keep supply and demand balanced so that consumers have an adequate and cost-effective energy supply. Load forecasting can, however, be useful to organizations other than electric utilities, such as load aggregators, power marketers, independent system operators, regulatory commissions, and even industrial/commercial companies, banks, trading firms, and insurance companies [1], [9]. Load forecasting is used by these organizations in power system planning/operations, revenue projection, rate design, energy trading, and other activities [2], [3], [10].

Over the last decade, there has been a surge in the adoption of renewable energy and distributed generation sources, as well as the advancement and implementation of smart grids and buildings to effectively meet rising energy demands. To integrate these developments without causing system disruptions, reliable load forecasting across multiple time horizons is required [11]. Electric load forecasting has been extensively researched [1], [7], [12], [13], with the majority of current research focusing on developing more accurate forecasts. Because of the deregulation of energy markets and several random factors, often governed by human behavior, the demand patterns used to drive modern technologies are complex and must be considered when forecasting future electricity demand. As a result, developing a forecasting model suitable for a specific power network

is not an easy task [3], [10], [14]. Over-forecasting, or forecasting more power than needed, results in the start-up of an excessive number of generating units, resulting in over-production and unnecessary expense. Conversely, underestimating the required demand because of higher-than-expected loads creates an electricity deficit. When this happens, the system operator is forced to buy potentially expensive peaking power to make up the difference, which is significantly higher than the market price. Both situations result in suboptimal generation scheduling and present the operator with technical challenges.

The Texas Electric Reliability Council documented a power system incident in February 2008 that prompted them to respond to a faster-than-expected evening load ramp-up in order to maintain load/generation balance [15]. They relied on willing power consumers to act as temporary curtailment loads to draw on reserve power and alleviate demand. Following-incident analysis revealed that more accurate forecasting of generation and demand could easily have avoided the need for an emergency response. A heatwave in South Korea in September 2011 significantly increased electricity demand. South Korea's power supply was disrupted for nearly 1.5 million people [16] due to a lack of available energy to meet the uptick caused by the heatwave. Although these scenarios are uncommon, they provide extreme examples of the potential consequences of imbalance and, as a result, the critical importance of accurate load forecasting.

Statistical techniques and machine learning (ML) have both been used to forecast load, and the distinction between these two techniques is becoming increasingly blurred with the widespread adoption of data science [1]. Machine learning techniques are more intelligent and capable of forecasting load more accurately than statistical techniques. While both statistical and machine learning techniques have been extensively reported in the literature

on load forecasting, deep learning techniques have only recently gained popularity due to their ability to interpret complex relationships in data more accurately [2], [17]. Deep learning techniques have proven to be extremely effective in dealing with complex sequential data [18], [19]. As a result, deep learning techniques have been successfully applied to load forecasting applications, outperforming a variety of benchmark models such as simple Artificial Neural Networks (ANN) and more traditional statistical time series forecasters such as the Auto-regressive Integrated Moving Average (ARIMA) [20].

Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) techniques are two of the most widely used deep learning techniques, and they serve as the foundation for the majority of recent load forecasting research. The goal of this research was to compare these deep learning forecasting techniques to some conventional forecasting techniques used by various utilities to see if deep learning can meet their needs more effectively. This work was done specifically to benefit Saint John Energy, a municipally owned utility reseller that works with UNB's Smart Grid Technologies team. They are looking for better ways to improve their forecasting performance, including more accurate prediction of load demand peaks.

## 1.1 Objectives

To meet the purpose of this work, the following objectives were established:

- Identifying and implementing benchmark forecasters - Forecasters were chosen to represent both statistical and machine learning techniques, as well as the techniques most commonly used by researchers and utilities. The selection criteria were narrowed to forecasters with well-documented models, which aided in the

reproducibility of this work. The techniques are described in detail in Chapter 2, and the implementation is described in Chapter 3.

- Identifying and implementing two deep learning forecasters - Forecasters were chosen to represent those that have recently demonstrated promising results in the literature on load forecasting. The forecasters are described in Chapter 2, and the implementation details are described in Chapter 3.
- Forecaster performance comparison - Overall accuracy and daily load demand peak prediction were used as performance metrics. The findings of the overall comparison are discussed in Chapter 3.
- Extensive performance analysis - A thorough analysis was carried out to determine whether certain forecasters performed better or worse at different times of day, days of the week, months of the year, or seasons. The findings of this in-depth investigation are presented in Chapter 4.

In 2021, Saint John Energy recruited UNB's smart grid team to forecast the day and time of peak loads. None of the statistical or machine learning forecasters in use were able to predict this data accurately. This is the primary reason for our interest in deep learning research. This work helps to advance the integration of deep learning techniques into load forecasting. Our comparison shows that deep learning forecasters outperform established benchmark forecasters. This work is meant to be replicable and to serve as a model for future research by our smart-grid team and others.

## 2 Overview of Load Forecasting

This chapter provides an overview of load forecasting. It covers the various factors that influence electricity demand, forecasting horizons, benchmark and deep learning techniques, peak load definition, and performance metrics used in this work.

The primary goal of a power company is to ensure a constant supply of electricity in order to avoid blackouts. They do this in part by forecasting load demand, including peak load magnitude and timing. Load forecasting aids them in the preparation of reverse power or demand response strategies to shave or reduce the peak around the predicted time. Demand response allows power system operators and electric utilities to relieve strain on the electricity distribution system by compensating commercial, industrial, and residential customers who reduce their electricity consumption during peak demand or system emergencies.

Because load forecasting and peak load prediction are complex, the power system operator must take into account a plethora of factors including variables that will be used as inputs for forecasting, as well as the length of their forecasting horizon [21]–[26].

### 2.1 Factors That Affect the Load Demand

Load demand can be affected by a variety of factors, including the region in question, the type of customers in the region, weather factors (e.g., temperature), the time of day, the day of the week, and other uncontrollable factors (e.g., coronavirus outbreak). These variables are divided into four categories: economic, chronological, meteorological, and random.

Economic factors include investments in a facility's infrastructure, such as new buildings, laboratories, and power plants, which can affect customer consumption and increase the load on the electric grid [27]. Economic factors have little impact on short-term load forecasting because they affect consumption patterns over a longer period of time [1].

Seasonal, weekly, and daily cycles, as well as holidays, can all have an impact on the load. Weekends are comparable to public holidays, and weekdays differ from weekends in that weekends have a lighter workload. Calendar data is useful to incorporate into load forecast models because time has an effect on how electricity is used [9], [21], [22], [25], [27]–[31].

The most important weather variable is temperature, which is frequently used as a variable in forecast models [21], [23], [26], [27], [32], [33]. Temperature factors, according to Hong and Shahidehpour [34], can account for more than 70% of load variability. The relationship between temperature and load is non-linear. This nonlinear relationship helps to explain why nonlinear load forecasting techniques are so widely used [1], [21]. Humidity, solar irradiance, wind speed, and precipitation are other weather variables that may affect the electric hourly load profile [9], [23], [42], [27], [35]–[41]. Janicki [43], [44] describes the numerous meteorological variables used in load forecasting.

Any additional random disturbances in the load pattern that cannot be explained by the preceding factors are referred to as random factors affecting the electrical load profile [27], [45]. Predictions will never be perfect because unexpected events such as coronavirus outbreaks and planned or unplanned power system outages can significantly alter the demand load profile.

## 2.2 Load Forecasting Horizons

Electrical utilities' operations and planning necessitate load forecasts with varying forecast horizons based on their forecasting requirements. These horizons are classified into three types: short-term load forecasting ( $STLF < 2\text{-weeks}$ ), medium-term load forecasting ( $MTLF < 3\text{-years}$ ), and long-term load forecasting ( $LTLF > 3\text{years}$ ). Depending on the time horizon, different forecasting models and methodologies can be used, which has an impact on what is available and chosen as inputs to forecasting models [1], [46].

$STLF$  has been the focus of recent research, with an emphasis on time horizons of less than two weeks. This horizon is critical for power system operation and maintenance, as well as planning, contingency analysis, load flow assessment, and power system operation and maintenance.  $STLF$  is a multifaceted process that is influenced by a variety of factors such as economic conditions, time of day, season, weather, and human activity [1], [21], [54], [46]–[53].

$MTLF$  has a longer time horizon, typically ranging from two weeks to three years.  $MTLF$  is influenced by demographic and economic factors.  $MTLF$  and  $STLF$  are inextricably linked; long-term planning must be integrated into short-term planning [55]–[58].  $LTLF$  considers time horizons longer than three years.  $LTLF$  is also required for planning purposes, such as the construction of new power plants, transmission system expansion, and electric utility expansion [59].

## 2.3 The Benchmark Forecasters

Many publications do not provide detailed information about their experimental setups, making direct comparisons with reported results difficult. The benchmark forecasters proposed in this work were chosen for their relevance and reproducibility; they have been available for many years and have been implemented and used by both researchers and utilities [1], [3], [10], [14], [60]–[62].

Based on the forecast model's construction technique, each benchmark forecaster is classified into one of two categories: statistical techniques or machine learning techniques. Statistical techniques include multiple linear regression (MLR) analysis [63], [64] and auto-regressive integrated moving average (ARIMA) modeling [65], [66]. ANNs (Artificial Neural Networks) [67], [68] are a type of machine learning technique [69], [70].

One limitation of statistical techniques such as ARIMA and MLR is their inability to discover non-linear relationships in data without these variables being provided as inputs to their forecasting model [12], [69]–[71]. Furthermore, they are incapable of intelligently learning and adapting to data changes caused by newer factors, such as temperature swings resulting in extreme weather conditions, or a plague, such as a coronavirus outbreak that occurred in the year 2020, resulting in the global shutdown of numerous operations [32], [47], [72]–[75].

### 2.3.1 The Seasonal Naive Forecaster (SNF)

A naive forecaster is a simple forecaster that is frequently used as a benchmark to develop more sophisticated forecasters [60], [72], [73], [76], [77]. They are used to show

how much value forecasters add in comparison – when a naive forecaster outperforms a more complex forecasting model, then the complex model adds no value.

Although historical or more recent means can be used as naive forecasters, Bracale et al. [72] state that "the simplest method to anticipate the next value in a time series is to assume it will have the same values as the current value." which serves as the foundation for the most common naive forecaster. The Seasonal Naïve Forecaster (SNF) improves the naïve forecaster by taking seasonal trends into account [78]. The SNF can be expressed mathematically using the simple relationship shown in (1):

$$\hat{y}_t = x_{t-l} \quad (1)$$

where  $x$  is the time series,  $\hat{y}$  is the forecasted value,  $t$  is the time of occurrence, and  $l$  is the seasonal period ( $l=24$  for hourly data if the previous day's hourly sample is used). The naïve formula uses the most recent observed value as the future value, whereas the seasonal naïve formula uses the previous season's value. The SNF forecaster is excellent for making short-term forecasts of variables that are generally stable or consistent. It is ineffective, however, at forecasting time series data that fluctuate significantly or are subject to irregular elements, such as temperature [73].

### 2.3.2 The Multiple Linear Regression Forecaster (MLR)

Multiple linear regression (MLR) is a popular statistical technique for forecasting load that has received a lot of attention in the literature [47], [60], [64], [70], [79]–[84]. The relationships between a continuous dependent variable and one or more independent variables are modeled by MLR forecasters. Mathematically, an MLR with two independent variables is expressed as:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + e \quad (2)$$

In the case of load forecasting,  $\hat{y}$  is the predicted load,  $x_1$  and  $x_2$  are independent variables such as temperature and time-of-day,  $\beta$ s are coefficients estimated by the model, and  $e$  is an error term [64]. MLR models are fitted in such a way that the sum-of-squares of actual and forecasted value differences is minimized.

According to Amral et al. [85], MLR models for forecasting short-term load are relatively simple to develop and maintain, and the primary flaw of MLR models is their reliance on previously recorded load and temperature data, which has a significant impact on the predicted output. The accuracy of MLRs is primarily determined by the relationships between the data and the independent variables included; however, while increasing the number of relevant independent variables improves predictive accuracy in general, the improvement eventually becomes negligible. MLRs can simulate nonlinear relationships, but only when the independent variables are specified explicitly [32], [47], [72]–[75], [86].

### **2.3.3 The Auto-Regressive Integrated Moving Average Forecaster (ARIMA)**

ARIMA is arguably one of the most widely used statistical forecasting techniques, with widespread application in the load forecasting literature [3], [87]–[94]. The ARIMA model explains data by taking time-series data from previous values and predicting linear regression outcomes. It allows the application of regression techniques to non-stationary data by coercing the time-series into stationarity via differencing, which is represented by ARIMA's integrated "I" component. Because linear regression models perform better on stationary signals, differencing is required [84], [95].

Lags are important components of time series analysis that are used to uncover relationships between past and future values. The "AR" or autoregression component in ARIMA indicates that the model is dependent on the relationship between the current and previous data values (lagged values). The "MA" component, or moving average, models the forecast as a function of previous forecast errors (lagged forecast errors).

Seasonality in data can be handled using the Seasonal ARIMA (SARIMA) model, which is a more complex version of the ARIMA model. Seasonality in data is explicitly addressed in this class of ARIMA models by including seasonal AR, MA, and differencing terms in the model. External variables can also be included in the model via an exogenous regressor term. SARIMAX (seasonal ARIMA with exogenous regressors) enables the user to include the effects of external variables in the model. Exogenous variables influence but are not influenced by a model. Temperature is regarded as an exogenous variable in the context of electrical load demand [93]. The SARIMAX forecaster outperforms traditional time series methods by explicitly accounting for seasonality in the data and external variables [96]. SARIMAX's properties make it an ideal class of models for use with time-series data on electricity load demand.

SARIMAX  $(p, d, q) \times (P, D, Q)_s$  is the SARIMAX model's general form. The AR term's order is indicated by the letter 'p', while 'd' denotes the order of differencing required to make the data stationary. The order of the MA term is indicated by the letter 'q'. The seasonal term orders are denoted by the letter 'P', 'D', and 'Q'. The number of time steps in a season is denoted by 'S' ( $S = 24$  for hourly data with daily seasonality). The mathematical representation of the SARIMAX model is shown in the equation below.

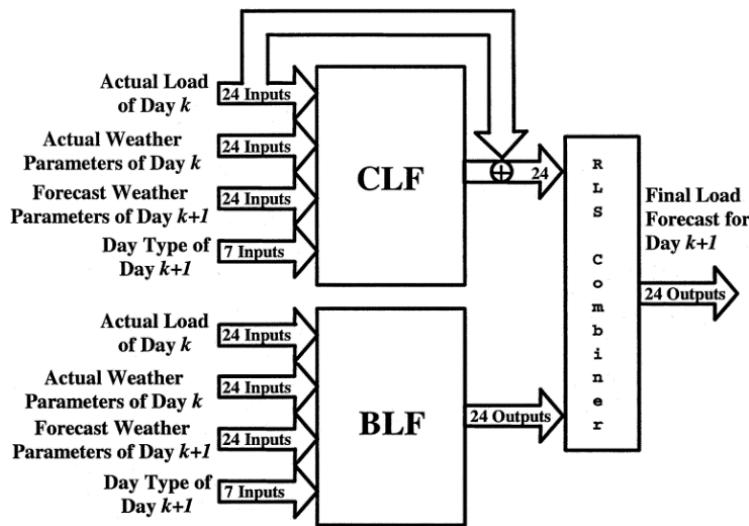
$$\hat{y}_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \dots + \beta_k X_{k,t} + \dots \\ + \frac{(1 - \theta_1 L - \theta_2 L^2 - \dots - \theta_q L^q)(1 - \Theta_1 L^S - \Theta_2 L^{2S} - \dots - \Theta_Q L^{QS})}{(1 - \phi_1 L - \phi_2 L^2 - \dots - \phi_p L^p)(1 - \Phi_1 L^S - \Phi_2 L^{2S} - \dots - \Phi_P L^{PS})} \varepsilon_t \quad (3)$$

where  $\hat{y}_t$  is the series' current value at time  $t$ .  $X_{1,t}, X_{2,t}, \dots, X_{K,t}$  denotes the exogenous variables' observations.  $\beta_0, \beta_1, \dots, \beta_k$  are the regression component's parameters. The weights of the nonseasonal autoregressive terms are denoted by  $\phi_1, \phi_2, \dots, \phi_p$ .  $\Phi_1, \Phi_2, \dots, \Phi_p$  denotes the seasonal autoregressive terms' weight.  $\theta_1, \theta_2, \dots, \theta_q$  denotes the weight of the terms in the nonseasonal moving average.  $\Theta_1, \Theta_2, \dots, \Theta_Q$  denotes the seasonal moving average terms' weight. The term  $L^s$  refers to the lag operator such that  $L^s y_t = y_{t-s}$ . The white noise terms are denoted by  $\varepsilon_t$ .

Using time series data from 2004 to 2014, Papaioannou et al. [97] forecasted national daily electricity demand in Greece. The SARIMAX model was used with weekday and temperature as exogenous variables, and it performed better than the other models in terms of forecasting unexpected increases in demand. Using data from 2003 to 2009, Felice et al. [98] forecasted electricity demand in Italy at the national and regional levels. They concluded that adding temperature as an exogenous variable to the ARIMAX model improved forecasting performance compared to SNF and ARIMA. While SARIMAX models can be accurate and reliable in the right circumstances, one of its main drawbacks is that the parameters are typically tuned manually, which can be a time-consuming process.

### 2.3.4 Artificial Neural Network Short Term Load Forecaster – Generation Three (ANNSTLF-G3)

The ANNSTLF is a popular machine-learning-based load forecaster [1], [61], [84]. The configuration of this load forecaster has undergone several revisions since it was first proposed [99], [100], and the focus of this work was the third-generation design (G3) [37] depicted in Figure 1. The ANNSTLF-G3 predicts short-term load using two shallow multi-layer feed-forward artificial neural networks (ANNs) in conjunction with a recursive least squares (RLS) combiner. The RLS is an adaptive filter algorithm that recursively finds the coefficients that minimize a weighted linear least squares error cost function related to the input signals. Additional information about the RLS algorithm is available in [101].

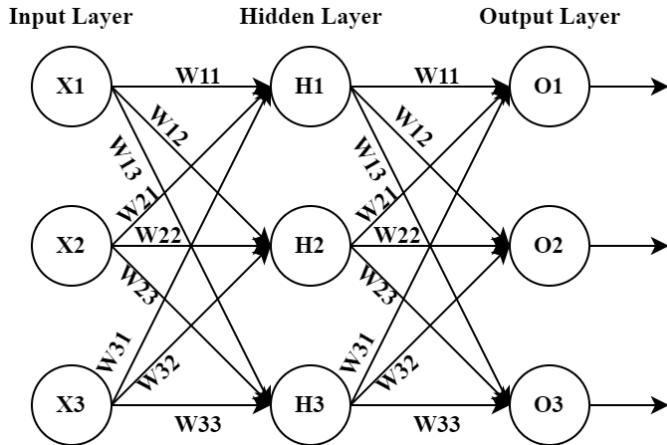


**Figure 1 - The Block Diagram of the Third Generation ANNSTLF [37]**

ANNs are neural networks that predict outputs by combining weighted inputs. The popularity of neural networks stems from their ability to uncover complex and non-linear correlations in historical data, which is more difficult to achieve using statistical techniques [14], [102]–[109]. ANNs use a learning algorithm to update weights in response to training

inputs and labelled outputs. The network, once trained, represents a prediction model that can be used with new inputs.

Figure 2 shows a simple feed-forward ANN with three inputs and three outputs. In the figure, the ANN's neurons are divided into three layers: input, hidden, and output. Each layer of the network in the example contains three neurons, and the network is fully connected, which means that each input is connected to each neuron. It is referred to as a feed-forward network because the outputs of each layer are only presented to the layers that follow it. The output of each neuron is the weighted sum of its inputs transformed by an activation function.



**Figure 2 - The Structure of a Simple Feed-forward ANN**

In the output layer, linear activation functions are typically used, whereas logistic sigmoid activation functions (often rescaled) are used in the hidden layer (for example,  $\tanh(x)$ ). These transformations are responsible for the network's nonlinearity. Backpropagation is used iteratively to update weights, ensuring that outputs eventually meet targets when presented with training inputs. When a network has been trained with

enough historical data, it can predict future load demand when a new set of inputs is introduced.

Back-propagation is used by both ANN blocks in the ANNSTLF-G3 to train two fully connected feedforward neural networks. The base-load forecaster (BLF) is trained to predict the next-day load, whereas the change-load forecaster (CLF) is trained to predict the load change from one day to the next. Because the BLF emphasizes normal load patterns and the CLF emphasizes short-term fluctuations, the two ANN forecasters complement each other. Combining these two independent forecasts improves accuracy. This is especially true when there are sudden changes in load caused by weather changes. The BLF has a slow response time to sudden changes in load. Conversely, because the CLF uses the previous day's load as a baseline and forecasts future changes in that load, it is more responsive to changing conditions [84], [110]–[112].

Both networks accept 79 inputs (as shown in Figure 1) and produce a 24x1 vector representing one day ahead, hourly forecasts. The CLF generates its final output by adding predicted changes to the previous day's actual values. The final forecast is based on a weighted average of each block's outputs, with the weights determined adaptively using an RLS algorithm. The ANNSTLF-G3 forecaster performs best when the hidden layer contains between 30 and 60 fully connected neurons and the model is trained using at least two to three years of data, according to the authors [37].

In terms of prediction accuracy and economic benefits, over thirty-five utilities in the United States and Canada have benefited from the ANNSTLF-G3; some of these utilities include BC Hydro, ISO New England, and San Diego Gas & Electric [86], [113]. In 2016,

Tao Hong [1] and several other publications had named the ANNSTLF-G3 the best forecaster available for short-term load forecasting [61], [84], [114]–[118].

In general, ANNs have been extensively studied in their application to load forecasting. In [119] and [87], Papalexopoulos et al. developed a neural network-based and regression-based technique. In 1991, both models were validated using training data from 1986 to 1990 on peak and hourly loads. For both peak load and hourly forecasts, the ANN model improved forecasting accuracy. Zhang et al. [120] evaluated and demonstrated the utility of neural networks in load forecasting but noted that while neural networks are capable of processing large amounts of historical load data with non-linear characteristics, they are a black box technique that lacks an explicit form for explaining and analyzing the relationships between inputs and outputs.

Finally, it is worth noting that a shallow ANN typically has only one hidden layer and increasing the number of neurons in that layer is insufficient to produce more accurate forecasts; the network becomes overtrained, impairing its ability to work with new datasets. Thus, more sophisticated neural networks with deeper networks of hidden layers are attracting the attention of researchers interested in improving the accuracy of load forecasters.

## 2.4 Deep Learning Forecasters

Deep learning is a process in which the number of hidden layers in a network is increased [121]–[127]. While they can provide added benefits such as including memory, or automated feature identification, they are computationally expensive, and sometimes require large data sets to be useful. Deep learning models have transformed computer

vision, speech recognition, machine translation, and board game programming, producing results comparable to, if not superior to, expert human performance [128], [129]. Deep learning models are expected to dominate the field of load forecasting due to increased computational power, access to large datasets, and the granularity of available data [2], [6].

To identify the most reliable features in machine learning, a domain expert is required. While deep learning models can be fed some input features, they can also extract high-level features from data incrementally, eliminating the need for domain expertise and time-consuming feature extraction. Machine learning forecasters may be preferable when the data set is small, but deep learning is preferable when the data set is large, feature introspection is burdensome, or the problem is complex.

Deep learning is a subclass of neural networks that includes a wide variety of architectures. The most common types of deep neural networks are convolutional neural networks (CNN) [2], [7], and recurrent neural networks (RNN) [8], which include networks such as the long short-term memory network (LSTM) [5]. Deep neural networks, such as CNN and LSTM, excel at detecting highly nonlinear relationships and shared uncertainties in data; the LSTM, in particular, introduces memory into neural networks, allowing them to predict today based on observations from yesterday or two weeks ago, making them ideal candidates for load forecasting.

Because of their ability to learn about temporal dependencies in data and rapidly adapt to sudden changes in load patterns, deep neural networks have piqued the interest of load forecasting researchers. While not exhaustively researched in the literature, at least a few recent studies have shown promising results of deep neural networks applied to load forecasting [74], [127], [130].

In recent years, load forecasting researchers have started to explore CNNs and LSTMs [2], [4], [20], [131]–[133]. The authors of [134] investigated seven distinct models using three real-world data sets, demonstrating that deep learning techniques such as CNN and LSTM can be used in load forecasting applications and outperform more traditional mathematical techniques such as ARIMA. The authors of [4] proposed a novel parallel model that combines CNN and RNN. Likewise, the authors of [135] used a combination of LSTM and CNN. In terms of load forecasting stability, their proposed model outperformed the individual CNN and LSTM models.

Similarly, the authors of [68] proposed a new Deep-Energy model for forecasting future load data [136], which combines a 1-D CNN with a fully connected network. They compared the proposed model's performance to that of five different machine and deep learning techniques, including LSTM and ANN. The results showed that the Deep-Energy model could make more accurate short-term load predictions than the other models.

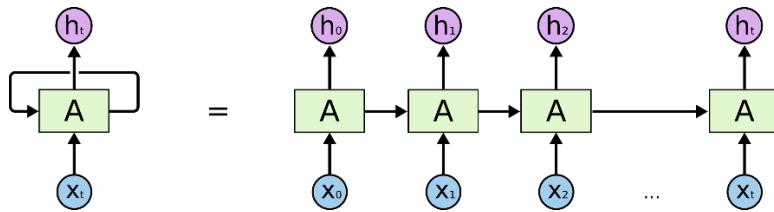
Another paper [137] introduced SEPNet, a new model that combines three forecasters: Variational Mode Decomposition (VMD), Convolutional Neural Networks (CNN), and Gated Neural Networks (GRU). When compared to other models such as LSTM, CNN, and VMD-CNN, the SEPNet model outperformed them all, significantly improving the overall performance of the forecasts. Amaradinghe et al. [2] concluded that CNN is a viable technique for individual building load forecasting when compared to LSTM, SVM, ANN, and other forecasters they implemented.

Because the ANNSTLF has consistently been recognized as a strong forecaster for short-term load forecasting, the CNN and LSTM implementations used in this work were used to simply augment the ANNSTLF architecture by replacing ANNs with CNNs and

LSTMs. The goal was to improve performance through better focused inputs, and they were given the same input data as the ANNs in the ANNSTLF architecture. The following subsections provide a brief explanation of how LSTMs and CNNs work.

#### 2.4.1 Recurrent Neural Networks and the Long Short Term Memory Network

The recurrent neural network (RNN) is a neural network model for time series analysis that was first proposed in the 1980s. In a traditional neural network, all inputs and outputs are assumed to be independent. RNNs do not assume such independence and rely on previous elements to influence current elements, making them an obvious choice for processing time-series data. RNNs, in effect, augment neural networks with memory, which aids in the modeling of sequential data. RNNs have been used successfully for machine translation, speech synthesis, and time series prediction [138]. The authors of [139] thoroughly investigated these networks.



**Figure 3 - Unrolled Recurrent Neural Network (RNN)** [140]

RNNs, like regular ANNs, predict outputs using a combination of weighted inputs, but the inputs are elements in a data sequence and also include a weighted previous state, as shown in Figure 3. Using the previous state as an input adds a hidden layer for each state. Because RNNs are typically trained using back-propagation, the gradient descent must traverse each layer to update the state weights, causing vanishing gradient issues and

limiting their effectiveness when dealing with large data sets or problems requiring long-term memory.

The long short-term memory network (LSTM) is a type of RNN that solves the vanishing gradient problem, allowing these networks to predict states over long time periods. The LSTM memory cell configuration outperforms any other deep neural network configuration currently available [6], [7], [107]. The LSTM architecture is made up of memory blocks, which are recurrently connected subnetworks (or cells). As shown in Figure 4, each memory block consists of a cell state, a forget gate, an input gate, and an output gate.

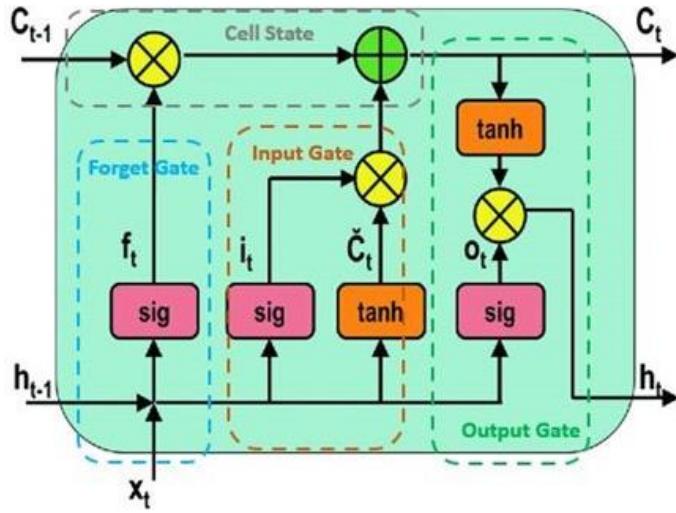


Figure 4 - The Block of Long-Term Short-Term Memory [141]

The cell state, represented by the operations on the top left of the figure, is essential for LSTMs to function. The three gates guard and regulate the cell state [7]. On an opt-in basis, gates allow information to pass through. They are constructed using sigmoid neural networks and pointwise multiplication. The sigmoid layer generates values between 0 and 1, indicating how much of each element should be allowed to pass through.

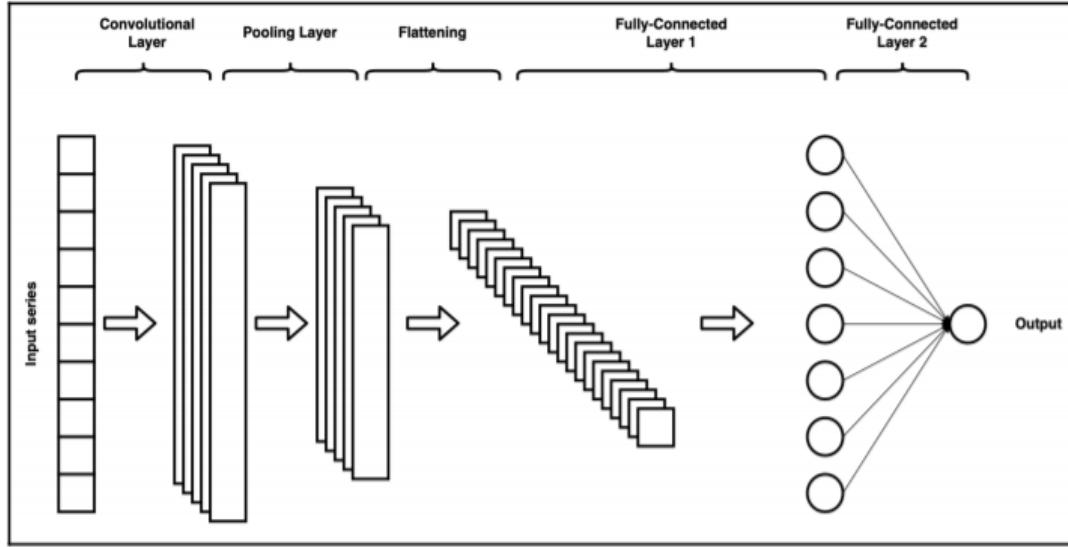
The first stage in LSTM is a sigmoid layer known as the "Forget Gate", used to decide what information from the previous cell state should be discarded. It examines the hidden layer's previous state and inputs to generate a multiplier between 0 and 1 for each value in the cell state. The previous cell state is effectively filtered as a result of this. The next step is to filter the outputs of a tanh layer used to update inputs to determine what new information will be stored in the cell state, which happens in the "Input Gate." The newly acquired information is then incorporated into the cell state. Finally, the network employs an "Output Gate" to generate outputs in the form of a filtered version of an updated cell state [142], [143]. Because of the method of forgetting and remembering information within a cell, LSTM is ideal for sequential data.

#### **2.4.2 The Convolutional Neural Network (CNN)**

CNNs have been successfully applied in a variety of applications, including computer vision, audio processing, activity recognition, natural language processing, drug discovery, video recognition, and time series forecasting [2], [7], [148]–[152], [88], [121], [134], [138], [144]–[147]. The ImageNet Large Scale Visual Recognition Competition (ILSVRC) is an annual international computer vision competition, and a CNN won it for the first time in 2012 [153]–[161]. CNNs are a type of deep learning network with a grid-like topology that can process time series and image data, which are represented as one-dimensional and two-dimensional implementations, respectively. Because load forecasting data are time series, the focus of this work was on one-dimensional CNNs.

In general, CNNs are used to extract rich feature sets from inputs. There has been some recent work focused on raw time-series data as inputs[162]–[164] (e.g., the advancement

of TCNs), but the focus of this work was on the use of CNNs in the more traditional sense, to transform a feature set, in order to improve a forecaster's performance. The architecture of a simple one-dimensional CNN used in this way is depicted in Figure 5.



**Figure 5 - A Simple One-Dimensional CNN's Architecture** [165]

A CNN's network architecture is constructed in stages. The first stage is made up of one or more convolutional layers and pooling layers. The second stage consists of a flattening layer which transforms the pooling layers' outputs into usable input for the next stage. The final stage is made up of fully connected layers used to produce final output(s) in the conventional way a feed-forward network does.

The convolutional stage is the most important stage in the CNN architecture. It is a pattern finder; it searches for patterns in the input data by applying filters (also known as kernels) repeatedly along the input data. Assume from the figure above that it depicts a 10x1 input array. In a load forecasting context, these inputs could represent 5 previous day loads and 5 predicted day temperature forecasts. In this context, the filter is a one-

dimensional weight vector that is convolved with the input vector. That is, it slides along the length of the inputs storing the dot product at each step. The result is a feature map made up of dot products whose length depends on the length of the filter (unless padding is utilized).

For example, a  $5 \times 1$  filter applied to the  $10 \times 1$  input vector would yield a  $6 \times 1$  feature map (assuming a stride length of 1 and no padding). More than 1 filter can be applied in this way to deepen the convolutional layer, and many stacked layers can deepen the network even further to yield more refined pattern detection. The application of filters to patches of inputs effectively identifies local patterns in the inputs and applying the same filter along the length of the inputs allows for such patterns to be recognized across the set of inputs. Weights in the filters are updated using backpropagation to gain a better representation of patterns in relation to what is being modeled or predicted.

Following the creation of feature maps in the convolutional layer, the elements contained in the feature maps are activated using an activation function, just like in a conventional neural network. The most common activation function used in a CNN is the rectified linear unit (ReLU). The ReLU activation function is a piecewise activation function that produces linear outputs for positive values but 0 for negative values. When the ReLU activation function is used, the gradients remain proportional to node activations, which helps to avoid vanishing gradients problems.

A pooling layer generally follows a convolutional layer, to reduce the spatial dimensions of a feature map, and also decrease localization sensitivity. Calculating the mean or taking the maximum across a set of elements in the feature map is typical. A single value is generated from the specified pool of neighboring elements on the same feature map. The

pooling layer reduces the complexity of the CNN, reducing computational load during training and also the possibility of over-fitting [121], [138], because the feature map is less sensitive to small translations that could be present in the original features.

The flattening layer comes after the pooling layers and transforms the outputs of the pooling layers into a one-dimensional array, which is then used as an input to the fully connected layer that follows. In the context of load forecasting, the fully connected layers are trained with backpropagation to produce forecasted values. The fully-connected layer of a CNN has the same structure as a fully-connected feed-forward ANN layer.

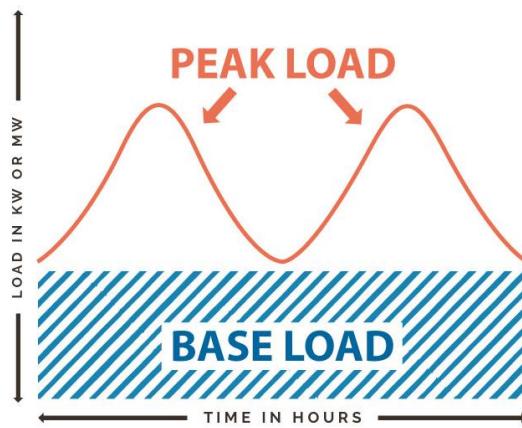
## 2.5 Peak Load

When ensuring that customers or electricity consumers have a stable supply of electricity, two load measures must be considered: base and peak load. Consider the electrical requirements of a house. The amount of electricity constantly required from the electrical grid is referred to as the base load. A peak load occurs when additional power is required, such as when an entire family is at home watching television and consuming a large amount of electricity. It is a momentary period of increased demand, as the family will soon be sleeping, turning off the television and lights, and conserving energy.

The term "base load" refers to the absolute minimum amount of electrical demand that must be met over the course of a 24-hour period. Base load requirements are also referred to as constant load requirements because they are relatively constant. Conversely, peak load refers to the maximum amount of energy drawn from the grid by a consumer over a specified period. Most utilities operate on a 15-minute cycle, but some operate on 30- or 60-minute cycles; the cycles refer to the minimum amount of time the demand must persist

in order to be identified as a peak load; if the demand does not persist for at least that cycle, it is declared a random spike [166]. Typically, demand spikes are caused by the activation of an electrically consuming device and last only a few seconds.

The visual distinction between the base and peak loads is depicted in Figure 6. The base load is more stable, but also less intense, because electricity is still required for things like heating, cooling, and power outlets, among other things. Peak load is less predictable than base load; it can surge when air conditioners are turned on or when a snowstorm hits and the heat must be turned up [167]. Peak load forecasting is critical for ensuring adequate generation, transmission, and distribution capacity.



**Figure 6 – Peak Load vs Base Load** [168]

Peaks have three main characteristics: magnitude, temporal location, and width or duration, with the temporal location being the most important. Knowing when a peak will occur allows utilities to plan reserve power and demand response strategies to help reduce the peak, resulting in significant savings for both the utility and its customers; as such, understanding peak load is critical for any business's energy management strategy [169]. Electricity is generally more expensive during peak hours, and the peak load determines

the rate that the utility charges the customer. In this study, forecasters were evaluated based on their accuracy in predicting daily peaks, considering both the peak's value and the time of occurrence.

## 2.6 Performance Metrics

Several performance metrics have been used in the load forecasting literature to evaluate forecast accuracy, each with its own set of advantages and disadvantages, but despite their limitations, they are simple tools for evaluating forecast accuracy. While the mean absolute error (MAE) is the simplest way to quantify forecast errors, it cannot be used to compare measurements across forecast scenarios with different scales because it is an absolute measure; therefore, the mean absolute percent error (MAPE) is the most commonly used load forecasting accuracy metric because it is easier to interpret [1], [9], [14], [135], [136], [170]. The table below summarizes some of the most commonly used metrics for determining load forecasting accuracy:

$MAE = \frac{1}{n} \sum_{i=1}^n  forecasts - actuals $	$MAPE = \frac{100}{n} \sum_{i=1}^n \left  \frac{forecasts - actuals}{actuals} \right $
$MBE = \frac{1}{n} \sum_{i=1}^n (forecasts - actuals)$	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (forecasts - actuals)^2}$

**Table 1 - Formulas for Several Frequently Used Performance Metrics**

The MAPE has limitations in that it cannot handle zero-valued actuals, it over-emphasizes high errors during low demand periods, and it over-emphasizes overshoot errors in comparison to undershoot errors for forecasting scenarios bounded by 0 (because undershoot errors cannot exceed 100 percent , but overshoot errors are unbounded) [1], [171]. However, MAE and MAPE are both insensitive to rare but significant errors, which

are better captured by the root mean square error (RMSE), though the RMSE is more difficult to interpret because it is not scaled to the original error [68]. To fully capture bias and precision, mean biased error (MBE) and standard deviation (STD) can also be used [172], [173].

## **2.7 The Myth of Finding the One Size Fits All Technique**

Tao Hong talked about the myth of discovering the best technique [1]. He concluded that researchers and users must understand that there is no technique that is universally superior. The method used for load forecasting should be determined by the forecasting requirements and the dataset under consideration. One approach is unlikely to be beneficial in all load forecasting scenarios. Different forecasters perform better or worse depending on the nature of the dataset. Furthermore, forecast errors differ greatly across utilities, utility zones, and time periods.

### **3 Investigation**

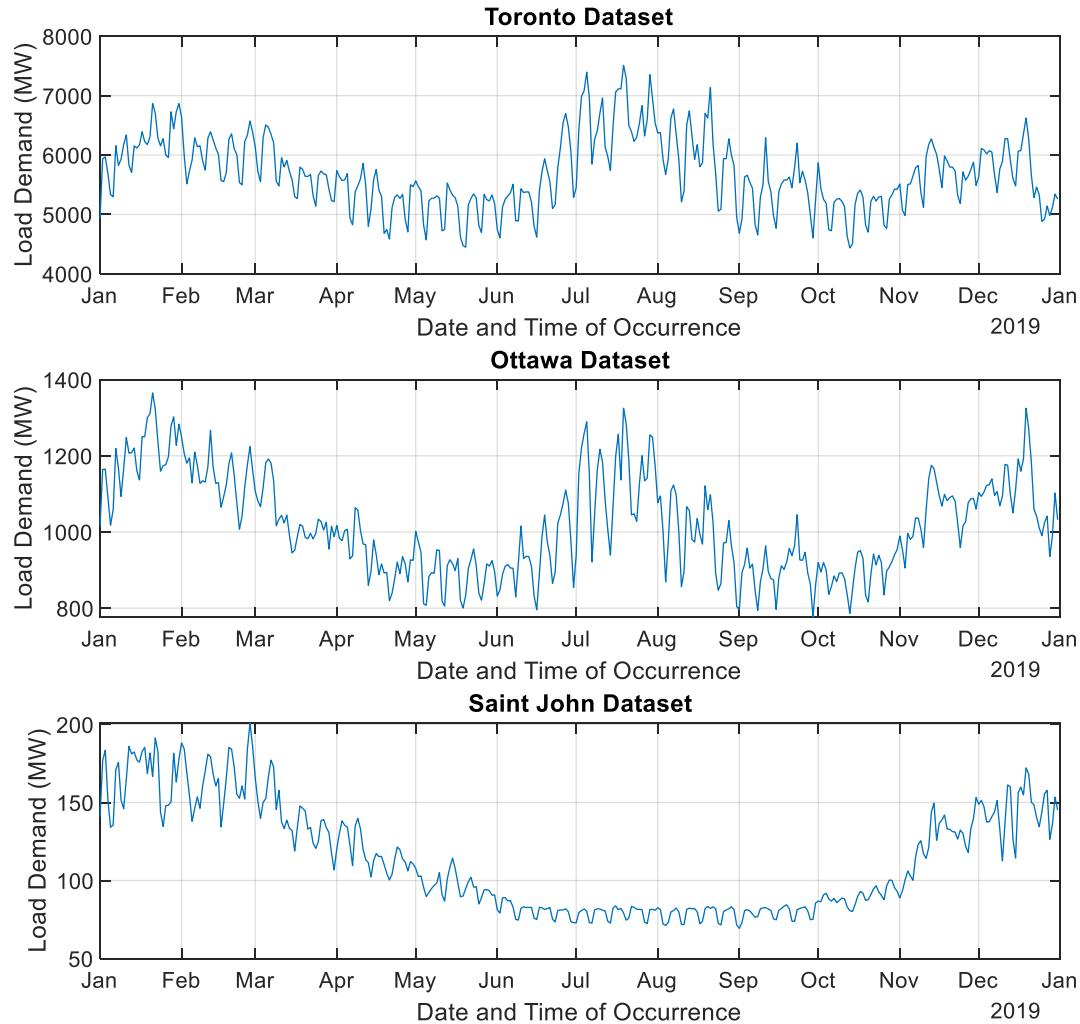
The purpose of this work was to determine whether deep learning techniques could improve forecasting accuracy for specific utilities by comparing the accuracy of deep learning forecasters to some of the conventional forecasters used by utilities. The focus of this work was on STLF horizons, specifically one day ahead forecasts, because they are an important component of a utility system's day-to-day operations and planning. A CNN forecaster and an LSTM forecaster were compared to four benchmark forecasters: a Seasonal Naive forecaster, a Multiple Linear Regression (MLR) forecaster, a Seasonal Auto-Regressive Integrated Moving Averages with Exogenous Regressors (SARIMAX) forecaster, and a shallow Artificial Neural Network forecaster (ANN). The forecasters' accuracy was assessed based on their ability to predict regular loads and daily peaks.

#### **3.1 Preparation of the Datasets**

This study used three distinct datasets. For visual clarity, Figure 7 depicts the daily mean for each data set in 2019. Two sets were obtained from an Independent Electrical System Operator in Ontario and were included because the data sets are publicly available, which aids in the reproducibility of this work. The first set was from Ottawa [174], and the second was from Toronto [174], and they both consist of hourly city-wide load aggregation measurements spanning ten years from 2010 to 2019.

Saint John Energy, a municipally owned utility reseller, provided the third set. This data set was included because the work is part of a larger Smart Grid Technologies project being carried out at UNB in collaboration with this utility reseller. Saint John Energy's data set is smaller than the others, spanning about 3.75 years from 2018 to October 20, 2021, but it

otherwise corresponds to hourly measurements of Saint John's load aggregates. For all datasets, all load demand variables are expressed in megawatts (MW). Temperature data from Environment Canada were used to supplement the time series data, which consisted of each city's hourly average temperature expressed in degrees Celsius [175].



**Figure 7 – 2019 Average Daily Demand for Loads Across All Datasets**

A Hampel filter was used to find and replace outliers in the datasets for both the load and temperature variables; outliers were defined as test sample values that differed by more

than three standard deviations from the median; a seven-sample window (length = 7 hours) was used, with the sample under test at the center [176].

The Min-Max method was used to normalize all data (load and temperature), which scales values between zero and one by using the time series' minimum and maximum values, as specified in the equation below:

$$\text{Normalized Value} = \frac{\text{Actual Value} - \text{Minimum Value}}{\text{Maximum Value} - \text{Minimum Value}} \quad (4)$$

Before any performance metrics were calculated, the minimum and maximum values were saved so they could be used to de-normalize the final forecasts. This normalization technique has been used by numerous researchers in the field of load forecasting [25], [177]–[181].

Table 2 shows how the data was divided into training and test sets. As detailed in the sections that follow, depending on the needs of a forecaster, training data was used to either fit a forecasting model, train a network in a forecaster, or pad input exemplars. For all forecasters, evaluation was performed on the test data.

	<b>Toronto</b>	<b>Ottawa</b>	<b>Saint John</b>
<b>Training</b>	2010/01 – 2018/12	2010/01 – 2018/12	2018/01/01 – 2020/10/20
<b>Testing</b>	2019/01 – 2019/12	2019/01 – 2019/12	2020/10/21 – 2021/10/20

**Table 2 – The Training and Testing Dataset Ranges Across All Datasets**

### **3.2 Implementation Specifications for Benchmark Forecasters**

MATLAB version R2021b was used to implement all forecasters. The forecasters generated hourly forecasts day by day, based on actual historical data from previous days. The temperature variables used in this study were all actual temperature values for both previous and forecasted temperatures. When simulations are run in real time and the actual

temperature values for the predicted day are unavailable, forecasted temperature values can be used, but this was not the case in this study.

### 3.2.1 The Seasonal Naïve Forecaster (SNF)

The seasonal naive forecaster was simple to implement; the forecasted value  $\hat{y}_t$  was calculated using an actual load lag value  $x_{t-l}$  for lag  $l = 168$  hours (1 week):

$$\hat{y}_t = x_{t-l \mid l=168} \quad (5)$$

The lag value of 168 was chosen because the load forecasting datasets showed significant seasonality between weekly lag values, which was also true for all days of the week, including weekends. No model fitting or training was required. This procedure was repeated for each hour in our test dataset to forecast a load value.

### 3.2.2 The Multiple Linear Regression Forecaster (MLR)

The MLR forecaster was implemented with ten independent variables (inputs), which are listed in Table 3. As shown in Equation 6, the model estimated a total of 56 coefficients, which included a constant, a linear term for each independent variable, and products of distinct independent variable pairs (no squared terms):

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_9 x_9 + \beta_{10} x_{10} + \beta_{11} x_1 x_2 + \beta_{12} x_1 x_3 + \dots + \beta_{55} x_8 x_{10} + \beta_{56} x_9 x_{10} + e \quad (6)$$

Here  $\hat{y}$  is the predicted load,  $x$  are the independent variables,  $\beta$  are coefficients estimated by the model, and  $e$  is an error term. The ordinary least squares algorithm [182] was used to fit the model to the training data. After the model was fully specified, it was used to forecast a value for each hour in the test set.

Variable Name	Variable Description
Actual Temperature ( $x_1$ )	
Hour of the Day ( $x_2$ )	1 ... 24
The month of the Year ( $x_3$ )	1 ... 12
Day of the Week ( $x_4$ )	Sunday is 1, ..., Saturday is 7
Weekend Indicator ( $x_5$ )	0 or 1
Maximum Hourly Demand from the Previous Day ( $x_6$ )	Load Demand
Minimum Hourly Demand from the Previous Day ( $x_7$ )	Load Demand
Average Hourly Demand from the Previous Day ( $x_8$ )	Average Load Demand
Load Demand Lag Value for Lag = 24 Hours (1 Day) ( $x_9$ )	Load Demand
Load Demand Lag Value for Lag = 168 Hours (1 Week) ( $x_{10}$ )	Load Demand

**Table 3 - The MLR Forecaster's Independent Variables**

### 3.2.3 The Seasonal Auto-Regressive Integrated Moving Averages with Exogenous Regressors Forecaster (SARIMAX)

The SARIMAX model was used in this study, which is a modified version of the ARIMA model that takes seasonality into account and includes the actual temperature as an exogenous variable. The equation below mathematically represents the SARIMAX model; it is repeated here from Chapter 2 for convenience.

$$\hat{y}_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \dots + \beta_k X_{k,t} + \dots \\ + \frac{(1 - \theta_1 L - \theta_2 L^2 - \dots - \theta_q L^q)(1 - \Theta_1 L^S - \Theta_2 L^{2S} - \dots - \Theta_Q L^{QS})}{(1 - \phi_1 L - \phi_2 L^2 - \dots - \phi_p L^p)(1 - \Phi_1 L^S - \Phi_2 L^{2S} - \dots - \Phi_P L^{PS})} \varepsilon_t \quad (7)$$

Dataset	(p, d, q) x (P, D, Q)s	Exact Lag Vectors - (p, d, q) x (P, D, Q)
Toronto	(1, 1, 2) x (7, 1, 7) <sub>24</sub>	(1, 1, [1, 2]) x ([24, 48, 72, 168], 24, [24, 168])
Ottawa	(1, 1, 1) x (2, 1, 7) <sub>24</sub>	(1, 1, 1) x ([24, 48], 24, [24, 168])
Saint John	(2, 1, 1) x (7, 1, 2) <sub>24</sub>	([1, 2], 1, 1) x ([24, 168], 24, [24, 48])

**Table 4 - The SARIMAX hyperparameters that were used for each dataset**

A trial-and-error approach guided by autocorrelation (AC) and partial autocorrelation (PAC) plots was used to specify the SARIMAX forecaster's hyperparameters. Table 4

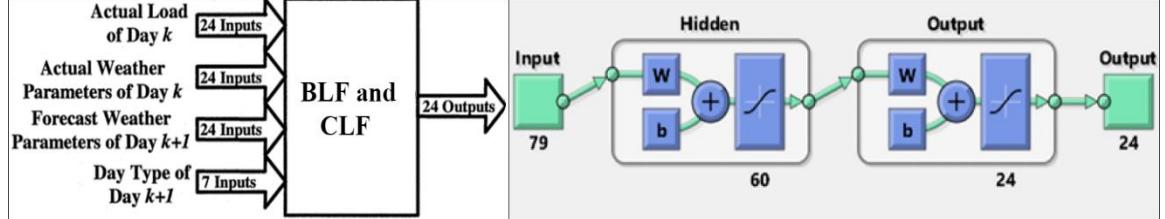
summarizes the parameters for each dataset, and Appendix A contains the AC and PAC plots that were used to help justify these decisions.

For each forecasted day, a model was fit using actual demand and actual temperature lag values from the 28 days preceding, yielding a unique model for each day. Fitting was limited to 28 days to avoid longer duration seasonality. For the first 28 days in the test dataset, parts of the training dataset were used to pad the data to fit the model.

To fit the model, the expectation-maximization algorithm was used, assuming a student's t distribution (which improved performance over a Gaussian distribution, based on preliminary trials). Once the model was fully specified, it was used to forecast hourly values for the forecasted day, with the next day's 24-hour actual temperature acting as an exogenous variable. This process was repeated for each day in the test set to produce a forecast for each hour.

### **3.2.4 The Artificial Neural Network Short Term Load Forecaster (ANNSTLF-G3)**

Figure 8 depicts the architecture of the BLF and CLF components of the ANNSTLF-G3 implementation. Both ANNs were fully connected across all layers, with sixty neurons in the hidden layer. In both the hidden and output layers, the activation function was a hyperbolic tangent sigmoid transfer function. Both networks were trained with resilient backpropagation [183]. To avoid overtraining, 80% of the training data was used to train the ANNs, while 20% was used for validation. The data was randomly divided for the training and validation sets.



**Figure 8 – The Structure of the BLF and CLF Network**

Both the BLF and the CLF were presented with 79 inputs, as shown in the figure above, where  $k+1$  represents the day to be predicted and  $k$  represents the previous day. The BLF was trained to generate load demands for each hour of day  $k+1$ , yielding 24 outputs. The CLF also yielded 24 outputs but was trained to generate hourly changes in load demand from day  $k$  to day  $k+1$ . As such, during training, the BLF was provided with actual load demand for day  $k+1$  targets, whereas the CLF was presented with the difference in actual loads from day  $k+1$  to day  $k$ .

Each ANN's output was fed into the RLS combiner, to provide a fine-tuned hourly load prediction, but before presenting the CLF outputs, they were added to actual loads from day  $k$  to reflect a load prediction rather than a change in load. The RLS combiner was initially configured to combine the outputs of both ANNs equally (i.e., weights were set to 0.5), and after the predicted day had passed, the weights for each hour were updated using a least-squares algorithm based on how large the difference between predicted outputs and actual values was. This process was repeated for each day in the test set to produce a forecast for each hour.

### 3.3 Implementation Specifications for the Deep Learning Forecasters

#### 3.3.1 The Long Short Term Memory Forecaster (LSTM)

The ANNSTLF structure was modified by replacing the ANNs in the BLF and CLF with LSTM networks, but the architecture's inputs and structure remained the same. Both the BLF and CLF LSTM blocks had four layers: a sequence input layer with 79 inputs, an LSTM layer with 100 hidden units, a fully connected layer with 24 outputs, and a regression layer. The number of hidden units was finally set to 100 after some trial and error.

The entire training set was used to train both LSTM networks using the Adam optimization algorithm, which is assumed to optimize a mini-batch backpropagation through time [184]–[186]. The Adam training algorithm used the training options specified in Table 5 (all unspecified options were set to default values in MATLAB R2021b):

Training Options	LSTMs
Initial Learning Rate	0.005
First Moment Decay Rate ( $\beta_1$ )	0.9
Second Moment Decay Rate ( $\beta_2$ )	0.999
Mini Batch Size	24
Shuffle	Every Epoch

**Table 5 – LSTM Networks Training Options**

Once trained, the LSTM networks produced 24 outputs representing 1-day ahead hourly forecasts, just like the standard ANNSTLF-G3. The final results of the two LSTM networks were then combined using the RLS combiner. This process was repeated for each day in the test set to generate a forecast for each hour.

### 3.3.2 The Convolutional Neural Network Forecaster (CNN)

CNN networks were implemented for use in the ANNSTLF architecture in a manner similar to that of LSTM networks. Both BLF and CLF CNNs had the following layers: an input layer with 79 inputs; a convolutional layer with 15, 6x1 filters; a rectified linear unit activation layer (ReLU); a max-pooling layer with a 2x1 pool size; a fully connected layer with 24 outputs; and a regression output layer.

The CNN networks, like the LSTM networks, were trained with the entire training set using the Adam optimization algorithm with the training options specified in Table 6 (all unspecified options were set to default values in MATLAB R2021b):

Training Options	CNNs
Initial Learning Rate	0.001
First Moment Decay Rate ( $\beta_1$ )	0.9
Second Moment Decay Rate ( $\beta_2$ )	0.999
Mini Batch Size	128
Shuffle	Every Epoch

**Table 6 – CNN Networks Training Options**

After training, the CNN networks generated 24 outputs representing 1-day ahead hourly forecasts, just like the standard ANNSTLF-G3. The final results of the two CNN networks were then combined using the RLS combiner. This process was repeated for each day in the test set to generate a forecast for each hour.

### 3.4 Performance Analysis

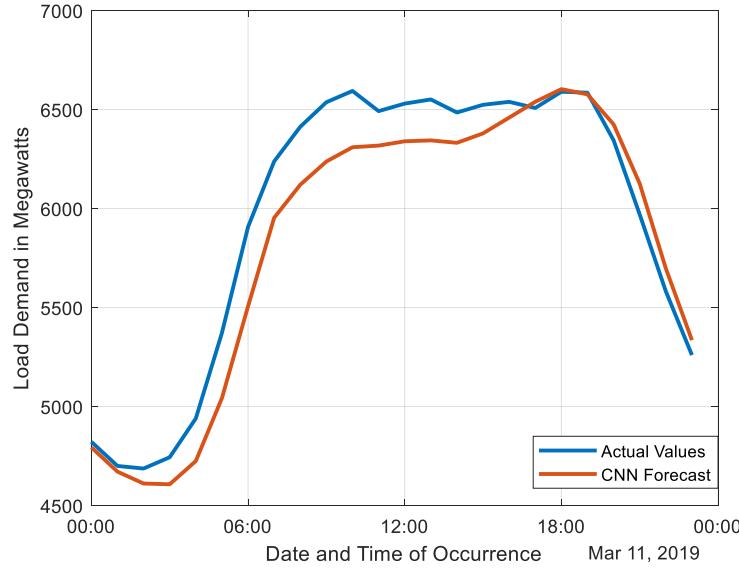
The goal of this study was to forecast 1-day ahead loads and identify daily peaks. MAPE and RMSE were used to evaluate forecaster performance in regular load forecasts. Because there are no values near zero in the datasets, MAPE limitations do not apply, and the RMSE

allowed for detection of significant forecast errors. For breadth, Appendix B contains information on the overall performance of all forecasters using all of the performance metrics mentioned in Chapter 2.

MAPE, MAE, and MBE were used to evaluate forecaster performance in predicting and identifying daily peaks. MAPE was used to quantify the accuracy of the forecasted peak magnitudes by calculating the difference between the forecast and actual peak values. MBE and MAE were used to calculate the time difference between occurrence of actual and forecasted peaks. MAE was used to determine the accuracy of the time differences, whereas the MBE was used to determine the models' overall bias, or whether they over or under forecasted based on the forecasted time of occurrence. When reported, MAE and MBE are denoted in minutes.

### **3.4.1 A Remark on Peak Detection Accuracy Metrics**

It is important to note that daily peaks are influenced by a variety of random variables, making them difficult to predict. This is because random peaks can produce significantly higher peaks than regular peaks, and because we are calculating the daily maximum, we use the random peak rather than the regular peak. We only mentioned peaks because our datasets, which are hourly aggregated load demand, do not contain any spikes. As a result, the daily maximum represents the day's actual peak. Figure 9 and the paragraph that follows provide concise summaries of the preceding statements.



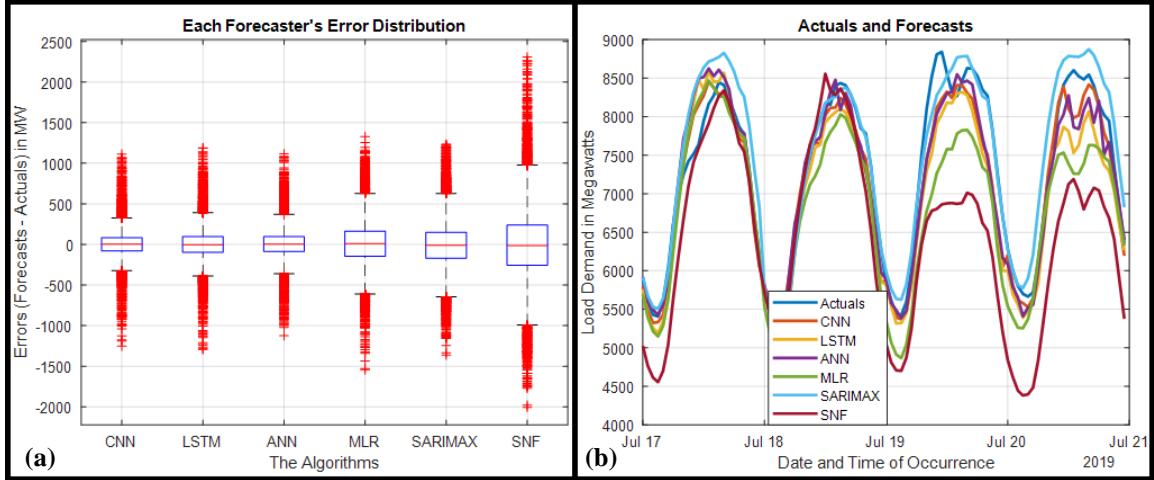
**Figure 9 - Load Demand on March 11, 2019, and CNN Forecast – Toronto Dataset**

Although the Toronto dataset typically peaks between 16:00 and 21:00 in the evenings, a random peak with a value of 6594 MW occurred at 10:00, which was greater than the second-highest peak with a value of 6590 MW at 18:00. Although it was only 4 MW more, because we used the daily maximum, we had to use the peak at 10:00. However, CNN predicted a peak at 18:00 with a value of 6603 MW. Nonetheless, because we will be comparing the predicted time to the one at 10:00, the random peak has an impact on our MAE and MBE accuracy metrics for the time difference. This is just one factor to consider when it comes to forecasters' ability to predict daily peaks. A significantly more accurate metric for comparing time differences could be used in future work.

### 3.5 The Performance of Forecasters on the Toronto Dataset

Table 7 summarizes the key performance metrics across the test dataset, while Figure 10a depicts the overall distribution of errors for each forecaster across the test dataset. Table 8 summarizes the forecaster's performance in predicting daily peaks. Figure 10b

depicts a snapshot of actual and forecasted load demand for the period of July 17th to July 21st; this period was chosen for visualization because it coincided with the month in which all forecasters performed the worst overall.



**Figure 10: (a) Overall Error Distribution for All Forecasters; (b) Actual and Forecasted Load Demand for July 17th-21st - Toronto Dataset**

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
<b>MAPE (%)</b>	2.16	2.54	2.30	3.75	4.00	6.09
<b>RMSE (MW)</b>	189.76	219.57	201.32	293.94	321.58	488.07

**Table 7 - Overall MAPE and RMSE for Each Forecaster – Toronto Dataset**

The data in Table 7 show how the forecasters fared overall across the Toronto test dataset. The MAPE and RMSE values for the CNN were the lowest, followed by the ANN and LSTM. Similarly, looking at the plot in Figure 10a, we can see that the CNN forecaster had the tightest error distribution of all forecasters. The SNF forecaster produced the worst results, with the widest error distribution and the worst global metrics.

According to the MAPE values in Table 8, the CNN was the most accurate at predicting the magnitude of daily peaks, followed by the ANN and LSTM. According to the MAE values, the CNN predicted the time of occurrence of the daily peaks the most accurately,

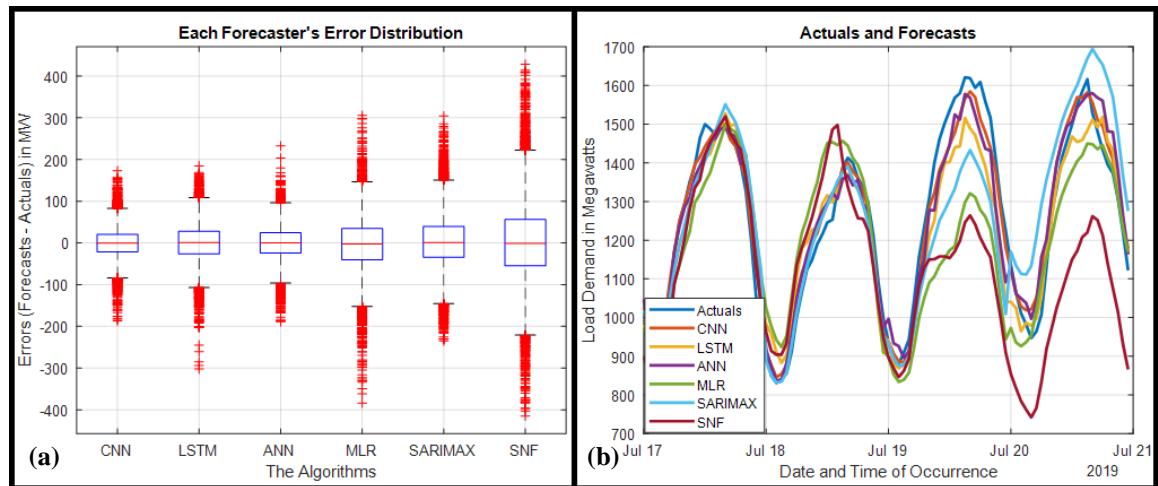
followed by the LSTM. According to the MBE values, the SNF and LSTM were the most accurate in terms of bias.

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
<b>MAPE - Peak Values</b>	2.36	2.76	2.45	3.65	4.41	6.18
<b>MAE - Time in Minutes</b>	80	90	93	96	93	92
<b>MBE - Time in Minutes</b>	22	6	-9	36	17	0

**Table 8 - Matrix Analysis of Peak Values and Time Difference – Toronto Dataset**

### 3.6 The Performance of Forecasters on the Ottawa Dataset

Table 9 summarizes the key performance metrics across the test dataset, while Figure 11a depicts the overall distribution of errors for each forecaster across the test dataset. Table 10 summarizes the forecaster's performance in predicting daily peaks. Figure 11b depicts a snapshot of actual and forecasted load demand for the period of July 17th to July 21st; this period was chosen for visualization because it coincided with the month in which all forecasters performed the worst overall.



**Figure 11: (a) Overall Error Distribution for All Forecasters; (b) Actual and Forecasted Load Demand for July 17th-21st - Ottawa Dataset**

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	2.72	3.44	3.09	4.78	4.98	7.33
RMSE (MW)	37.13	46.82	41.93	65.77	68.58	102.83

**Table 9 - Overall MAPE and RMSE for Each Forecaster – Ottawa Dataset**

The overall performance of the Ottawa dataset is comparable to that of the Toronto dataset, as shown in Table 9. The MAPE and RMSE values of the CNN are the lowest, followed by the ANN and LSTM. The CNN had the narrowest error distribution in Figure 11a, while the SNF had the worst overall performance metrics and the widest error distribution.

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE - Peak Values	2.38	3.10	2.78	4.21	4.77	6.98
MAE - Time in Minutes	52	69	64	70	61	79
MBE - Time in Minutes	13	22	16	35	39	0

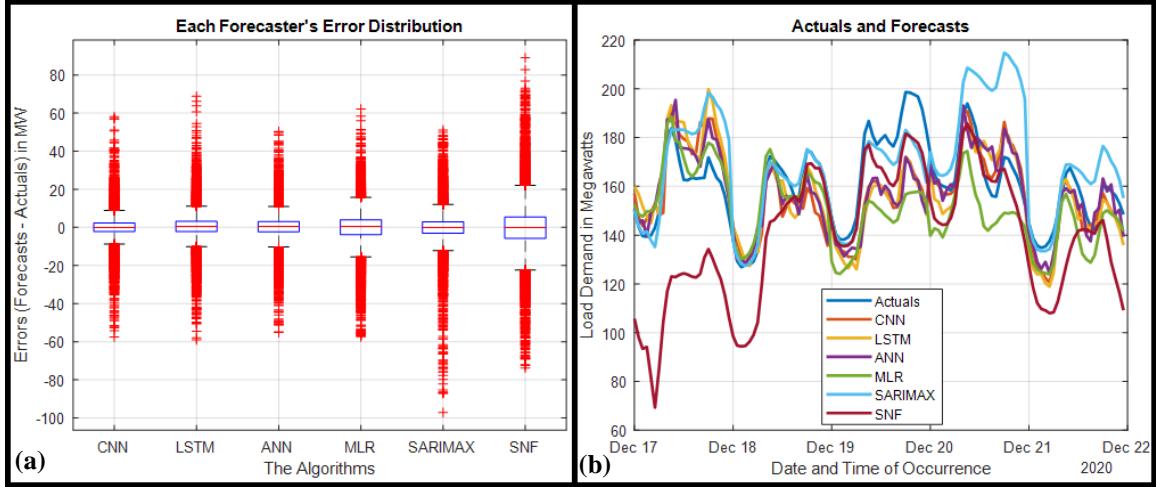
**Table 10 - Matrix Analysis of Peak Values and Time Difference – Ottawa Dataset**

The CNN was also the most accurate at predicting the magnitude of daily peaks, according to MAPE values in Table 10, followed by the ANN and LSTM. According to the MAE values, the CNN was the most accurate at predicting when the daily peaks occurred. The SNF and CNN were the most accurate in terms of bias, according to the MBE values.

### 3.7 The Performance of Forecasters on the Saint John Dataset

Table 11 summarizes the key performance metrics across the test dataset, while Figure 12a depicts the overall distribution of errors for each forecaster across the test dataset. Table 12 summarizes the forecaster's performance in predicting daily peaks. Figure 12b depicts a snapshot of actual and forecasted load demand for the period of December 17th

to December 21st; this period was chosen for visualization because it coincided with the month in which all forecasters performed the worst overall.



**Figure 12: (a) Overall Error Distribution for All Forecasters; (b) Actual and Forecasted Load Demand for December 17th-21st – Saint John Dataset**

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
<b>MAPE (%)</b>	3.89	4.55	4.33	6.11	5.33	9.39
<b>RMSE (MW)</b>	8.06	8.95	8.29	11.06	10.82	16.90

**Table 11 - Overall MAPE and RMSE for Each Forecaster – Saint John Dataset**

The CNN performs best according to the MAPE and RMSE, followed by the ANN and LSTM. The SNF forecaster had the worst overall performance and the widest error distribution.

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
<b>MAPE - Peak Values</b>	3.71	4.07	3.90	5.42	5.28	8.34
<b>MAE - Time in Minutes</b>	160	180	175	191	185	209
<b>MBE - Time in Minutes</b>	13	4	9	-14	-19	-2

**Table 12 - Matrix Analysis of Peak Values and Time Difference – Saint John Dataset**

According to the MAPE values in Table 12, the CNN was the most accurate at predicting the magnitude of daily peaks, followed by the ANN and LSTM. The MAE

values show that the CNN, ANN, and LSTM forecasters were the most accurate at predicting when the daily peaks occurred when compared to other forecasters. According to the MBE values, the SNF and LSTM forecasters were the most accurate in terms of bias.

### **3.8 Conclusion**

Based on the overall accuracy, we can conclude that the CNN performed the best overall, followed by the ANN and the LSTM. The CNN appears to have the tightest distribution of errors across all test datasets in the error distribution plots. The SNF forecaster had the worst overall performance with the widest error distribution, which is good because the fact that the SNF forecaster performs the worst of all forecasters implies that all other forecasters contributed in some way to the forecasts.

In terms of predicting the magnitude of daily peaks, the CNN had the lowest MAPE values across all test datasets; the CNN was followed by the ANN and the LSTM. The MAE values show that the CNN forecaster predicted the time of occurrence of the daily peaks the best; the LSTM and ANN forecasters trail the CNN in the Toronto and Saint John datasets; however, the SARIMAX forecaster outperforms the LSTM and ANN in the Ottawa dataset.

The results in this chapter show that CNN, LSTM, and ANN forecasters had the best overall performance and have a lot of potential in the field of load forecasting. However, Chapter 4 will go into greater detail about evaluating the performance of these forecasters on an hourly, daily, monthly, and seasonal basis.

## **4 Comprehensive Evaluation of the Forecasters' Performance**

Chapter 3 evaluated the overall performance of all forecasters in predicting regular load and daily peaks across all test datasets. This section will look at the performance of all forecasters across all test datasets on hourly, daily, monthly, and seasonal time periods. Following the above analysis, we included a brief discussion of our overall findings for each dataset. To keep the scope of this thesis manageable, only the boxplots of our top forecasters for each section were included here; additional box plots for the remaining forecasters can be found in Appendix B.

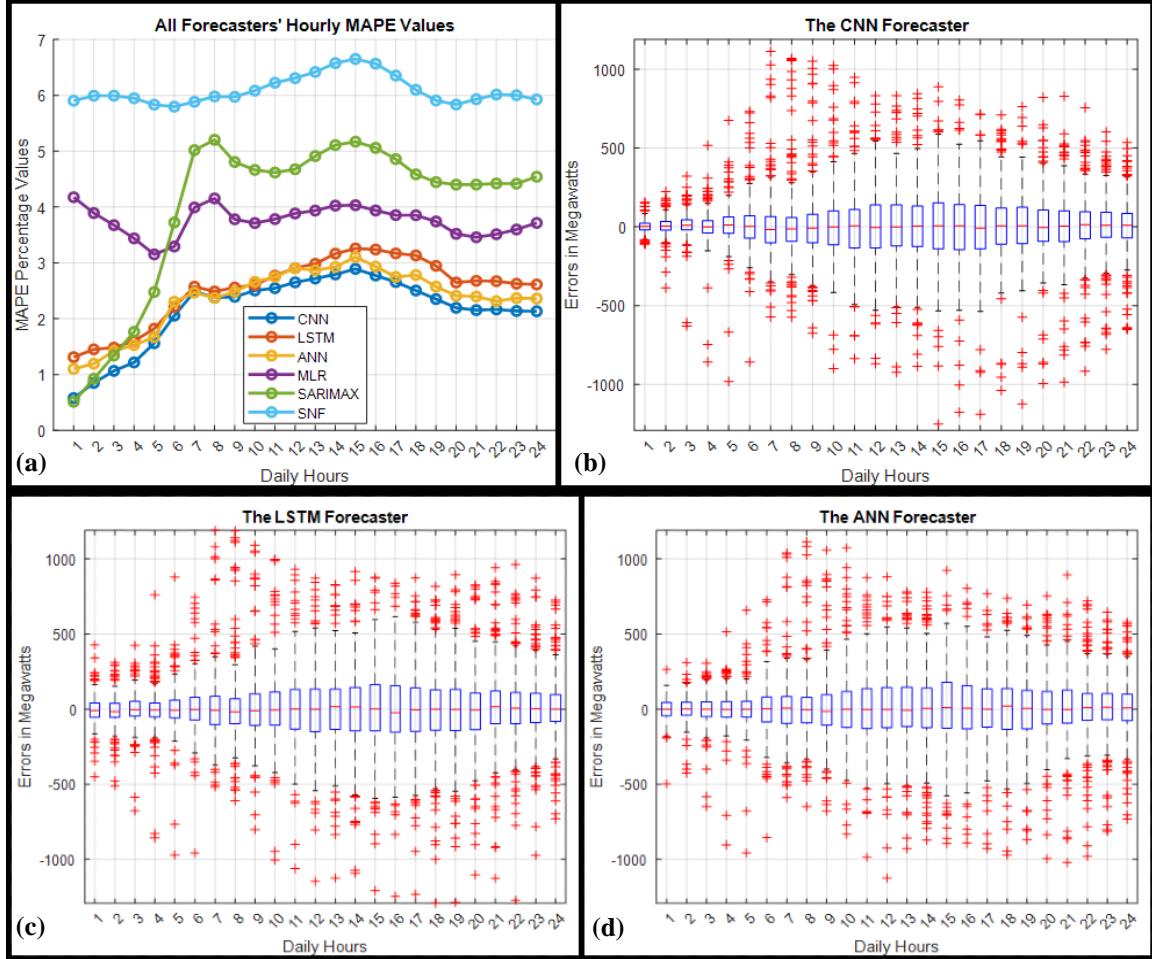
The forecasters' performance during the four seasons of the year (winter, summer, autumn, and spring) was examined; the winter months are December, January, and February. March, April, and May are spring months; June, July, and August are summer months; and September, October, and November are autumn/fall months. It is also worth noting that the hours (00:00...23:00) are labeled as (1:00...24:00) in the hourly performance sections, with 1:00 denoting the first hour and 24:00 denoting the last hour of the day.

### **4.1 The Toronto Dataset**

The Toronto dataset comprises hourly load aggregation measurements taken throughout the city from 2010 to 2019. The years 2010-2018 were used to train the forecasters, and 2019 was used to test them.

#### 4.1.1 The Hourly Performance

Figure 13a illustrates the MAPE values for each forecaster when aggregated as hourly averages. The following plots are boxplots of the hourly error distributions for CNN, LSTM, and ANN forecasters on an hourly basis.



**Figure 13: (a) Hourly MAPE for All Forecasters, (b-d) Hourly Error Distributions for CNN, LSTM, and ANN Forecasters – Toronto Dataset**

##### 4.1.1.1 A Snippet on Hourly Performance

When the average MAPE values in Figure 13a are compared to the box plots, we see a similar pattern of errors, with a wider distribution of errors in situations with higher MAPE

values and vice versa. For each hour of the day, the CNN had the tightest error distribution and the lowest MAPE values. It was followed by the ANN and the LSTM forecasters. Around 15:00, all three forecasters CNN, ANN, and LSTM made their worst predictions; quieter times, such as around 1:00, were predicted much more accurately than busier times.

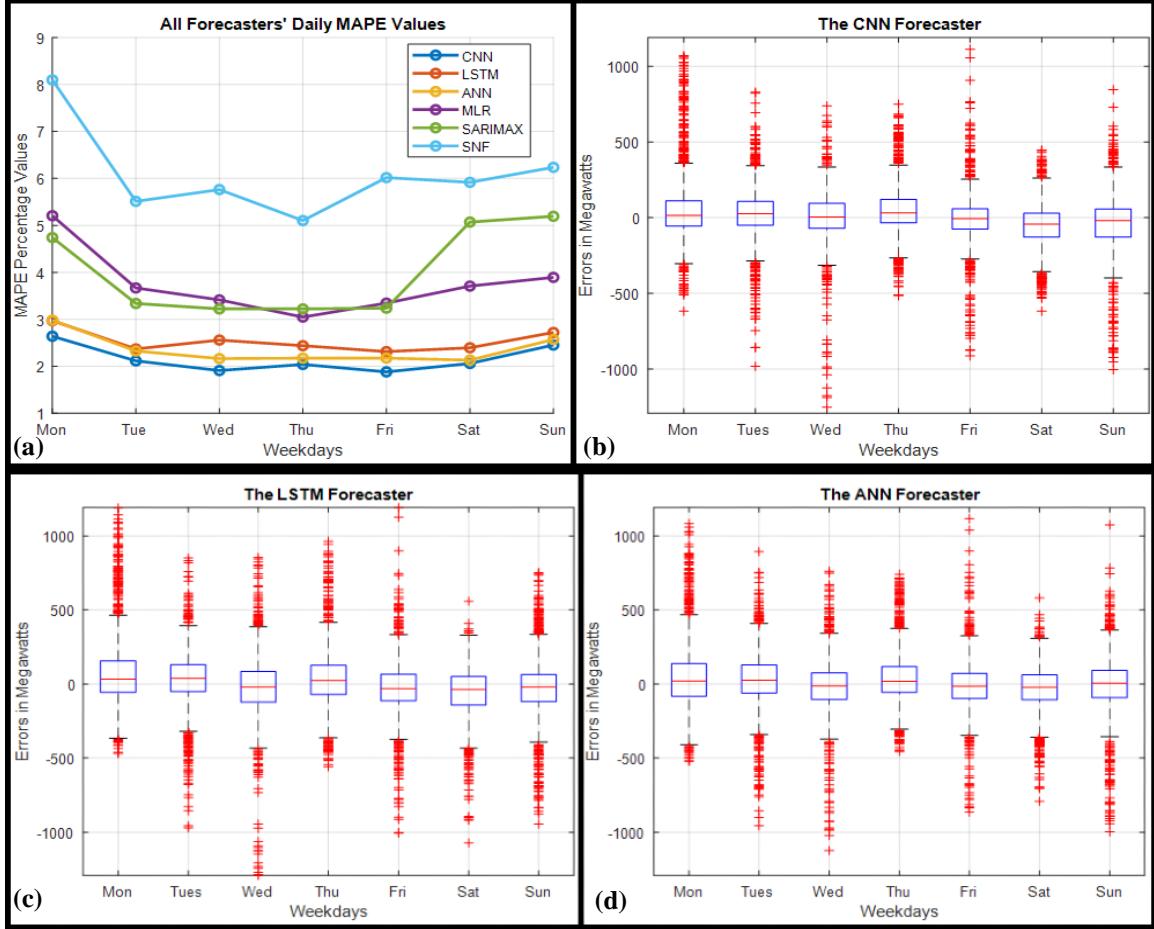
CNN had the best predictions across almost all hours of the day, with ANN and LSTM coming in second and third place, respectively. The SNF performed the worst overall, with the worst predictions occurring between 13:00 and 17:00, which is unsurprising given that it was based on previous week's values and that these are popular times for electricity usage in the Toronto dataset.

#### **4.1.2 The Daily Performance**

Figure 14a shows the MAPE values for each forecaster aggregated as daily averages for each day of the week. The following plots are boxplots of the daily error distributions for CNN, LSTM, and ANN forecasters on a daily basis.

##### **4.1.2.1 A Snippet on Daily Performance**

When we compare the MAPE values in Figure 14a to the boxplots of the error distribution; we can see, Monday was the worst day for almost all forecasters to predict, but Sunday was the worst day for the SARIMAX forecaster. Every other forecaster's second worst day was Sunday, but SARIMAX's second worst day was Saturday. Tuesdays through Fridays were the most predictable days for the forecasters. CNN's MAPE values were the lowest on all seven days of the week, and all of its boxplots were the narrowest. The ANN is ranked second, with the LSTM edging it out in Monday's predictions. The LSTM forecaster is ranked third. Overall, the SNF performed the worst.



**Figure 14: (a) Daily MAPE for All Forecasters, (b-d) Daily Error Distributions for CNN, LSTM, and ANN Forecasters – Toronto Dataset**

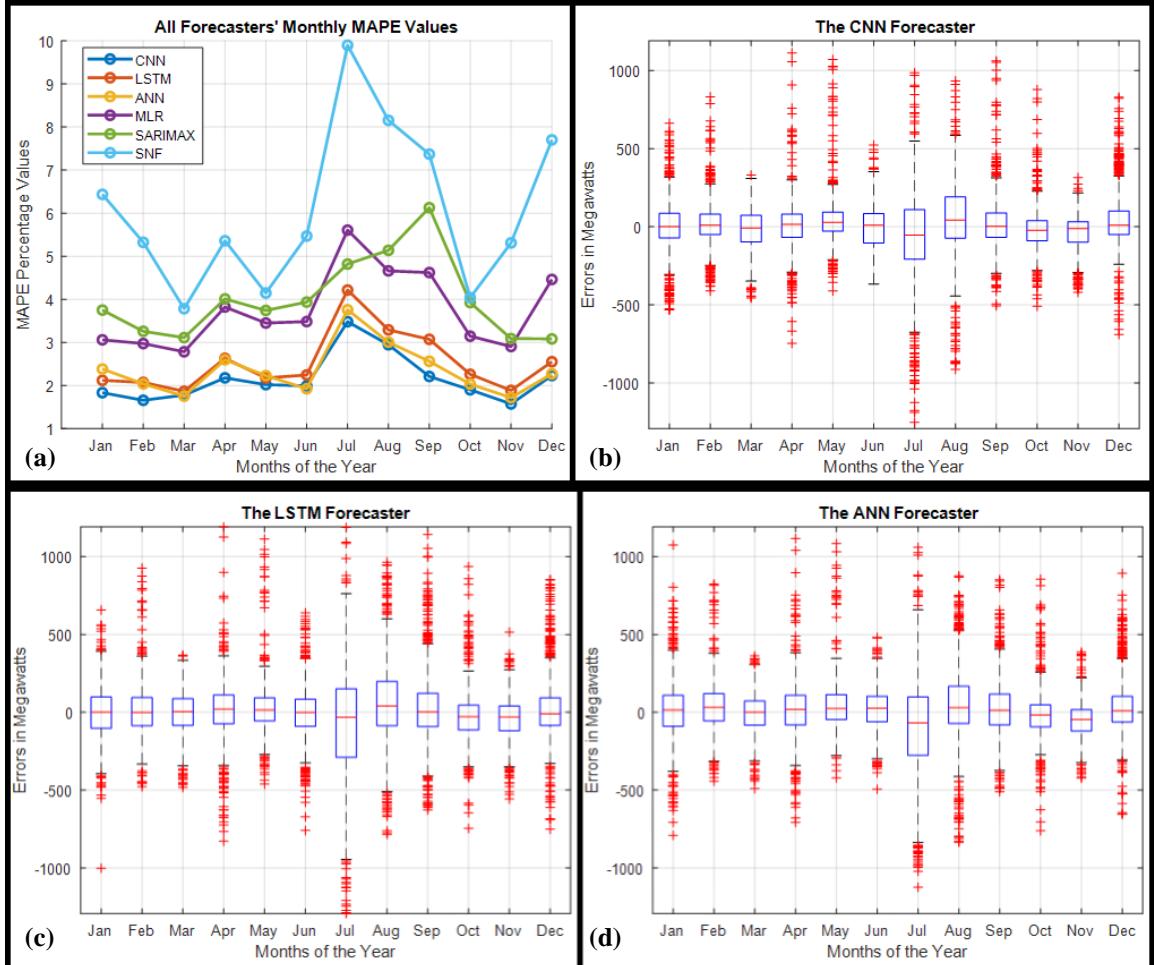
#### 4.1.3 The Monthly Performance

The MAPE values for each forecaster are aggregated in Figure 15a as monthly averages for each month of the year 2019. The following plots are boxplots of the monthly error distributions for CNN, LSTM, and ANN forecasters on a monthly basis.

##### 4.1.3.1 A Snippet on Monthly Performance

Except for SARIMAX, which had the worst predictions in September, all forecasters had their worst predictions in July, but the other forecasters also found September difficult

to predict. August was the month in which all forecasters made their second-worst predictions. Forecasters found February, March, and November to be relatively simple to predict. Over ten months, the CNN had the lowest MAPE values and was only outperformed by the ANN in March and June. The ANN is ranked second because it was only surpassed by the LSTM in January, while the LSTM is ranked third. The SNF had the worst overall performance across all months.



**Figure 15: (a) Monthly MAPE for All Forecasters, (b-d) Monthly Error Distributions for CNN, LSTM, and ANN Forecasters – Toronto Dataset**

#### 4.1.4 Performance During the Seasons

The table below summarizes the MAPE and RMSE values obtained in the Toronto test dataset for the average of various seasons. Summer was the season that forecasters had the most trouble predicting. CNN had the lowest MAPE and RMSE values across all four seasons. The ANN is ranked second, having only been surpassed in the winter by the LSTM, which is ranked third. Autumn was the season with the lowest metric values for the CNN and ANN, while spring was the season with the lowest metric values for the LSTM. Across all seasons, the SNF had the worst overall performance.

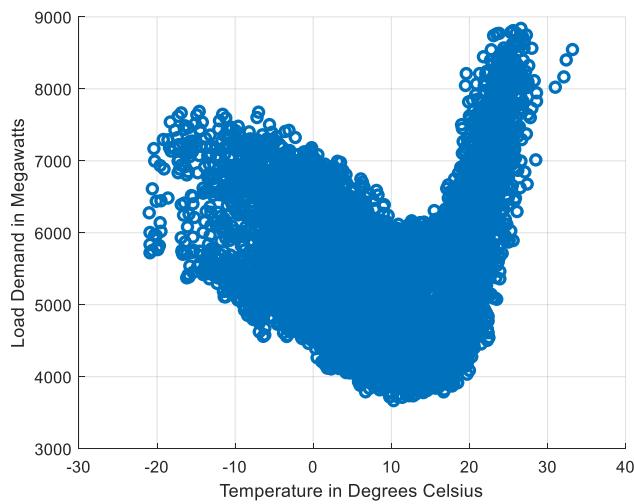
Winter						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	1.92	2.25	2.26	3.43	3.37	6.55
RMSE (MW)	172.22	192.89	206.20	270.47	283.12	508.21
Spring						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	1.99	2.22	2.16	3.35	3.62	4.42
RMSE (MW)	160.14	174.98	170.82	236.15	266.77	338.62
Summer						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	2.82	3.26	2.93	4.60	4.64	7.87
RMSE (MW)	254.72	293.83	267.77	371.81	384.33	629.32
Autumn / Fall						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	1.89	2.40	2.14	3.55	4.38	5.56
RMSE (MW)	153.31	195.48	166.36	272.44	337.96	430.89

Table 13 - Seasonal MAPE and RMSE for the Toronto Dataset

#### 4.1.5 Comprehensive Analysis Discussion

In the Toronto dataset, the summer months of June through August saw the highest demand for electricity; these were also the months during which forecasters had the most difficulty forecasting, as illustrated in the seasonal and monthly time periods section. July

was the month with the highest demand and the highest error rates among most forecasters. This can be attributed to two factors: first, the increased demand can be attributed to the fact that the summer air in Toronto is typically hot throughout the day, including evenings, necessitating the use of air conditioning by almost everyone. Furthermore, Toronto is a popular tourist destination, and everyone wants to visit during the summer when the weather is nicer. August had the second highest number of errors after July, for the same reasons.



**Figure 16 - Scatter Plot of Load Demand versus Temperature – Toronto Dataset**

The scatter plot, as shown in Figure 16, shows a strong correlation between demand and temperature. The highest demand occurs during the summer when the temperature is higher. When the temperature drops, especially in the winter, the second-highest demand occurs. Furthermore, because demand is most stable in the spring and autumn, forecasters found these seasons to be easier to forecast. The overall accuracy of the SNF suggests that load demand is relatively stable in the spring and autumn.

Peak demand in Toronto occurs between 16:00 and 21:00, as we discovered while working on the dataset. The majority of forecasters found it easier to forecast quieter times, such as around midnight in the early mornings, than busy times, such as late mornings and afternoons, when everyone is awake and demand for electricity is high.

We saw a lot of fluctuation in demand on Mondays, and because almost all of our forecasters used the previous day's demand as a predictor, Mondays and Sundays were the most difficult to predict for the majority of forecasters. Forecasters were more accurate from Tuesday to Friday, during the middle of the week; because these days were more stable and have a high degree of similarity.

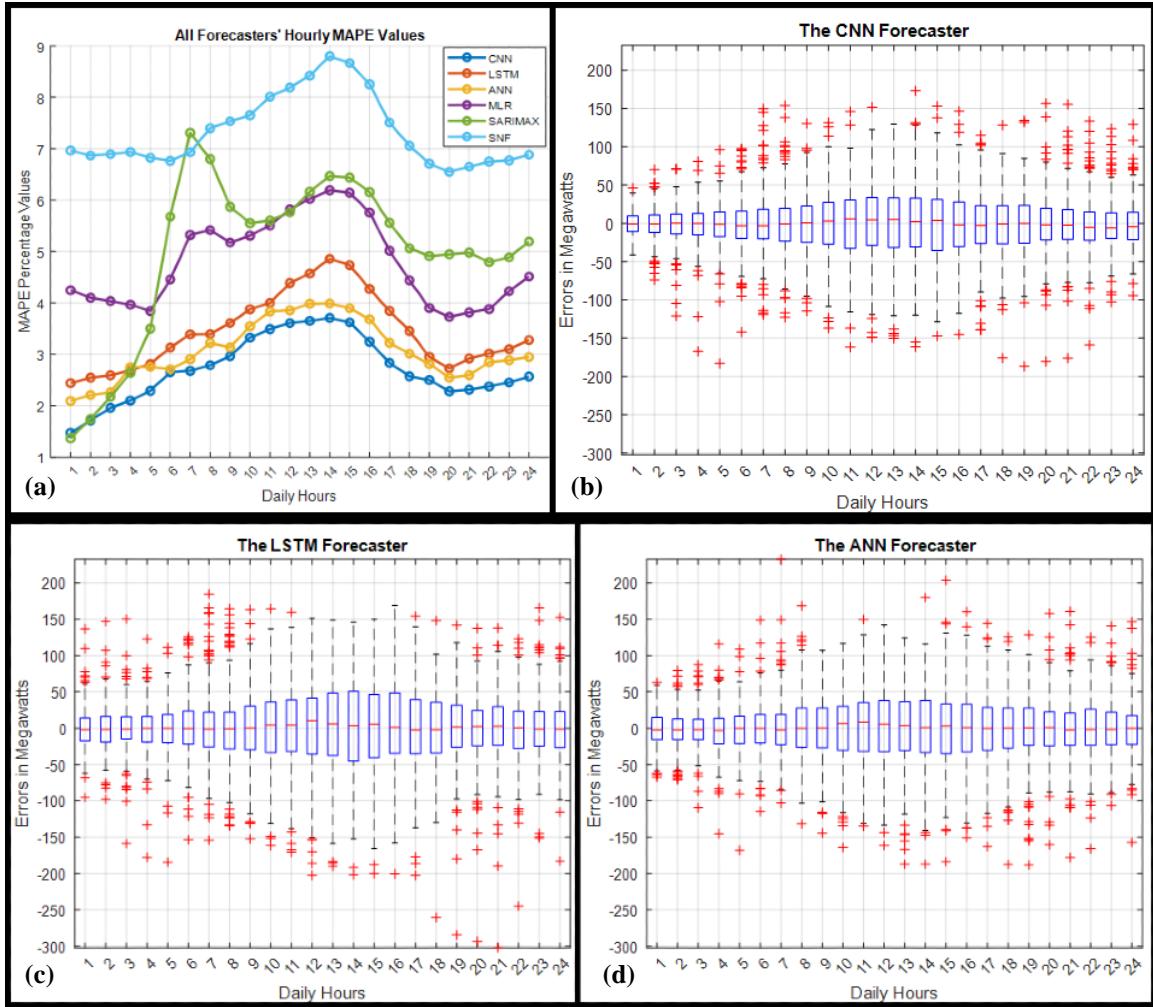
The CNN forecaster performed the best across all time periods and seasons, with the ANN and LSTM finishing second and third, respectively. The SNF is ranked last because it has consistently performed poorly across all time periods and seasons.

## 4.2 The Ottawa Dataset

The Ottawa dataset comprises hourly load aggregation measurements taken throughout the city from 2010 to 2019. The years 2010-2018 were used to train the forecasters, and 2019 was used to test them.

### 4.2.1 The Hourly Performance

Figure 17a illustrates the MAPE values for each forecaster when aggregated as hourly averages. The following plots are boxplots of the hourly error distributions for CNN, LSTM, and ANN forecasters on an hourly basis.



**Figure 17: (a) Hourly MAPE for All Forecasters, (b-d) Hourly Error Distributions for CNN, LSTM, and ANN Forecasters – Ottawa Dataset**

#### 4.2.1.1 A Snippet on Hourly Performance

When we compare the average MAPE values in Figure 17a to the box plots, we see a similar pattern of errors, with a wider distribution of errors in cases where the MAPE value is higher, and vice versa. While the CNN had the tightest error distribution and lowest MAPE values for nearly every hour of the day, the SARIMAX outperformed it at 01:00. The ANN comes in second, with the LSTM outperforming it only during the 4:00 hour;

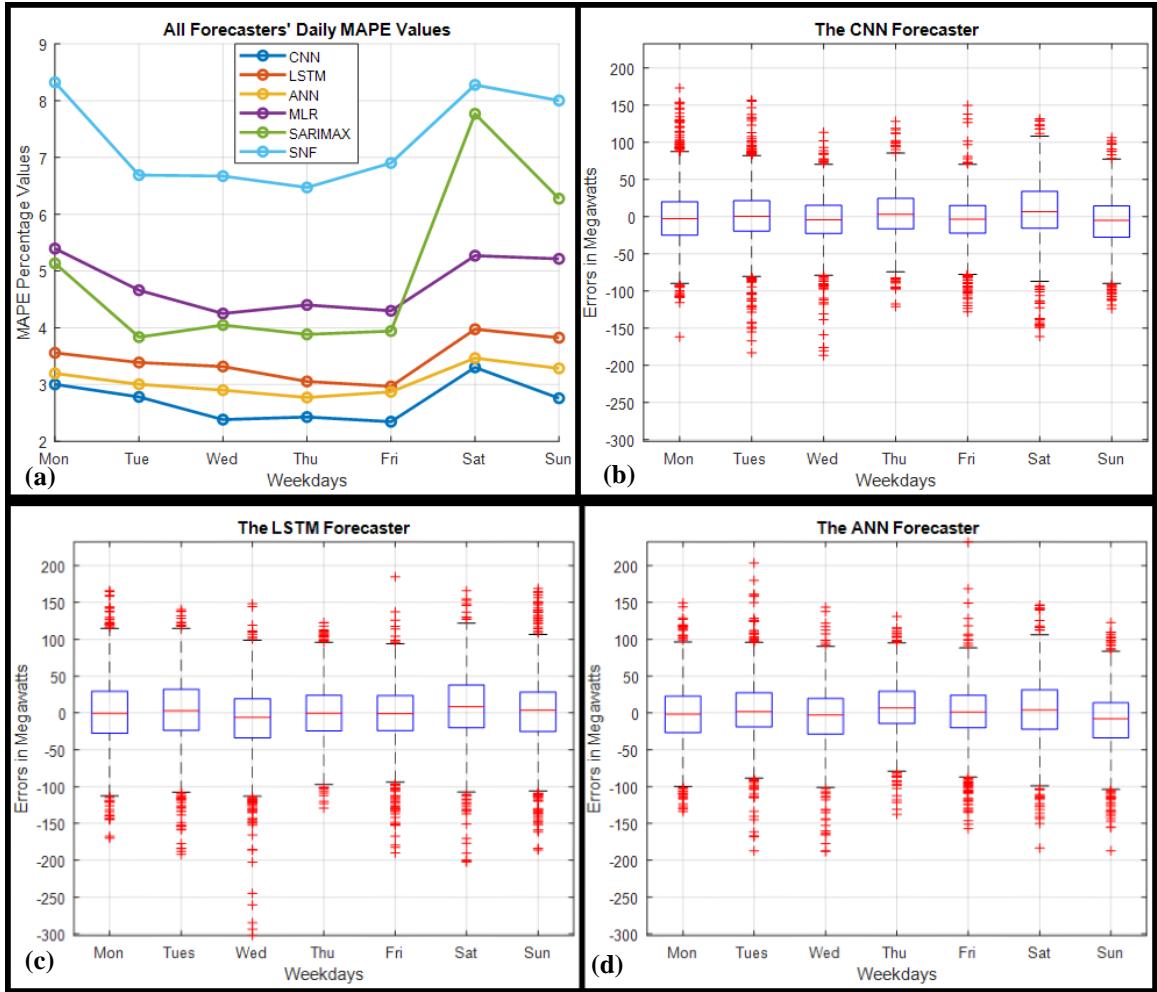
the SARIMAX also outperforms the ANN between 01:00 and 04:00 hours. The LSTM comes in third place. CNN, LSTM, and ANN all made their worst predictions around 14:00. They were more accurate at predicting quieter times, such as around midnight, than they were at busier times. The SNF performs poorly overall, outperforming the SARIMAX only at 7:00, which is also when the SARIMAX made its worst predictions.

#### **4.2.2 The Daily Performance**

Figure 18a shows the MAPE values for each forecaster aggregated as daily averages for each day of the week. The following plots are boxplots of the daily error distributions for CNN, LSTM, and ANN forecasters on a daily basis.

##### **4.2.2.1 A Snippet on Daily Performance**

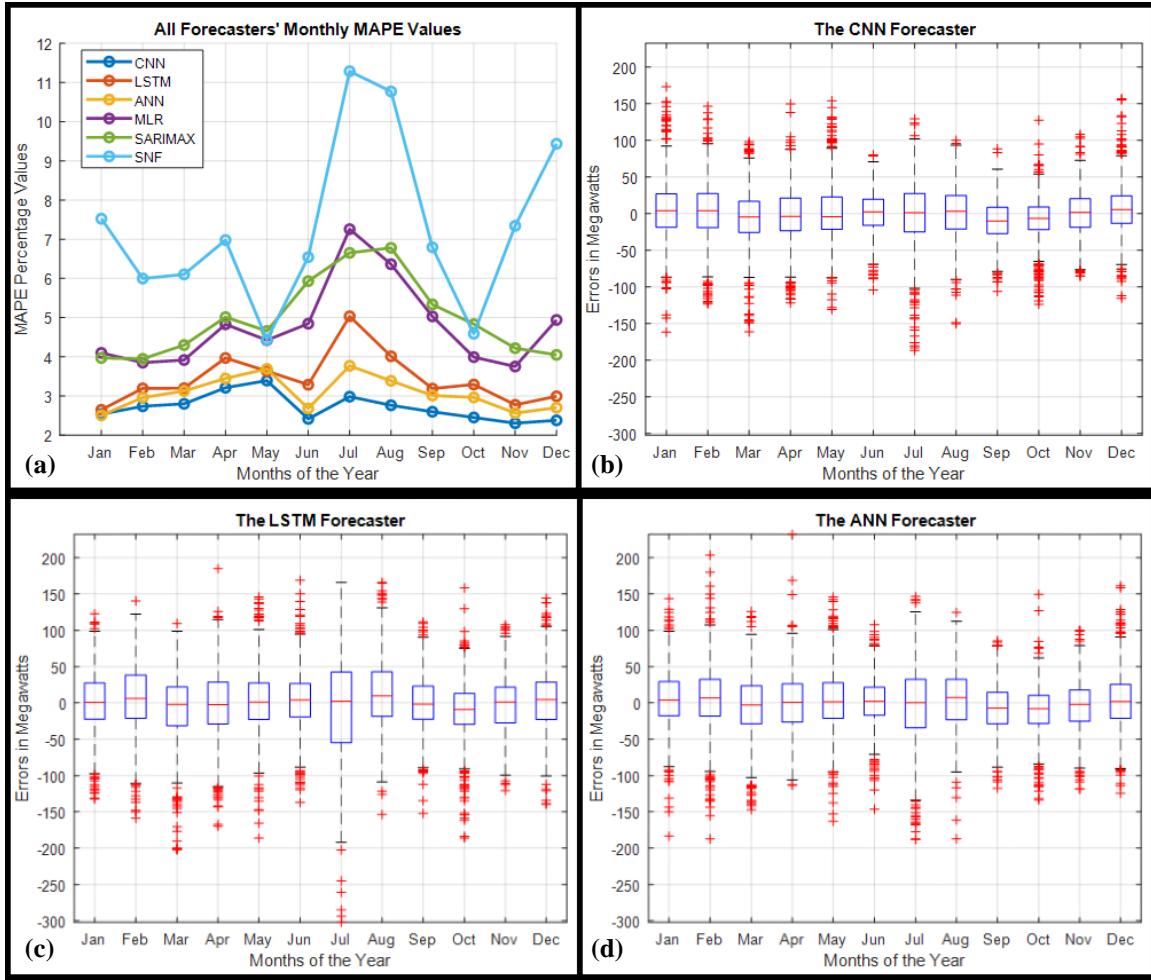
When the MAPE values in Figure 18a are compared to the error distribution boxplots, we can see that Saturdays and Mondays are the most difficult days to forecast for almost all forecasters, with the exception of the SARIMAX, which has its most difficult predictions on Saturday. The most predictable days were Tuesdays through Fridays. Among all forecasters, the CNN had the lowest MAPE values and the narrowest boxplots. The ANN comes in second, and the LSTM comes in third. The SNF performed the worst overall.



**Figure 18: (a) Daily MAPE for All Forecasters, (b-d) Daily Error Distributions for CNN, LSTM, and ANN Forecasters – Ottawa Dataset**

#### 4.2.3 The Monthly Performance

The MAPE values for each forecaster are aggregated in Figure 19a as monthly averages for each month of the year 2019. The following plots are boxplots of the monthly error distributions for CNN, LSTM, and ANN forecasters on a monthly basis.



**Figure 19:** (a) Monthly MAPE for All Forecasters, (b-d) Monthly Error Distributions for CNN, LSTM, and ANN Forecasters – Ottawa Dataset

#### 4.2.3.1 A Snippet on Monthly Performance

When the MAPE values in Figure 19a are compared to the box plots, it is clear that, with the exception of CNN, July was the most difficult month to forecast. CNN determined that July is a relatively easier month to predict, with the worst predictions coming in April and May. Across almost all months, the CNN had the lowest MAPE values and the narrowest error distribution; it is only surpassed in January by the ANN. The ANN is ranked second, having been surpassed by the LSTM in May. The LSTM is ranked third.

Most forecasters considered January to March and October to November to be relatively easier forecasting periods. Overall, the SNF performed poorly, outperforming the SARIMAX only in May and October.

#### 4.2.4 Performance During the Seasons

The table below summarizes the MAPE and RMSE values obtained from the Ottawa test dataset for the average across all four seasons.

Winter						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	2.55	2.94	2.86	4.30	3.99	7.73
RMSE (MW)	39.68	43.83	43.37	63.71	62.66	111.73
Spring						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	3.13	3.60	3.43	4.39	4.65	5.83
RMSE (MW)	40.02	46.08	43.29	55.12	59.98	76.45
Summer						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	2.72	4.12	3.22	6.17	6.46	9.57
RMSE (MW)	37.62	56.78	43.97	86.15	87.49	131.76
Autumn / Fall						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	2.45	3.09	2.91	4.26	4.80	6.22
RMSE (MW)	30.42	38.51	35.85	52.51	60.04	81.81

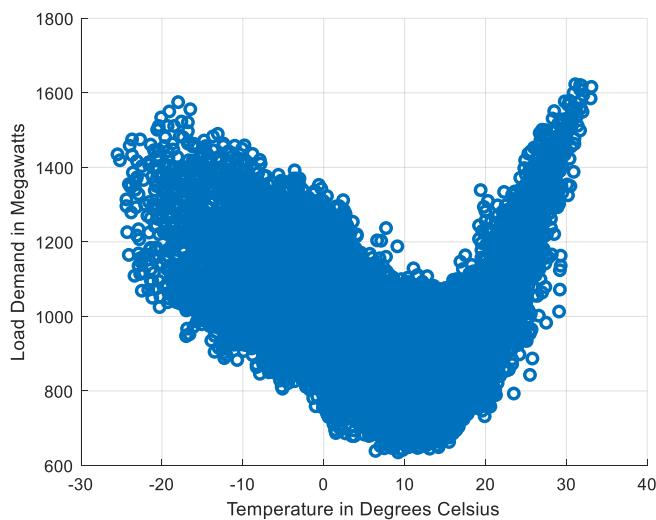
Table 14 - Seasonal MAPE and RMSE for the Ottawa Dataset

CNN and ANN made their most dire predictions in the spring. All other forecasters, including the LSTM, had their worst predictions in the summer. The LSTM and ANN had the easiest time forecasting the winter months. CNN and MLR made their best predictions in the autumn. To facilitate interpretation, all previous observations were made with MAPE values rather than RMSE values. However, CNN had the lowest MAPE and RMSE values

across all four seasons. The ANN comes in second, with the LSTM coming in third. The SNF had the worst overall performance metric values across all seasons.

#### 4.2.5 Comprehensive Analysis Discussion

The Ottawa dataset is similar to the Toronto dataset, but there are a few key differences. The months with the highest average demand were January, February, July, and December. Except for CNN, July proved to be the most difficult month to forecast. CNN ranked July as the third most difficult month, trailing only April and May. Spring was the most difficult season to forecast for the ANN and CNN, with summer coming in second. Most forecasters, including the LSTM, found summer to be the most difficult season to forecast. Winter was the most predictable season for the LSTM and the ANN.



**Figure 20 - Scatter Plot of Load Demand versus Temperature – Ottawa Dataset**

Figure 20 shows a scatter plot for the Ottawa dataset, which is similar to the scatter plot for the Toronto dataset. They appear to be identical, but the Ottawa scatterplot has nearly the same level of demand in the winter and summer, when temperatures are cold and hot,

respectively. In the Toronto dataset, demand was significantly higher in the summer months than it was in the winter months.

The average peak demand for the Ottawa dataset occurs between 16:00 and 21:00, as we discovered while working on the dataset. All forecasters, with the exception of SARIMAX, made their worst predictions between 11:00 and 16:00, when the majority of people are awake and using electricity or working.

Mondays and Saturdays were the most difficult days to forecast, possibly because Monday is the first working day of the week and Saturday is the first day of the weekend. This is similar to what we saw in the Toronto section because almost all of our forecasters used the previous day's demand as a predictor, and Monday demand differs from Sunday demand, as does Saturday demand from Friday demand. Forecasters also had a difficult time predicting Sundays, but Mondays and Saturdays were the most difficult. Among all forecasters, the middle of the week, specifically Tuesdays through Fridays, were the easiest days to predict.

The CNN forecaster performed the best across all time periods and seasons, with the ANN and LSTM forecasters coming in second and third place, respectively. The SNF is ranked last due to its poor performance across all periods and seasons; it only outperformed the SARIMAX in a few cases.

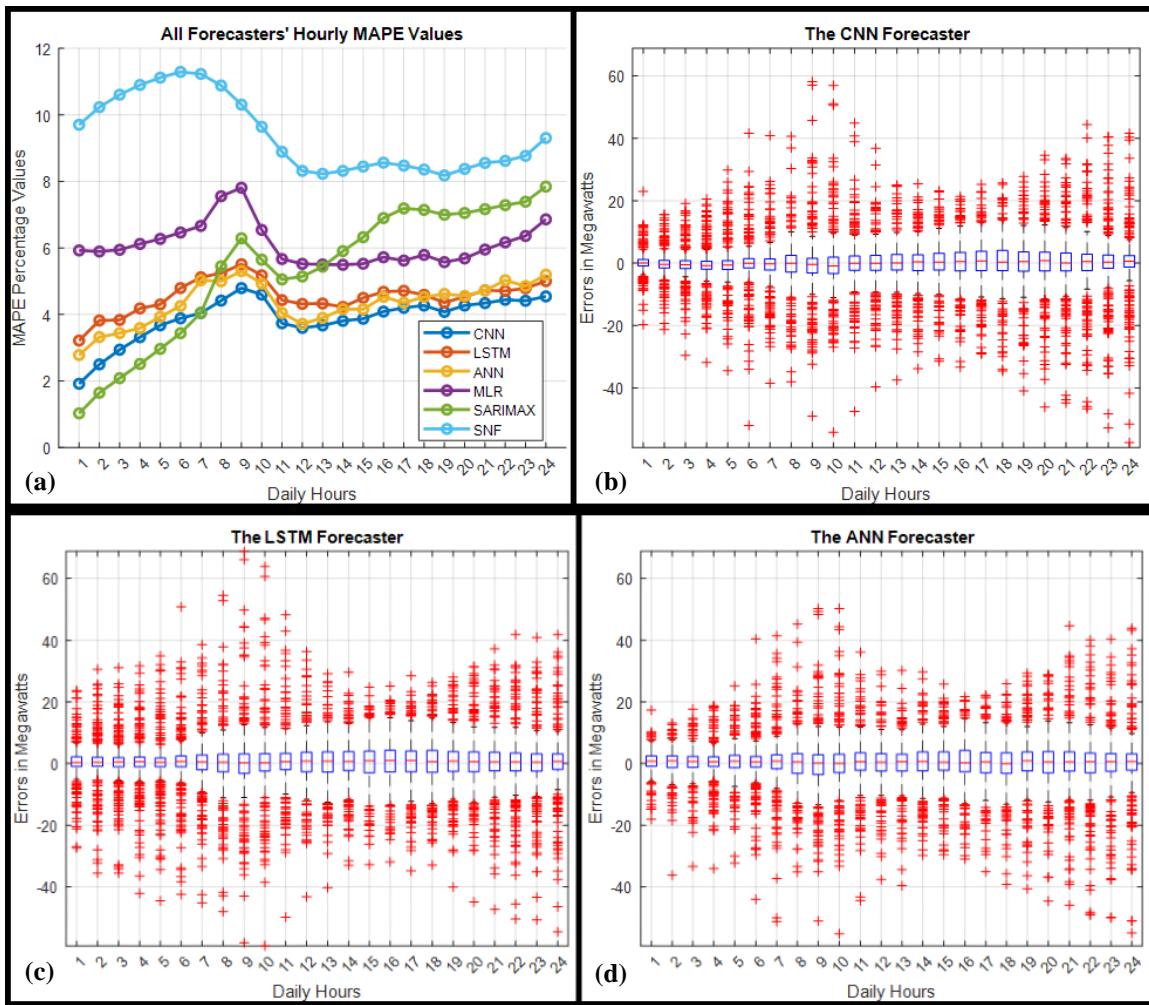
### 4.3 The Saint John Dataset

The Saint John dataset consists of hourly load aggregation measurements collected throughout the city between January 2018 and October 2021. The forecasters were trained

between January 1<sup>st</sup>, 2018 and October 20<sup>th</sup>, 2020, and tested between October 21<sup>st</sup>, 2020, and October 20th, 2021.

#### 4.3.1 The Hourly Performance

Figure 21a illustrates the MAPE values for each forecaster when aggregated as hourly averages. The following plots are boxplots of the hourly error distributions for CNN, LSTM, and ANN forecasters on an hourly basis.



**Figure 21: (a) Hourly MAPE for All Forecasters, (b-d) Hourly Error Distributions for CNN, LSTM, and ANN Forecasters – Saint John Dataset**

#### **4.3.1.1 A Snippet on Hourly Performance**

The average peak demand in Saint John occurs between 10:00 and 13:00, with a second peak around 19:00, as we discovered while working on the datasets. When we compare the average MAPE values in Figure 21a to the box plots, we see a similar pattern of errors, with a wider distribution of errors in cases where the MAPE value is higher, and vice versa. With the exception of the SNF and SARIMAX, all forecasters had their worst predictions around 9:00; the SNF's worst predictions were around 6:00, and the SARIMAX's worst predictions were around 24:00.

The SARIMAX forecaster made the most accurate predictions between 1:00 and 6:00; however, as demand increased throughout the day, the SARIMAX forecaster lost out to CNN. CNN performed best between 7:00 and 24:00, as well as during the day's peak hours. As a result, CNN had the best overall performance and the tightest error distribution. The ANN is ranked second, with the SARIMAX outperforming it between 1:00 and 7:00 and the LSTM outperforming it between 19:00 and 22:00. The LSTM is ranked third. The SNF performed the worst overall.

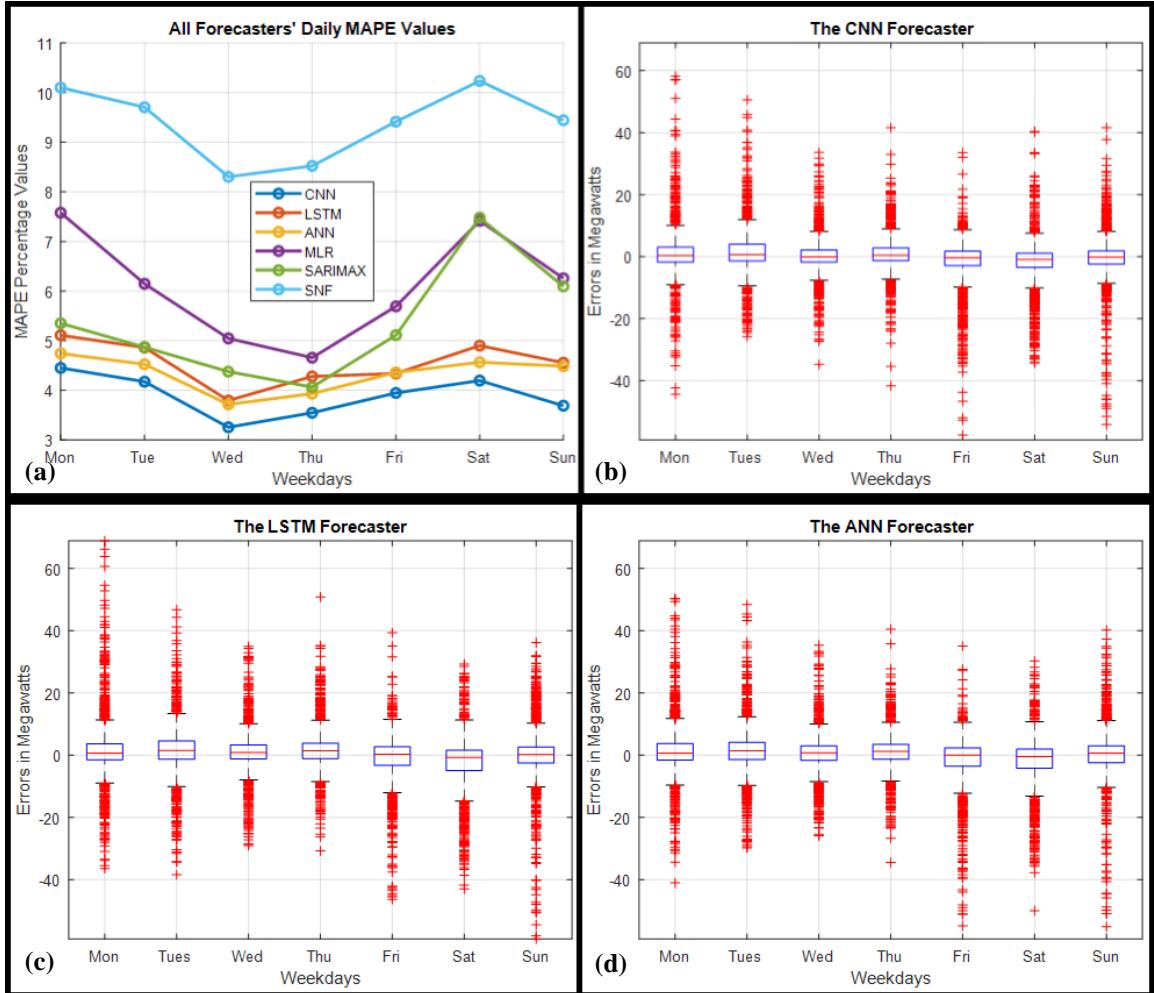
#### **4.3.2 The Daily Performance**

Figure 22a shows the MAPE values for each forecaster aggregated as daily averages for each day of the week. The following plots are boxplots of the daily error distributions for CNN, LSTM, and ANN forecasters on a daily basis.

##### **4.3.2.1 A Snippet on Daily Performance**

When we compare the MAPE values in Figure 22a to the error distribution boxplots, we can see that the forecasters had the most difficulty predicting Mondays and Saturdays.

The most predictable days were Wednesdays and Thursdays. The CNN provided the best predictions on all seven days of the week. The ANN is second, with the LSTM outperforming it only on Fridays, and the LSTM is third, with the SARIMAX outperforming it only on Thursdays. The SNF performed the worst overall.



**Figure 22: (a) Daily MAPE for All Forecasters, (b-d) Daily Error Distributions for CNN, LSTM, and ANN Forecasters – Saint John Dataset**

### 4.3.3 The Monthly Performance

Figure 23 shows the MAPE values for each forecaster as monthly averages for each month of the year. Figure 24 depicts monthly boxplots of error distributions for CNN, LSTM, ANN, and SARIMAX forecasters.

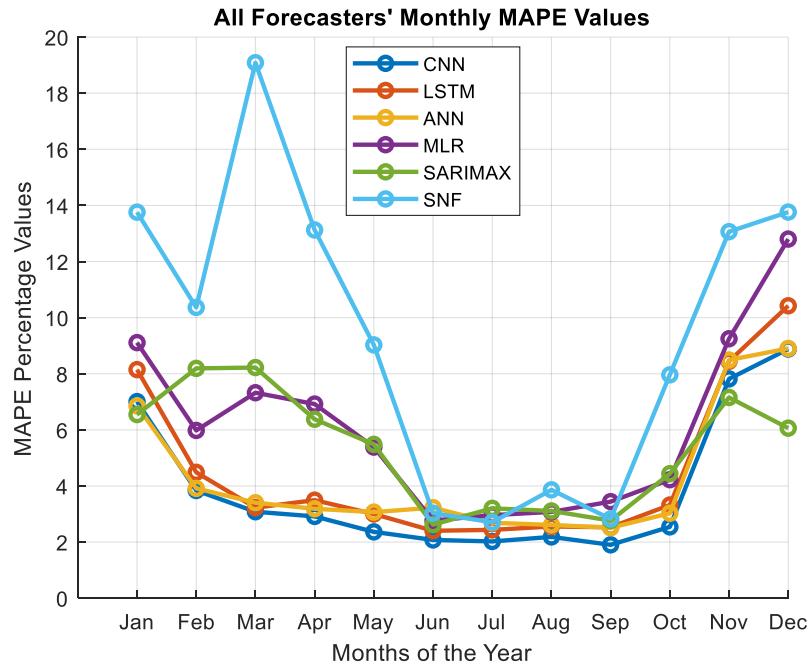
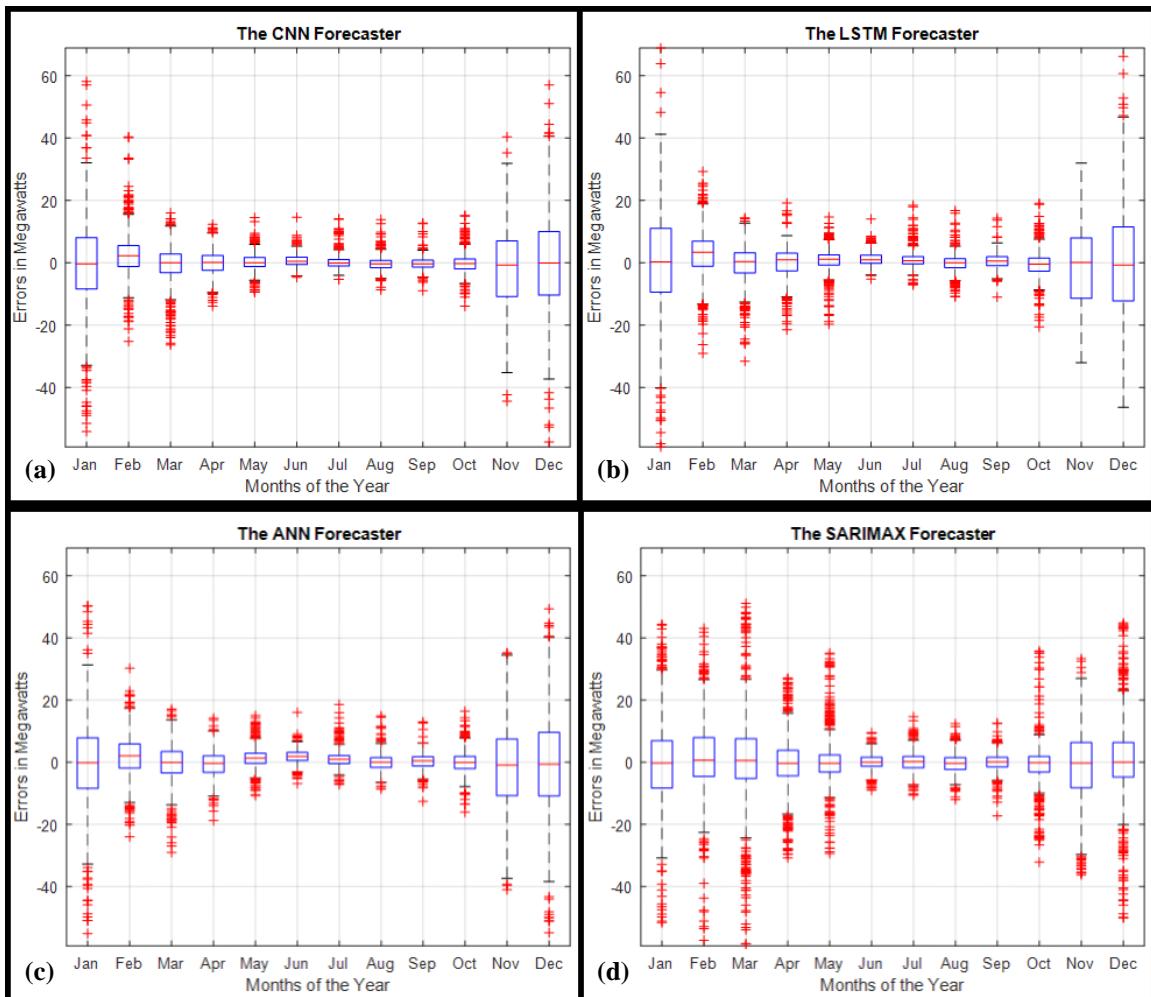


Figure 23 - Monthly MAPE for Each Forecaster – Saint John Dataset

#### 4.3.3.1 A Snippet on Monthly Performance

The city of Saint John experiences its highest demand during the winter months. With the exception of the SARIMAX and SNF, the majority of the forecasters made their worst predictions in December. The SARIMAX forecaster produced the most accurate forecasts in January, November, and December, while CNN's predictions for the rest of the year were the most accurate.

Because the ANN and LSTM are inextricably linked, determining which produces more accurate predictions is difficult. In January, February, April, September, October, and December, the ANN outperformed the LSTM. In March, May, June, July, August, and November, the LSTM outperformed the ANN. Furthermore, in June, the MLR and SNF outperformed the ANN. In April, the SARIMAX outperforms the MLR. The most error-prone months for the SNF were January to May and October to December.



**Figure 24 - Monthly Error Distribution for CNN, LSTM, ANN, and SARIMAX Forecasters – Saint John Dataset**

Each forecaster has a similar performance to the others, with one outperforming the others one month and the others the next. As a result, ranking them from first to third is difficult, because forecasters like SARIMAX performed better during the winter months but less well during the rest of the year. CNN had the most accurate predictions for the remaining nine months of the year, but it cannot be declared the winner because SARIMAX's best months coincided with the months with the highest demand. The SNF outperformed some forecasters during warmer months but struggled during colder months.

#### 4.3.4 Performance During the Seasons

The table below summarizes the MAPE and RMSE values obtained for the average of the Saint John test dataset across all seasons.

Winter						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	6.67	7.79	6.64	9.41	6.89	12.70
RMSE (MW)	13.33	14.95	13.34	17.57	15.55	23.08
Spring						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	2.79	3.24	3.22	6.54	6.70	13.75
RMSE (MW)	4.41	4.99	4.81	9.24	12.27	20.41
Summer						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	2.09	2.47	2.84	2.94	2.98	3.20
RMSE (MW)	2.20	2.65	2.75	2.97	3.03	3.36
Autumn / Fall						
Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
MAPE (%)	4.09	4.76	4.68	5.62	4.78	7.95
RMSE (MW)	7.74	8.23	8.29	9.48	8.29	13.66

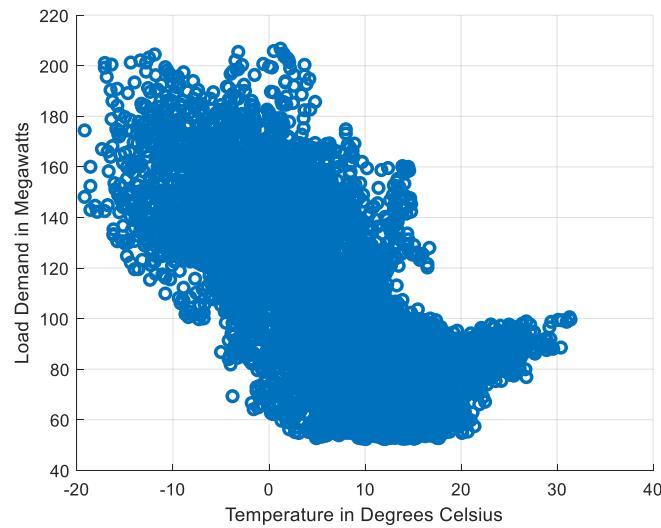
Table 15 - Seasonal MAPE and RMSE for the Saint John Dataset

Except for SNF, all forecasters had their worst predictions in the winter; SNF had its worst predictions in the spring. Summer was the most predictable season for forecasters

because demand was more stable. CNN performed the best overall across all four seasons, while SNF performed the worst.

#### 4.3.5 Comprehensive Analysis Discussion

The Saint John test dataset differs significantly from the test datasets in Toronto and Ottawa. The Saint John dataset has the highest load demand during the winter months when temperatures are at their lowest, as shown in the scatter plot below, while demand is relatively lower during the remaining months of the year. Summer months had the lowest demand for this dataset. As a result, most forecasters found the winter months to be the most difficult to predict, while the summer months were the easiest.



**Figure 25 - Scatter Plot of Load Demand versus Temperature – Saint John Dataset**

The SARIMAX forecaster was the most accurate in some of the winter months with the highest demand, such as January, November, and December, but it was not very accurate in the remaining months of the year, whereas the CNN forecaster was the most accurate. This means that the SARIMAX forecasts can be used as a primary reference point for these

specific months, while the CNN forecasts can be used for the rest of the year. A more in-depth investigation into why the SARIMAX performance was accurate during the winter months could have been conducted, but this is beyond the scope of this thesis, which was to assess the performance of deep learning techniques to see if they have potential in the area of load forecasting. One possible explanation for the SARIMAX forecaster's performance is that the Saint John test dataset is highly affected by weather conditions, particularly during the winter, and the only exogenous variable that the SARIMAX forecaster uses are temperature lag values from the previous 28 days as well as the 24-hour temperatures values for the day we are predicting.

Mondays and Saturdays were the most difficult days to forecast for the same reasons as the Ottawa and Toronto datasets, primarily because nearly all of our forecasters use the previous day's demand as a predictor. Wednesdays and Thursdays were the most predictable days for all forecasters. The CNN performed the best overall across all days of the week, followed by the ANN and LSTM.

The peak period for demand in Saint John is between 10:00 and 13:00, as we discovered while working on the dataset. The majority of forecasters struggled to forecast around 9:00. The SARIMAX forecaster was the most accurate between 1:00 and 6:00, but CNN was the most accurate throughout the rest of the day, including the peak hours.

While CNN performed best overall on an hourly, daily, monthly, and seasonal basis, SARIMAX's accuracy during the majority of the winter months should not be overlooked. Furthermore, the SNF also outperformed a few forecasters from June to September, when demand was low and generally stable.

## 5 Conclusion

This thesis's purpose and objectives, as stated in Chapter 1, have been fully met. We identified and implemented benchmarks as well as deep learning forecasters, and the deep learning forecasters' performance in predicting regular load and daily peaks was compared across three distinct datasets. A more in-depth analysis was then carried out to assess the forecasters' performance at various times of the day, days of the week, months of the year, and seasons. The subsections that follow provide a summary of this study, contributions, and potential future research directions.

### 5.1 Summary

Using three different datasets, we compared four benchmark forecasters to two deep learning techniques, CNN and LSTM, with the goal of determining overall performance in forecasting regular load and daily peaks. In terms of overall accuracy, CNN, LSTM, and ANN were the most accurate forecasters. The performance of all forecasters was then examined on an hourly, daily, monthly, and seasonal basis across all test datasets.

CNN, ANN, and LSTM were the top forecasters across all time periods and seasons in the Toronto and Ottawa datasets; however, in the Saint John dataset, the SARIMAX forecaster outperformed them in January, November, and December, but the SARIMAX forecaster was not as accurate for the rest of the year. This is to be expected given that forecasters perform differently across datasets and time periods; each dataset is unique based on the area from which it was obtained, such as the type of customers in the area (residential, commercial, industrial), and the weather conditions of the area, which can affect the heating and cooling requirements of the area. The SNF had the worst overall

performance across all periods and seasons, which is to be expected given that load demand is generally not stable week to week, as we discovered when working with the various datasets.

Mondays, Saturdays, and Sundays were difficult to predict for all forecasters, but the middle of the week was easier to predict because demand was more stable. The months with the highest demand, such as July in the Toronto and Ottawa datasets and December in the Saint John dataset, were the most difficult for forecasters to predict, while most months with lower or relatively stable demand were easier. The forecasters were also able to predict quieter times, such as around midnight and early mornings, much more accurately than busier times.

Overall, CNN and LSTM ranked first and third among all forecasters in predicting the test datasets over all time periods. The ANN is ranked second, but most LSTM and ANN predictions were quite similar, and the LSTM outperformed the ANN in several cases. Based on our overall analysis and comparison across datasets, we can conclude that deep learning techniques such as CNN and LSTM have a lot of potential in the field of load forecasting and can assist researchers, utilities, and power system operators in improving their load forecasting performance.

## **5.2 Contributions**

Deep learning techniques were considered due to their exceptional performance when applied to a wide range of problems; we assessed the CNN and LSTM for their added value by comparing their performance to that of some conventional forecasters. This research contributes to the maturation of the ongoing debate about the use of deep learning

techniques in load forecasting. We implemented forecasters that have been shown in the recent literature to be more adaptable to external factors such as annual increases in electricity demand or temperature shifts; we also implemented forecasters that can recognize complex data relationships without explicit user specifications [2], [46], [107], [132]. We evaluated the performance of all forecasters on three datasets, and because two of them are publicly available, this work is reproducible and will serve as a valuable benchmark for future research both within and outside of our smart-grid team.

### 5.3 Future Work

Several future research directions include increasing forecaster accuracy, incorporating more exogenous variables, developing more complex forecasters that are hybrids or improved models of the ones used in this work, forecasting the width of daily peaks, implementing more deep learning forecasters, developing separate models to forecast different days and months (e.g., summer, winter), and incorporating a holiday indicator as an input that specifies which days are holidays. Several of the points raised above will be discussed briefly.

In terms of forecaster accuracy, we recommend the following: by incorporating additional exogenous variables, such as those used by the MLR, the ANN, CNN, LSTM, and SARIMAX forecasters' accuracy can be improved. These variables include the hour of the day, the month, the weekend or holiday indicator, the previous day's maximum, minimum, and average demand, and the hourly lag from the previous week. Incorporating more lag values, such as those from the previous week, rather than just those from the previous day, can improve the performance of most forecasters in predicting Mondays,

Saturdays, and Sundays. Furthermore, depending on the analyst's objectives, weather variables such as humidity, dewpoint, and wind direction/speed can be used in addition to temperature.

Rather than using the ANNSTLF architecture, another approach for the CNN and LSTM that may improve their load forecasting performance is to implement them as temporal versions, which simply means feeding their models with the entire training datasets rather than the 79 inputs that were used. Additionally, hybrid models incorporating CNNs, and LSTMs may be an option, as some researchers have discovered that combining these two models improves performance, as discussed in the deep learning literature section. Tao Hong et al. [1] also stated in their review paper that the majority of the best load forecasting models had been discovered to be hybrid models.

When it comes to daily peaks, utilities benefit from knowing when they will occur and how long they will last. As a result, another approach for future work is determining the width of demand peaks. Furthermore, some researchers have observed improved performance when separate models are implemented to forecast specific days, such as weekdays and weekends, or specific months, such as winter and summer [48], [187]. Implementing unique models to forecast the summer months can improve load forecasting performance in the Toronto and Ottawa datasets, and unique models to forecast the winter months can improve load forecasting performance in the Saint John dataset.

As a result of the preceding paragraphs, we can see that there are numerous possibilities and that more research is required. These are intriguing paths that could be taken, and they can help utilities and power system operators in the future plan for and ensure a stable supply of electricity for everyone.

## Bibliography

- [1] T. Hong and S. Fan, “Probabilistic electric load forecasting: A tutorial review,” *Int. J. Forecast.*, vol. 32, no. 3, pp. 914–938, 2016, doi: 10.1016/j.ijforecast.2015.11.011.
- [2] K. Amarasinghe, D. L. Marino, and M. Manic, “Deep neural networks for energy load forecasting,” 2017, doi: 10.1109/ISIE.2017.8001465.
- [3] C. Kuster, Y. Rezgui, and M. Mourshed, “Electrical load forecasting models: A critical systematic review,” *Sustainable Cities and Society*. 2017, doi: 10.1016/j.scs.2017.08.009.
- [4] W. He, “Load Forecasting via Deep Neural Networks,” 2017, doi: 10.1016/j.procs.2017.11.374.
- [5] J. Zheng, C. Xu, Z. Zhang, and X. Li, “Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network,” 2017, doi: 10.1109/CISS.2017.7926112.
- [6] D. L. Marino, K. Amarasinghe, and M. Manic, “Building energy load forecasting using Deep Neural Networks,” *IECON Proc. (Industrial Electron. Conf.)*, pp. 7046–7051, 2016, doi: 10.1109/IECON.2016.7793413.
- [7] A. Almalaq and G. Edwards, “A review of deep learning methods applied on load forecasting,” *Proc. - 16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2017*, vol. 2017-Decem, pp. 511–516, 2017, doi: 10.1109/ICMLA.2017.0-110.
- [8] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network,” *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019, doi:

- 10.1109/TSG.2017.2753802.
- [9] S. Saurabh, H. Shoeb, A. B. Mohammad, S. Singh, S. Hussain, and M. A. Bazaz, “Short term load forecasting using artificial neural network,” in *2017 4th International Conference on Image Information Processing, ICIIP 2017*, 2018, pp. 159–163, doi: 10.1109/ICIIP.2017.8313703.
  - [10] J. Zhang, Y. M. Wei, D. Li, Z. Tan, and J. Zhou, “Short term electricity load forecasting using a hybrid model,” *Energy*, 2018, doi: 10.1016/j.energy.2018.06.012.
  - [11] A. Rahman, V. Srikumar, and A. D. Smith, “Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks,” *Appl. Energy*, 2018, doi: 10.1016/j.apenergy.2017.12.051.
  - [12] B. Yildiz, J. I. Bilbao, and A. B. Sproul, “A review and analysis of regression and machine learning models on commercial building electricity load forecasting,” *Renewable and Sustainable Energy Reviews*. 2017, doi: 10.1016/j.rser.2017.02.023.
  - [13] A. Baliyan, K. Gaurav, and S. Kumar Mishra, “A review of short term load forecasting using artificial neural network models,” 2015, doi: 10.1016/j.procs.2015.04.160.
  - [14] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. F. Adekoya, “Electricity load forecasting: a systematic review,” *J. Electr. Syst. Inf. Technol.*, 2020, doi: 10.1186/s43067-020-00021-8.
  - [15] E. Ela and B. Kirby, “ERCOT Event on February 26, 2008: Lessons Learned,” 2008, Accessed: Sep. 17, 2021. [Online]. Available: <http://www.osti.gov/bridge>.
  - [16] “Freak Blackouts Plunge Korea into Darkness - The Chosun Ilbo (English Edition):

- Daily News from Korea - national/politics > national,” 2011.  
[http://english.chosun.com/site/data/html\\_dir/2011/09/16/2011091600558.html](http://english.chosun.com/site/data/html_dir/2011/09/16/2011091600558.html)  
(accessed Sep. 17, 2021).
- [17] S. Khan, N. Javaid, A. Chand, A. B. M. Khan, F. Rashid, and I. U. Afridi, “Electricity Load Forecasting for Each Day of Week Using Deep CNN,” 2019, doi: 10.1007/978-3-030-15035-8\_107.
- [18] M. Baccouche, F. Mamalet, and C. Wolf, “(RGB)Sequential deep learning for human action recognition,” *Int. Work. Hum. Behav. Underst.*, 2011.
- [19] D. Yu, L. Deng, I. Jang, P. Kudumakis, M. Sandler, and K. Kang, “Deep learning and its applications to signal and information processing,” *IEEE Signal Process. Mag.*, 2011, doi: 10.1109/MSP.2010.939038.
- [20] M. Vos, C. Bender-Saebelkampf, and S. Albayrak, “Residential Short-Term Load Forecasting Using Convolutional Neural Networks,” 2018, doi: 10.1109/SmartGridComm.2018.8587494.
- [21] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural networks for short-term load forecasting: A review and evaluation,” *IEEE Trans. Power Syst.*, 2001, doi: 10.1109/59.910780.
- [22] R. Houimli, M. Zmami, and O. Ben-Salha, “Short-term electric load forecasting in Tunisia using artificial neural networks,” *Energy Syst.*, 2020, doi: 10.1007/s12667-019-00324-4.
- [23] D. C. Park, R. J. Marks, L. E. Atlas, and M. J. Damborg, “Electric load forecasting using an artificial neural network - Power Systems, IEEE Transactions on,” *IEEE Transactions Power Syst.*, 1991.

- [24] A. G. Bakirtzis, V. Petridis, S. J. Klartzis, M. C. Alexiadis, and A. H. Maassis, “A neural network short term load forecasting model for the greek power system,” *IEEE Trans. Power Syst.*, 1996, doi: 10.1109/59.496166.
- [25] L. C. P. Velasco, C. R. Villezas, P. N. C. Palahang, and J. A. A. Dagaang, “Next day electric load forecasting using Artificial Neural Networks,” 2016, doi: 10.1109/HNICEM.2015.7393166.
- [26] H. Hahn, S. Meyer-Nieberg, and S. Pickl, “Electric load forecasting methods: Tools for decision making,” *Eur. J. Oper. Res.*, 2009, doi: 10.1016/j.ejor.2009.01.062.
- [27] G. Gross and F. D. Galiana, “SHORT-TERM LOAD FORECASTING.,” *Proc. IEEE*, 1987, doi: 10.1109/PROC.1987.13927.
- [28] E. J. Wicksteed, “Short term electric load forecasting for British Columbia, Canada: an exploration of the use of numerical weather prediction data as a predictor in an artificial neural network,” University of British Columbia, 2021.
- [29] A. Muñoz, E. F. Sánchez-Úbeda, A. Cruz, and J. Marín, “Short-term Forecasting in Power Systems: A Guided Tour,” 2010.
- [30] D. Srinivasan and M. A. Lee, “Survey of hybrid fuzzy neural approaches to electric load forecasting,” 1995, doi: 10.1109/icsmc.1995.538416.
- [31] C. N. Lu, H. T. Wu, and S. Vemuri, “Neural Network Based Short Term Load Forecasting,” *IEEE Trans. Power Syst.*, 1993, doi: 10.1109/59.221223.
- [32] T. Hong, “Short Term Electric Load Forecasting dissertation,” 3442639, 2010.
- [33] J. Foster, “Electric load forecasting with increased embedded renewable generation,” Queen’s University, 2020.
- [34] T. Hong and M. Shahidehpour, “Load Forecasting Case Study,” *U.S. Dep. Energy*,

2015.

- [35] E. Taylor, “Short-term Electrical Load Forecasting for an Institutional/Industrial Power System Using an Artificial Neural Network,” The University of Tennessee, Knoxville, 2013.
- [36] J. W. Taylor and R. Buizza, “Neural network load forecasting with weather ensemble predictions,” *IEEE Trans. Power Syst.*, 2002, doi: 10.1109/TPWRS.2002.800906.
- [37] A. Khotanzad, R. Afkhami-Rohani, and R. Af, “ANNSTLF - Artificial neural network short-term load forecaster - generation three,” *IEEE Trans. Power Syst.*, vol. 13, no. 4, pp. 1413–1422, 1998, doi: 10.1109/59.736285.
- [38] M. Sobhani, A. Campbell, S. Sangamwar, C. Li, and T. Hong, “Combining weather stations for electric load forecasting,” *Energies*, 2019, doi: 10.3390/en12081510.
- [39] T. Hong, P. Wang, and L. White, “Weather station selection for electric load forecasting,” *Int. J. Forecast.*, 2015, doi: 10.1016/j.ijforecast.2014.07.001.
- [40] S. N. Fallah, M. Ganjkhani, S. Shamshirband, and K. wing Chau, “Computational intelligence on short-term load forecasting: A methodological overview,” *Energies*. 2019, doi: 10.3390/en12030393.
- [41] S. Moreno-Carbonell, E. F. Sánchez-Úbeda, and A. Muñoz, “Rethinking weather station selection for electric load forecasting using genetic algorithms,” *Int. J. Forecast.*, 2020, doi: 10.1016/j.ijforecast.2019.08.008.
- [42] S. Fan, K. Methaprayoon, and W. J. Lee, “Multi-area load forecasting for system with large geographical area,” 2008, doi: 10.1109/ICPS.2008.4606287.
- [43] M. JANICKI, “Methods of weather variables introduction into short-term electric

- load forecasting models - a review," *PRZEGŁĄD ELEKTROTECHNICZNY*, 2017, doi: 10.15199/48.2017.04.18.
- [44] L. Friedrich and A. Afshari, "Short-term Forecasting of the Abu Dhabi Electricity Load Using Multiple Weather Variables," 2015, doi: 10.1016/j.egypro.2015.07.616.
  - [45] E. L. Taylor, "Short-term Electrical Load Forecasting for an Institutional/ Industrial Power System Using an Artificial Neural Network," University of Tennessee, 2013.
  - [46] Z. Deng, B. Wang, Y. Xu, T. Xu, C. Liu, and Z. Zhu, "Multi-scale convolutional neural network with time-cognition for multi-step short-Term load forecasting," *IEEE Access*, vol. 7, pp. 88058–88071, 2019, doi: 10.1109/ACCESS.2019.2926137.
  - [47] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 456–462, 2014, doi: 10.1109/TSG.2013.2274373.
  - [48] P. Mandal, T. Senju, N. Urasaki, and T. Funabashi, "A neural network based several-hour-ahead electric load forecasting using similar days approach," *Int. J. Electr. Power Energy Syst.*, 2006, doi: 10.1016/j.ijepes.2005.12.007.
  - [49] J. Luo, T. Hong, and M. Yue, "Real-time anomaly detection for very short-term load forecasting," *J. Mod. Power Syst. Clean Energy*, 2018, doi: 10.1007/s40565-017-0351-7.
  - [50] K. Liu, "Comparison of very short-term load forecasting techniques," *IEEE Trans. Power Syst.*, 1996, doi: 10.1109/59.496169.
  - [51] W. Charytoniuk and M. S. Chen, "Very short-term load forecasting using artificial neural networks," *IEEE Trans. Power Syst.*, 2000, doi: 10.1109/59.852131.
  - [52] J. W. Taylor, "An evaluation of methods for very short-term load forecasting using

- minute-by-minute British data,” *Int. J. Forecast.*, 2008, doi: 10.1016/j.ijforecast.2008.07.007.
- [53] E. Kyriakides and M. Polycarpou, “Short term electric load forecasting: A tutorial,” *Stud. Comput. Intell.*, 2006, doi: 10.1007/978-3-540-36122-0\_16.
- [54] Ö. Ö. Bozkurt, G. Biricik, and Z. C. Taysi, “Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market Ö,” *PLoS One*, 2017, doi: 10.1371/journal.pone.0175915.
- [55] S. Dwijayanti, “Short Term Load Forecasting Using a Neural Network Based Time Series Approach,” Oklahoma State University, 2013.
- [56] G. J. Tsekouras, N. D. Hatziargyriou, and E. N. Dialynas, “An optimized adaptive neural network for annual midterm energy forecasting,” *IEEE Trans. Power Syst.*, 2006, doi: 10.1109/TPWRS.2005.860926.
- [57] E. Doveh, P. Feigin, D. Greig, and L. Hyams, “Experience with FNN models for medium term power demand predictions,” *IEEE Trans. Power Syst.*, 1999, doi: 10.1109/59.761878.
- [58] J. Reneses, E. Centeno, and J. Barquín, “Coordination between medium-term generation planning and short-term operation in electricity markets,” *IEEE Trans. Power Syst.*, 2006, doi: 10.1109/TPWRS.2005.857851.
- [59] M. S. Kandil, S. M. El-Debeiky, and N. E. Hasanien, “Long-term load forecasting for fast developing utility using a knowledge-based expert system,” *IEEE Trans. Power Syst.*, 2002, doi: 10.1109/TPWRS.2002.1007923.
- [60] T. Hong, P. Wang, and H. L. Willis, “A naïve multiple linear regression benchmark for short term load forecasting,” 2011, doi: 10.1109/PES.2011.6038881.

- [61] K. Methaprayoon, W. J. Lee, S. Rasmiddatta, J. R. Liao, and R. J. Ross, “Multistage artificial neural network short-term load forecasting engine with front-end weather forecast,” *IEEE Trans. Ind. Appl.*, 2007, doi: 10.1109/TIA.2007.908190.
- [62] A. K. Singh, Ibraheem, S. Khatoon, M. Muazzam, and D. K. Chaturvedi, “Load forecasting techniques and methodologies: A review,” 2012, doi: 10.1109/ICPCES.2012.6508132.
- [63] S. Kumar, S. Mishra, and S. Gupta, “Short term load forecasting using ANN and multiple linear regression,” 2016, doi: 10.1109/CICT.2016.44.
- [64] A. Y. Saber and A. K. M. R. Alam, “Short term load forecasting using multiple linear regression for big data,” *2017 IEEE Symp. Ser. Comput. Intell. SSCI 2017 - Proc.*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/SSCI.2017.8285261.
- [65] L. Tang, Y. Yi, and Y. Peng, “An ensemble deep learning model for short-term load forecasting based on ARIMA and LSTM,” 2019, doi: 10.1109/SmartGridComm.2019.8909756.
- [66] B. Nepal, M. Yamaha, A. Yokoe, and T. Yamaji, “Electricity load forecasting using clustering and ARIMA model for energy management in buildings,” *Japan Archit. Rev.*, 2020, doi: 10.1002/2475-8876.12135.
- [67] A. Badri, Z. Ameli, and A. Motie Birjandi, “Application of artificial neural networks and fuzzy logic methods for short term load forecasting,” 2012, doi: 10.1016/j.egypro.2011.12.965.
- [68] P. H. Kuo and C. J. Huang, “A high precision artificial neural networks model for short-Term energy load forecasting,” *Energies*, 2018, doi: 10.3390/en11010213.
- [69] S. Humeau, T. K. Wijaya, M. Vasirani, and K. Aberer, “Electricity load forecasting

- for residential customers: Exploiting aggregation and correlation between households,” 2013, doi: 10.1109/SustainIT.2013.6685208.
- [70] G. Dudek, “Pattern-based local linear regression models for short-term load forecasting,” *Electr. Power Syst. Res.*, 2016, doi: 10.1016/j.epsr.2015.09.001.
- [71] N. Amjady, “Short-term hourly load forecasting using time-series modeling with peak load estimation capability,” *IEEE Trans. Power Syst.*, vol. 16, no. 4, pp. 798–805, 2001, doi: 10.1109/59.962429.
- [72] A. Bracale, G. Carpinelli, P. De Falco, and T. Hong, “Short-term industrial load forecasting: A case study in an Italian factory,” 2017, doi: 10.1109/ISGTEurope.2017.8260176.
- [73] P. Wang, B. Liu, and T. Hong, “Electric load forecasting with recency effect: A big data approach,” *Int. J. Forecast.*, 2016, doi: 10.1016/j.ijforecast.2015.09.006.
- [74] Y. Cao, M. Raoof, S. Montgomery, J. Ottosson, and I. Näslund, “Predicting Long-Term Health-Related Quality of Life after Bariatric Surgery Using a Conventional Neural Network: A Study Based on the Scandinavian Obesity Surgery Registry,” *J. Clin. Med.*, 2019, doi: 10.3390/jcm8122149.
- [75] Y. Wang, N. Zhang, Y. Tan, T. Hong, D. S. Kirschen, and C. Kang, “Combining Probabilistic Load Forecasts,” *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3664–3674, 2019, doi: 10.1109/TSG.2018.2833869.
- [76] G. Papacharalampous, H. Tyralis, and D. Koutsoyiannis, “Predictability of monthly temperature and precipitation using automatic time series forecasting methods,” *Acta Geophys.*, 2018, doi: 10.1007/s11600-018-0120-7.
- [77] M. Rana and I. Koprinska, “Forecasting electricity load with advanced wavelet

- neural networks,” *Neurocomputing*, 2016, doi: 10.1016/j.neucom.2015.12.004.
- [78] Da Liu, K. Sun, H. Huang, and P. Tang, “Monthly load forecasting based on economic data by decomposition integration theory,” *Sustain.*, 2018, doi: 10.3390/su10093282.
- [79] T. Hong, M. Gui, M. E. Baran, and H. L. Willis, “Modeling and forecasting hourly electric load by multiple linear regression with interactions,” *IEEE PES Gen. Meet. PES 2010*, pp. 1–8, 2010, doi: 10.1109/PES.2010.5589959.
- [80] T. Hong and P. Wang, “Fuzzy interaction regression for short term load forecasting,” *Fuzzy Optim. Decis. Mak.*, 2014, doi: 10.1007/s10700-013-9166-9.
- [81] M. Abuella and B. Chowdhury, “Solar power probabilistic forecasting by using multiple linear regression analysis,” 2015, doi: 10.1109/SECON.2015.7132869.
- [82] K. Panklib, C. Prakasvudhisarn, and D. Khummongkol, “Electricity Consumption Forecasting in Thailand Using an Artificial Neural Network and Multiple Linear Regression,” *Energy Sources, Part B Econ. Plan. Policy*, 2015, doi: 10.1080/15567249.2011.559520.
- [83] X. Sun, Z. Ouyang, and D. Yue, “Short-term load forecasting based on multivariate linear regression,” 2017, doi: 10.1109/EI2.2017.8245401.
- [84] R. Weron, *Modeling and forecasting electricity loads and prices: A statistical approach*. wiley, 2006.
- [85] N. Amral, C. S. Özveren, and D. King, “Short term load forecasting using multiple linear regression,” 2007, doi: 10.1109/UPEC.2007.4469121.
- [86] T. Hong, “Short Term Electric Load Forecasting,” North Carolina State University, 2010.

- [87] A. D. Papalexopoulos and T. C. Hesterberg, “A regression-based approach to short-term system load forecasting,” *IEEE Trans. Power Syst.*, 1990, doi: 10.1109/59.99410.
- [88] M. Cai, M. Pipattanasomporn, and S. Rahman, “Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques,” *Appl. Energy*, 2019, doi: 10.1016/j.apenergy.2018.12.042.
- [89] E. Stellwagen and L. Tashman, “ARIMA : The Models of Box and Jenkins,” *Foresight Int. J. Appl. Forecast.*, 2013.
- [90] K. Goswami, A. Ganguly, and A. K. Sil, “Day ahead forecasting and peak load management using multivariate auto regression technique,” *Proc. 2018 IEEE Appl. Signal Process. Conf. ASPCON 2018*, no. 1, pp. 279–282, 2018, doi: 10.1109/ASPCON.2018.8748661.
- [91] G. N. Shilpa and G. S. Sheshadri, “ARIMAX Model for Short-Term Electrical Load Forecasting,” *Int. J. Recent Technol. Eng.*, 2019, doi: 10.35940/ijrte.d7950.118419.
- [92] H. Cui and X. Peng, “Short-Term City Electric Load Forecasting with Considering Temperature Effects: An Improved ARIMAX Model,” *Math. Probl. Eng.*, 2015, doi: 10.1155/2015/589374.
- [93] A. Shadkam, “Using SARIMAX to forecast electricity demand and consumption in university buildings,” The University of British Columbia, 2020.
- [94] I. Fernández, C. E. Borges, and Y. K. Penya, “Efficient building load forecasting,” 2011, doi: 10.1109/ETFA.2011.6059103.
- [95] R. Bonetto and M. Rossi, “Parallel multi-step ahead power demand forecasting through NAR neural networks,” *2016 IEEE Int. Conf. Smart Grid Commun.*

*SmartGridComm* 2016, pp. 314–319, Dec. 2016, doi: 10.1109/SmartGridComm.2016.7778780.

- [96] M. Cools, E. Moons, and G. Wets, “Investigating the Variability in Daily Traffic Counts through use of ARIMAX and SARIMAX Models: Assessing the Effect of Holidays on Two Site Locations,” <https://doi.org/10.3141/2136-07>, 2009.
- [97] G. Papaioannou, C. Dikaiakos, A. Dramountanis, and P. Papaioannou, “Analysis and Modeling for Short- to Medium-Term Load Forecasting Using a Hybrid Manifold Learning Principal Component Model and Comparison with Classical Statistical Models (SARIMAX, Exponential Smoothing) and Artificial Intelligence Models (ANN, SVM): Th,” *Energies*, 2016, doi: 10.3390/en9080635.
- [98] M. De Felice, A. Alessandri, and P. M. Ruti, “Electricity demand forecasting over Italy: Potential benefits using numerical weather prediction models,” *Electr. Power Syst. Res.*, 2013, doi: 10.1016/j.epsr.2013.06.004.
- [99] A. Khotanzad, R. C. Hwang, A. Abaye, and D. Maratukulam, “An Adaptive Modular Artificial Neural Network Hourly Load Forecaster and its Implementation at Electric Utilities,” *IEEE Trans. Power Syst.*, 1995, doi: 10.1109/59.466468.
- [100] A. Khotanzad, R. Afkhami-Rohani, T. L. Lu, A. Abaye, M. Davis, and D. J. Maratukulam, “ANNSTLF - A neural-network-based electric load forecasting system,” *IEEE Trans. Neural Networks*, 1997, doi: 10.1109/72.595881.
- [101] “Recursive least squares filter - Wikipedia.” [https://en.wikipedia.org/wiki/Recursive\\_least\\_squares\\_filter](https://en.wikipedia.org/wiki/Recursive_least_squares_filter) (accessed Oct. 08, 2021).
- [102] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous

- activity,” *Bull. Math. Biophys.*, 1943, doi: 10.1007/BF02478259.
- [103] D. O. Hebb, “The first stage of perception: growth of the assembly,” *Organ. Behav.*, 1949, doi: 10.1016/0301-0082(84)90021-2.
- [104] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychol. Rev.*, 1958, doi: 10.1037/h0042519.
- [105] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, 1986, doi: 10.1038/323533a0.
- [106] X. H. Le, H. V. Ho, G. Lee, and S. Jung, “Application of Long Short-Term Memory (LSTM) neural network for flood forecasting,” *Water (Switzerland)*, 2019, doi: 10.3390/w11071387.
- [107] M. Munem, T. M. Rubaith Bashar, M. H. Roni, M. Shahriar, T. B. Shawkat, and H. Rahaman, “Electric power load forecasting based on multivariate LSTM neural network using bayesian optimization,” *2020 IEEE Electr. Power Energy Conf. EPEC 2020*, vol. 3, 2020, doi: 10.1109/EPEC48502.2020.9320123.
- [108] V. Dehalwar, A. Kalam, M. L. Kolhe, and A. Zayegh, “Electricity load forecasting for urban area using weather forecast information,” *2016 IEEE Int. Conf. Power Renew. Energy, ICPRE 2016*, pp. 355–359, 2017, doi: 10.1109/ICPRE.2016.7871231.
- [109] A. Si. Walia, “Activation functions and it’s types-Which is better?,” *Towards Data Science*, 2017..
- [110] A. Khotanzad, E. Zhou, and H. Elragal, “A neuro-fuzzy approach to short-term load forecasting in a price-sensitive environment,” *IEEE Trans. Power Syst.*, vol. 17, no. 4, pp. 1273–1282, Nov. 2002, doi: 10.1109/TPWRS.2002.804999.

- [111] P. R. J. Campbell and K. Adamson, “Methodologies for load forecasting,” 2006, doi: 10.1109/IS.2006.348523.
- [112] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox<sup>TM</sup> 7 User’s Guide*. 2010.
- [113] B. F. Hobbs, “Analysis of the value for unit commitment of improved load forecasts,” *IEEE Trans. Power Syst.*, 1999, doi: 10.1109/59.801894.
- [114] M. Buhari and S. S. Adamu, “Short-term load forecasting using artificial neural network,” 2012, doi: 10.1109/icit.2000.854220.
- [115] A. Khotanzad, E. Zhou, and H. Elragal, “A Neuro-Fuzzy Approach to Short-Term Load Forecasting in a Price-Sensitive Environment,” *IEEE Power Eng. Rev.*, 2008, doi: 10.1109/mper.2002.4312570.
- [116] A. Webberley and D. W. Gao, “Study of artificial neural network based short term load forecasting,” 2013, doi: 10.1109/PESMG.2013.6673036.
- [117] E. A. Feinberg and D. Genethliou, “Load Forecasting,” in *Applied Mathematics for Restructured Electric Power Systems*, 2006.
- [118] D. W. Bunn, “Forecasting loads and prices in competitive power markets,” *Proc. IEEE*, 2000, doi: 10.1109/5.823996.
- [119] A. D. Papalexopoulos, S. Hao, and T. M. Peng, “An implementation of a neural network based load forecasting model for the EMS,” *IEEE Trans. Power Syst.*, 1994, doi: 10.1109/59.331456.
- [120] Zhang, G., E. Patuwo, and M. Y. Hu, “Forecasting with Artificial neural networks,” *Int. J. Forecast.*, 1998.
- [121] G. H. Yann LeCun, Yoshua Bengio, “Deep learning (2015), Y. LeCun, Y. Bengio

- and G. Hinton,” *Nature*, 2015.
- [122] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, 2006, doi: 10.1162/neco.2006.18.7.1527.
- [123] S. Suresh, “An Analysis of Short-term Load Forecasting on Residential Buildings Using Deep Learning Models,” Virginia Polytechnic Institute and State University, Blacksburg, 2020.
- [124] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” 2007, doi: 10.7551/mitpress/7503.003.0024.
- [125] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2015.
- [126] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” 2013, doi: 10.1109/ICASSP.2013.6638947.
- [127] H. Shi, M. Xu, and R. Li, “Deep Learning for Household Load Forecasting-A Novel Pooling Deep RNN,” *IEEE Trans. Smart Grid*, 2018, doi: 10.1109/TSG.2017.2686012.
- [128] D. Silver, J. Schrittwieser, K. Simonyan, I. A.- Nature, and U. 2017, “Mastering the game of Go without human knowledge,” *Nature*. 2016.
- [129] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, 2015, doi: 10.1038/nature14236.
- [130] C. J. Huang and P. H. Kuo, “Multiple-Input Deep Convolutional Neural Network Model for Short-Term Photovoltaic Power Forecasting,” *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2921238.
- [131] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, “Optimal deep learning LSTM

model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches,” *Energies*, 2018, doi: 10.3390/en11071636.

- [132] H. J. Sadaei, P. C. de Lima e Silva, F. G. Guimarães, and M. H. Lee, “Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series,” *Energy*, 2019, doi: 10.1016/j.energy.2019.03.081.
- [133] I. Koprinska, D. Wu, and Z. Wang, “Convolutional Neural Networks for Energy Time Series Forecasting,” 2018, doi: 10.1109/IJCNN.2018.8489399.
- [134] A. Gasparin, S. Lukovic, and C. Alippi, “Deep Learning for Time Series Forecasting: The Electric Load Case,” 2019, [Online]. Available: <http://arxiv.org/abs/1907.09207>.
- [135] C. Tian, J. Ma, C. Zhang, and P. Zhan, “A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network,” *Energies*, 2018, doi: 10.3390/en11123493.
- [136] B. Farsi, “On Short-Term Load Forecasting Using Machine Learning Techniques,” Concordia University, 2020.
- [137] C. J. Huang, Y. Shen, Y. H. Chen, and H. C. Chen, “A novel hybrid deep neural network model for short-term electricity price forecasting,” *Int. J. Energy Res.*, 2021, doi: 10.1002/er.5945.
- [138] B. Y. Goodfellow I., “Courville A-Deep learning-MIT (2016),” *Nature*, 2016.
- [139] C. Gallicchio, A. Micheli, and L. Pedrelli, “Design of deep echo state networks,” *Neural Networks*, 2018, doi: 10.1016/j.neunet.2018.08.002.
- [140] “Understanding LSTM Networks -- colah’s blog.”

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed Dec. 08, 2021).

- [141] “Introduction to LSTM Units in RNN | Pluralsight.”  
<https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn> (accessed Nov. 18, 2021).
- [142] P. P. Phyo, “Deep Learning for Short-term Electricity Load Forecasting,” Sirindhorn International Institute of Technology, 2018.
- [143] C. Olah, “Understanding LSTM Networks [Blog],” *Web Page*, 2015.
- [144] M. Imani and H. Ghassemian, “Sequence to Image Transform Based Convolutional Neural Network for Load Forecasting,” 2019, doi: 10.1109/IranianCEE.2019.8786456.
- [145] R. Garg, B. G. Vijay Kumar, G. Carneiro, and I. Reid, “Unsupervised CNN for single view depth estimation: Geometry to the rescue,” 2016, doi: 10.1007/978-3-319-46484-8\_45.
- [146] T. T. Um, V. Babakeshizadeh, and D. Kulic, “Exercise motion classification from large-scale wearable sensor data using convolutional neural networks,” 2017, doi: 10.1109/IROS.2017.8206051.
- [147] Y. Zhang, S. Roller, and B. C. Wallace, “MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification,” 2016, doi: 10.18653/v1/n16-1178.
- [148] E. Gawehn, J. A. Hiss, and G. Schneider, “Deep Learning in Drug Discovery,” *Molecular Informatics*. 2016, doi: 10.1002/minf.201501008.
- [149] C. Lang, “Machine Learning Approaches for Energy Forecasting,” University of

Regensburg, Regensburg, 2021.

- [150] N. Singh, C. Vyjayanthi, and C. Modi, “Multi-step Short-term Electric Load Forecasting using 2D Convolutional Neural Networks,” 2020, doi: 10.1109/HYDCON48903.2020.9242917.
- [151] R. Fukuoka, H. Suzuki, T. Kitajima, A. Kuwahara, and T. Yasuno, “Wind Speed Prediction Model Using LSTM and 1D-CNN,” *J. Signal Process.*, 2018, doi: 10.2299/jsp.22.207.
- [152] A. Brunel *et al.*, “A CNN adapted to time series for the classification of Supernovae,” 2019, doi: 10.2352/ISSN.2470-1173.2019.14.COLOR-090.
- [153] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” 2012.
- [154] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016, doi: 10.1109/CVPR.2016.90.
- [155] C. L. Liu, F. Yin, Q. F. Wang, and D. H. Wang, “ICDAR 2011 Chinese handwriting recognition competition,” 2011, doi: 10.1109/ICDAR.2011.291.
- [156] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” 2012.
- [157] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Mitosis detection in breast cancer histology images with deep neural networks,” 2013, doi: 10.1007/978-3-642-40763-5\_51.
- [158] G. E. Dahl, M. Ranzato, A. R. Mohamed, and G. Hinton, “Phone recognition with the mean-covariance restricted Boltzmann machine,” 2010.
- [159] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using Context-

- Dependent Deep Neural Networks,” 2011, doi: 10.21437/interspeech.2011-169.
- [160] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” *IEEE Trans. Audio, Speech Lang. Process.*, 2014, doi: 10.1109/TASLP.2014.2339736.
- [161] L. Deng and J. C. Platt, “Ensemble deep learning for speech recognition,” 2014, doi: 10.21437/interspeech.2014-433.
- [162] Y. Shen, Y. Ma, S. Deng, C. J. Huang, and P. H. Kuo, “An ensemble model based on deep learning and data preprocessing for short-term electrical load forecasting,” *Sustain.*, 2021, doi: 10.3390/su13041694.
- [163] S. He *et al.*, “A per-unit curve rotated decoupling method for CNN-TCN based day-ahead load forecasting,” *IET Gener. Transm. Distrib.*, 2021, doi: 10.1049/gtd2.12214.
- [164] Y. Wang *et al.*, “Short-term load forecasting for industrial customers based on TCN-LightGBM,” *IEEE Trans. Power Syst.*, 2021, doi: 10.1109/TPWRS.2020.3028133.
- [165] “Stock Price Time Series Forecasting using Deep CNN.”  
<https://www.analyticsvidhya.com/blog/2021/08/hands-on-stock-price-time-series-forecasting-using-deep-convolutional-networks/> (accessed Dec. 22, 2021).
- [166] “Dispelling the Myth: How Peak Demand REALLY Occurs | Energy Sentry News.”  
<https://energysentry.com/newsletters/disPELLING-myth.php> (accessed Oct. 24, 2021).
- [167] “Peak Load & Base Electricity - Understand Differences - EnergyWatch.”  
<https://energywatch-inc.com/peak-load-base-electricity/> (accessed Oct. 07, 2021).
- [168] “Base Load and Peak Load: understanding both concepts.”

<https://sinovoltaics.com/learning-center/basics/base-load-peak-load/> (accessed Oct. 24, 2021).

[169] “What is Peak Load? | Aquicore.” <https://aonicore.com/blog/what-is-peak-load/> (accessed Oct. 07, 2021).

[170] X. Kong, C. Li, C. Wang, Y. Zhang, and J. Zhang, “Short-term electrical load forecasting based on error correction using dynamic mode decomposition,” *Appl. Energy*, 2020, doi: 10.1016/j.apenergy.2019.114368.

[171] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, “Deep belief network based electricity load forecasting: An analysis of Macedonian case,” *Energy*, 2016, doi: 10.1016/j.energy.2016.07.090.

[172] S. Papadopoulos and I. Karakatsanis, “Short-term electricity load forecasting using time series and ensemble learning methods,” 2015, doi: 10.1109/PECI.2015.7064913.

[173] W. Kim, Y. Han, K. J. Kim, and K. W. Song, “Electricity load forecasting using advanced feature selection and optimal deep learning model for the variable refrigerant flow systems,” *Energy Reports*, 2020, doi: 10.1016/j.egyr.2020.09.019.

[174] “Independent Electricity System Operator - Hourly Zonal Demand Report.” <http://reports.ieso.ca/public/DemandZonal/> (accessed Jun. 05, 2021).

[175] “Historical Climate Data - Climate - Environment and Climate Change Canada.” <https://climate.weather.gc.ca/> (accessed Jan. 05, 2021).

[176] D. C. Wu, B. Bahrami Asl, A. Razban, and J. Chen, “Air compressor load forecasting using artificial neural network,” *Expert Syst. Appl.*, 2021, doi: 10.1016/j.eswa.2020.114209.

- [177] L. Kuan *et al.*, “Short-term electricity load forecasting method based on multilayered self-normalizing GRU network,” 2017, doi: 10.1109/EI2.2017.8245330.
- [178] L. Li, K. Ota, and M. Dong, “Everything is image: CNN-based short-term electrical load forecasting for smart grid,” 2017, doi: 10.1109/ISPAN-FCST-ISCC.2017.78.
- [179] M. Dong and L. Grumbach, “A Hybrid Distribution Feeder Long-Term Load Forecasting Method Based on Sequence Prediction,” *IEEE Trans. Smart Grid*, 2020, doi: 10.1109/TSG.2019.2924183.
- [180] L. Yin and J. Xie, “Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems,” *Appl. Energy*, 2021, doi: 10.1016/j.apenergy.2020.116328.
- [181] S. Panigrahi, Y. Karali, and H. S. Behera, “Normalize Time Series and Forecast using Evolutionary Neural Network,” *Int. J. Comput. Appl.*, 2013.
- [182] “Fit linear regression model - MATLAB fitlm.”  
<https://www.mathworks.com/help/stats/fitlm.html> (accessed Nov. 21, 2021).
- [183] “Resilient Back Propagation - Wikipedia.” <https://en.wikipedia.org/wiki/Rprop> (accessed Dec. 31, 2021).
- [184] S. Krishnan, Y. Xiao, and R. A. Saurous, “Neumann optimizer: A practical optimization algorithm for deep neural networks,” 2018.
- [185] C. Chen, L. Shen, F. Zou, and F. Wei Liu, “Towards Practical Adam: Non-Convexity, Convergence Theory, and Mini-Batch Acceleration,” Jan. 2021, Accessed: Jan. 16, 2022. [Online]. Available: <https://arxiv.org/abs/2101.05471v1>.
- [186] O. Yadan, K. Adams, Y. Taigman, and M. Ranzato, “Multi-GPU training of

convnets,” 2014.

- [187] M. Barman and N. B. Dev Choudhury, “Season specific approach for short-term load forecasting based on hybrid FA-SVM and similarity concept,” *Energy*, 2019, doi: 10.1016/j.energy.2019.03.010.

## Appendix A

### 1 Determining the SARIMAX Model's Optimal Parameters

Autocorrelation is also known as serial correlation in discrete-time. As a function of delay, it correlates a signal with a delayed copy of itself. Informally, it refers to the similarity of observations as a function of time lag. The auto correlation function (ACF) can be used to identify the most common seasonality in a time series. It also aids us in determining the optimal number of differences and identifying instances where our series may be over or under-differentiated. The proper order of differencing is the smallest number that can produce a near-stationary series that oscillates around a defined mean, and the auto-correlation plot quickly approaches zero. Finally, ACF helps us determine the lags that can be used in our MA terms for seasonal and non-seasonal components.

Dataset	(p, d, q) x (P, D, Q)s	Exact Lag Vectors - (p, d, q) x (P, D, Q)
Toronto	(1, 1, 2) x (7, 1, 7) <sub>24</sub>	(1, 1, [1, 2]) x ([24, 48, 72, 168], 24, [24, 168])
Ottawa	(1, 1, 1) x (2, 1, 7) <sub>24</sub>	(1, 1, 1) x ([24, 48], 24, [24, 168])
Saint John	(2, 1, 1) x (7, 1, 2) <sub>24</sub>	([1, 2], 1, 1) x ([24, 168], 24, [24, 48])

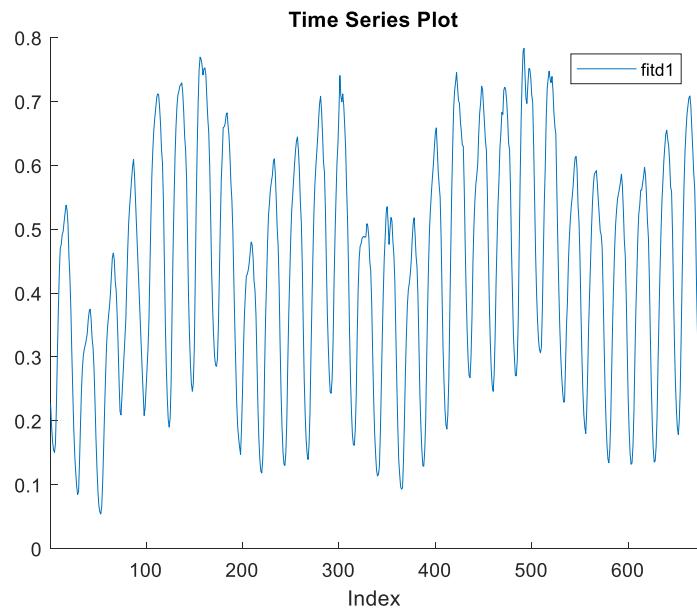
**Table 16 - The SARIMAX hyperparameters that were used across all datasets**

The parameters used for the SARIMAX model for each dataset are shown in Table 16. The partial autocorrelation function (PACF) in time series analysis calculates the partial correlation of a stationary time series with its own lagged values by regressing the time series' values at all shorter lags. After removing intermediate lags, the correlation between a time series and its lag is known as partial autocorrelation. Thus, partial autocorrelation captures the unambiguous relationship between a lag and a series. The PACF helps us

determine the lags that could be used for the seasonal and non-seasonal components of the AR terms.

### 1.1 Statistical Analysis of the Toronto Dataset

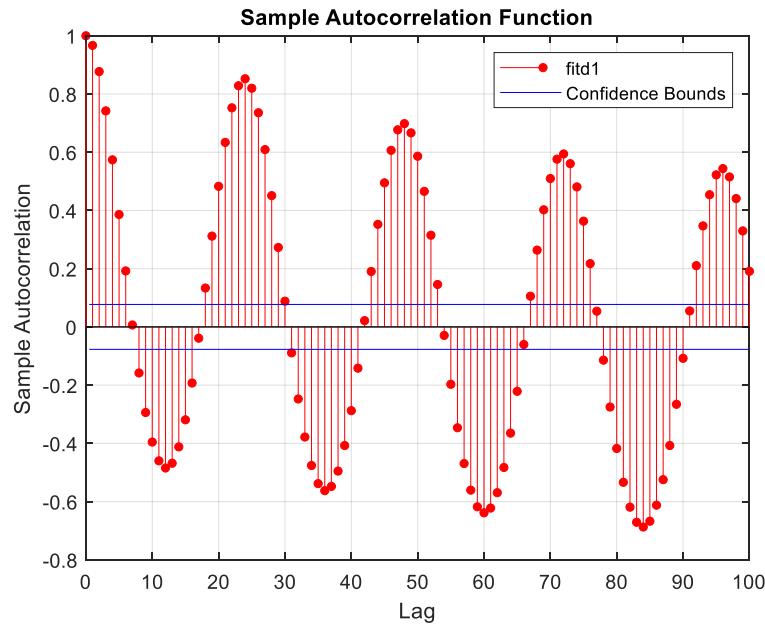
The seasonal patterns in the load demand time series from the Toronto dataset could be exploited using statistical techniques such as the SARIMAX. Figure 26 depicts hourly load data spanning 28 days, with two major seasonalities identified: daily and weekly. Segments such as the one depicted in this figure are good candidates for estimating the best parameters for the SARIMAX model; the segment depicted here was used in the case of the Toronto dataset.



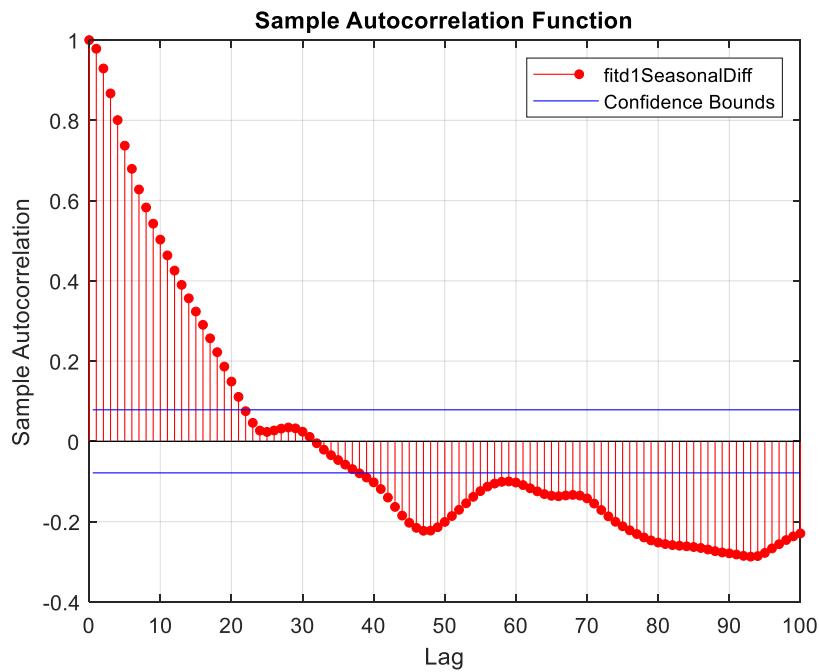
**Figure 26 – Excerpt from the Toronto Dataset**

Figure 27 depicts the initial autocorrelation function with no differencing. We can see a high degree of seasonality on a 24-hour basis, which is why we chose 24 as the seasonal difference denoted by parameter S. As illustrated in Figure 28, the ACF plot approaches

zero very slowly after seasonal differencing. This implies that we would need more differentiation, which the non-seasonal component can provide.

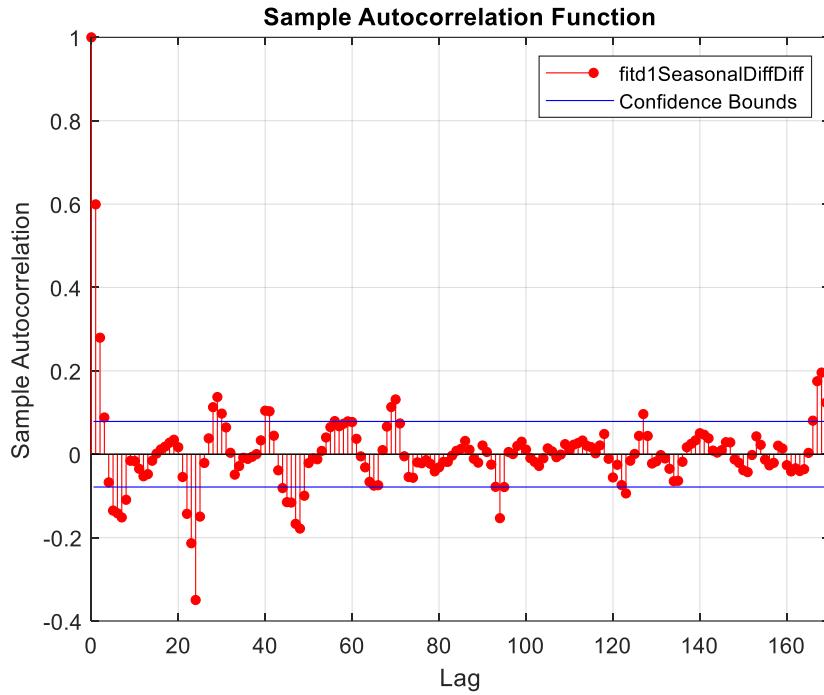


**Figure 27 – Plot of the Initial Auto Correlation – Toronto Dataset**



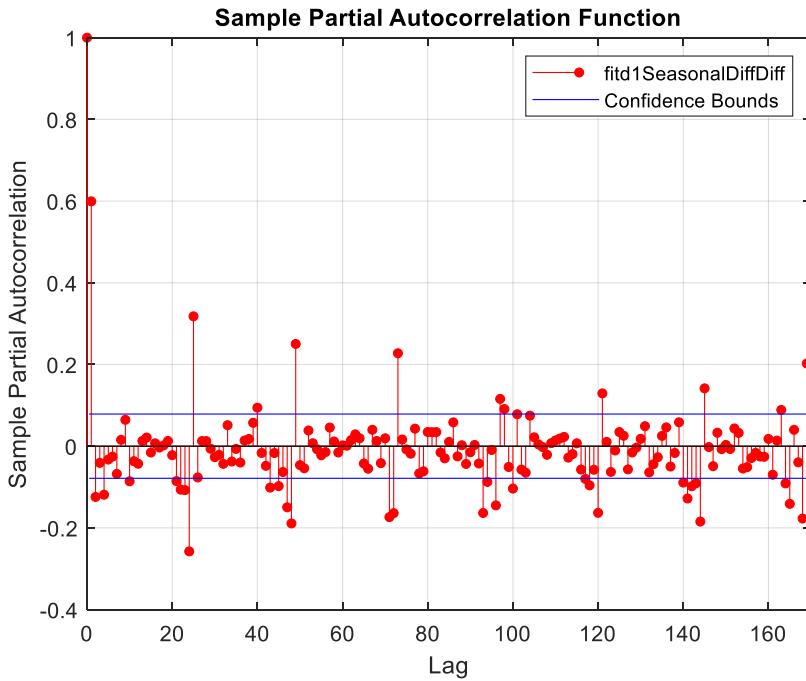
**Figure 28 – ACF Plot Following Seasonal Differencing – Toronto Dataset**

As shown in Figure 29, the new ACF plot approaches zero fairly quickly, indicating that nonseasonal differencing  $d$  can be set to 1. The first and second lags are highly significant, as shown by the same figure; we set  $q$  to 1 and 2. The big  $Q$  values were set to 24 and 168 because these are the points with the highest auto correlation.



**Figure 29 – ACF Plot After Seasonal and Non-Seasonal Differencing – Toronto Dataset**

The values of our small  $p$  and large  $P$  lags can be obtained from the PACF plot in Figure 30. We can see a significant lag at 1 that is greater than the subsequent ones, so we can set  $p$  to 1. There are significant lags at other points as well, including 24, 48, 72, and 168; however, in order to keep things simple, we chose only the points listed above to serve as  $Q$ .



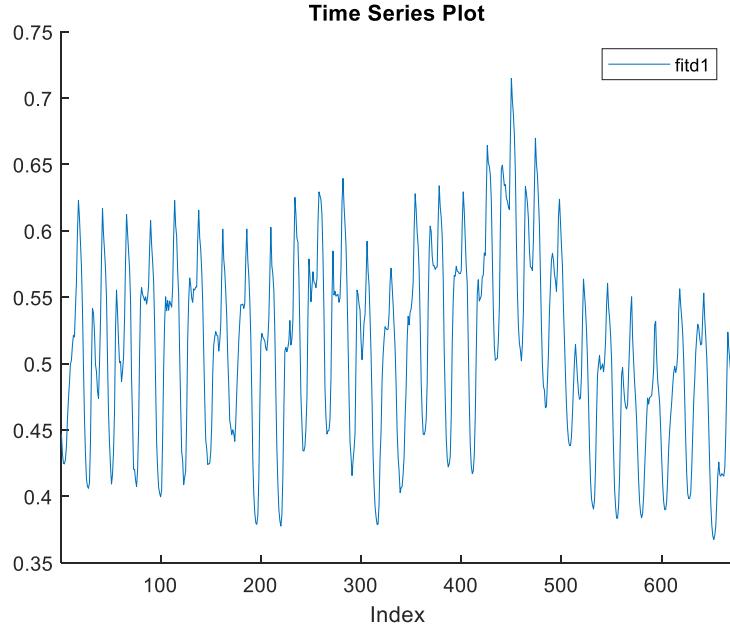
**Figure 30 - PACF Plot After Seasonal and Non-Seasonal Differencing – Toronto Dataset**

## 1.2 Statistical Analysis of the Ottawa Dataset

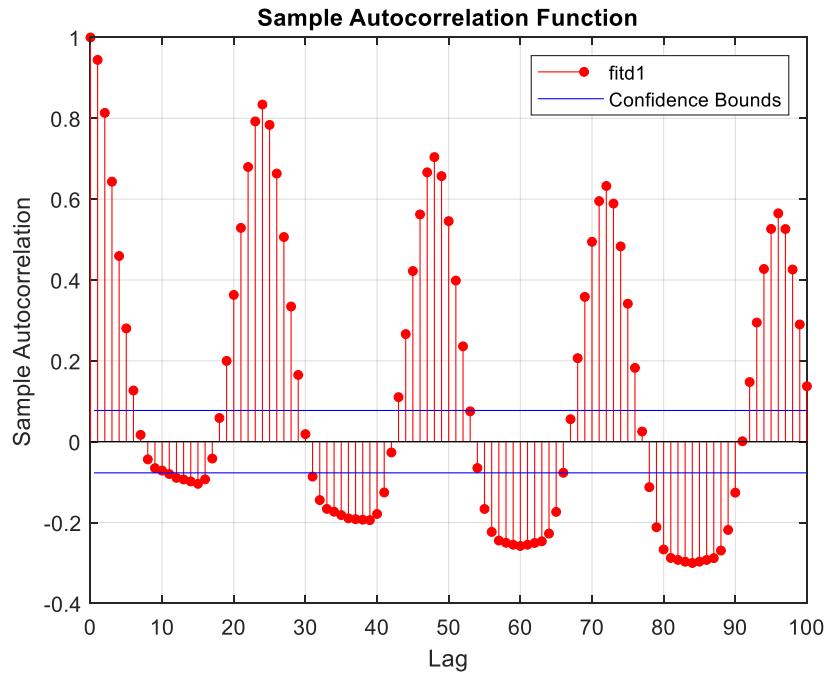
As with the Toronto dataset, a subset of 28 days of hourly values was chosen to determine the model's optimal parameters. The plot of the segment we used is shown in Figure 31.

Figure 32 depicts the initial autocorrelation function without differencing. On a 24-hour basis, we can see a high degree of seasonality, which is why we chose 24 as the seasonal difference denoted by parameter S. The ACF plot approaches zero very slowly after seasonal differencing, as shown in Figure 33. This implies that additional differentiation is required, which the non-seasonal component can provide. The new ACF plot approaches zero relatively quickly, as shown in Figure 34, indicating that nonseasonal differencing d can be set to 1. As shown in the exact figure, the first lag is extremely significant, so q is

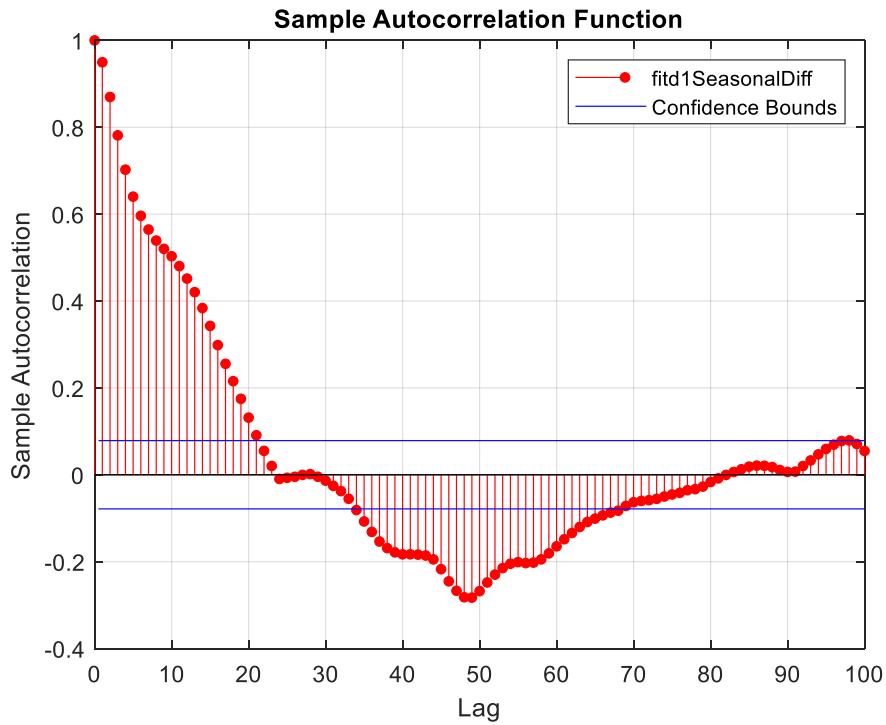
set to 1. Big Q values of 24 and 168 were chosen because these were the points with the highest auto correlation.



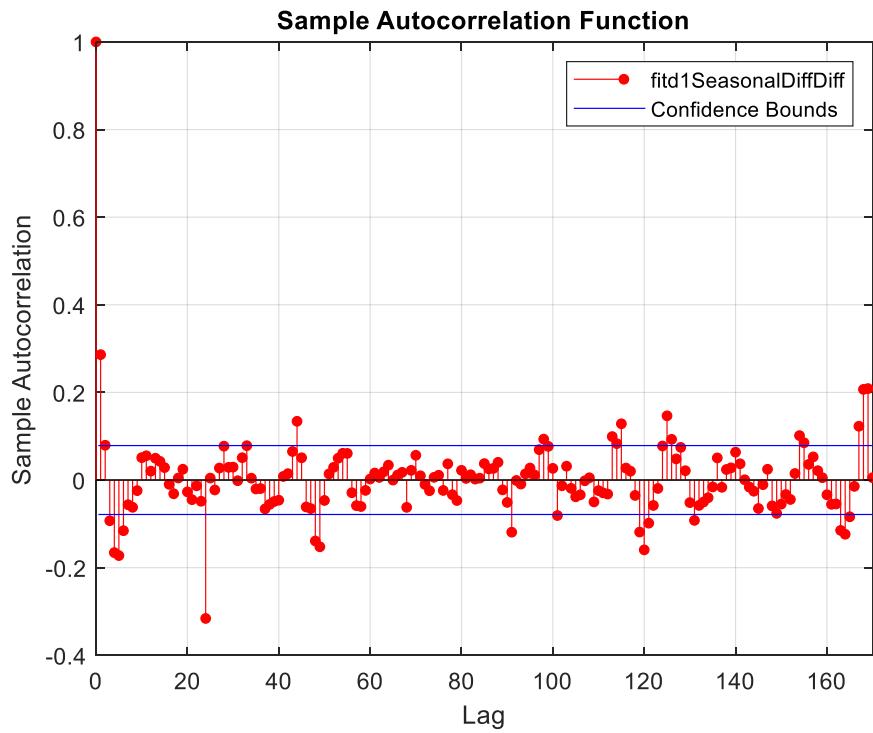
**Figure 31 - Excerpt from the Ottawa Dataset**



**Figure 32 - Plot of the Initial Auto Correlation – Ottawa Dataset**

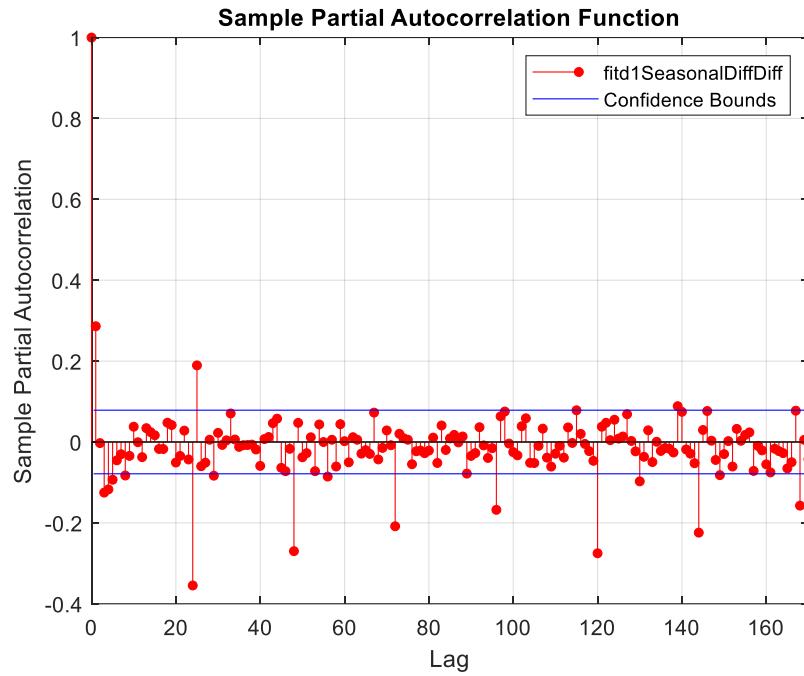


**Figure 33 - ACF Plot Following Seasonal Differencing – Ottawa Dataset**



**Figure 34 - ACF Plot After Seasonal and Non-Seasonal Differencing – Ottawa Dataset**

The values of our small p and large P lags can be found in Figure 35's PACF plot. We notice a significant lag at 1 that is greater than the lag at the subsequent ones, so we set p to 1. We considered lags of 24 and 48 for large P; to keep our model simple, we included only the points with the highest partial autocorrelation.



**Figure 35 - PACF Plot After Seasonal and Non-Seasonal Differencing – Ottawa Dataset**

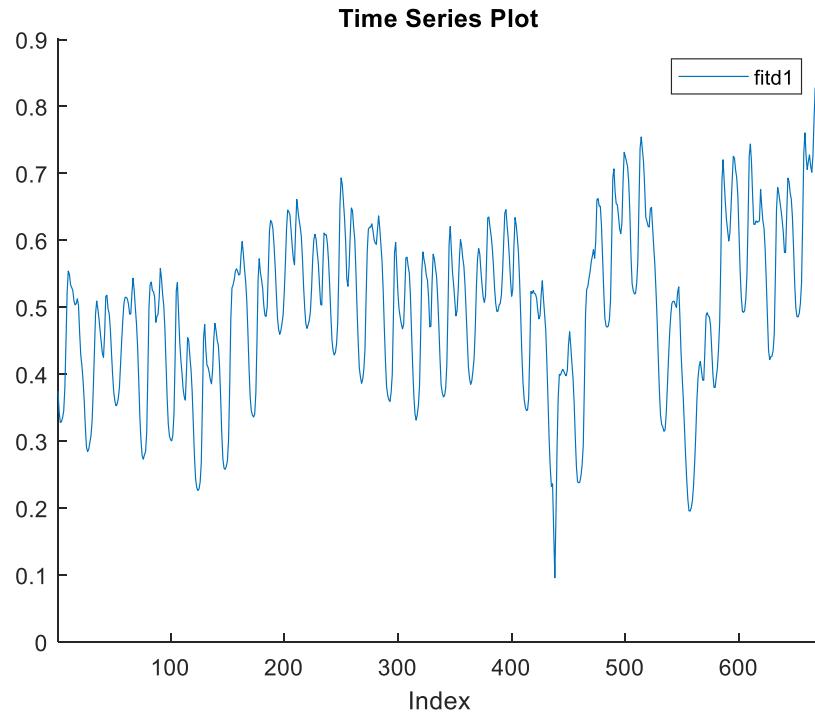
### 1.3 Statistical Analysis of the Saint John Dataset

To determine the optimal model parameters, a subset of 28 days of hourly values was chosen, as was the case with the Toronto and Ottawa datasets. The plot of the segment we used is shown in Figure 36.

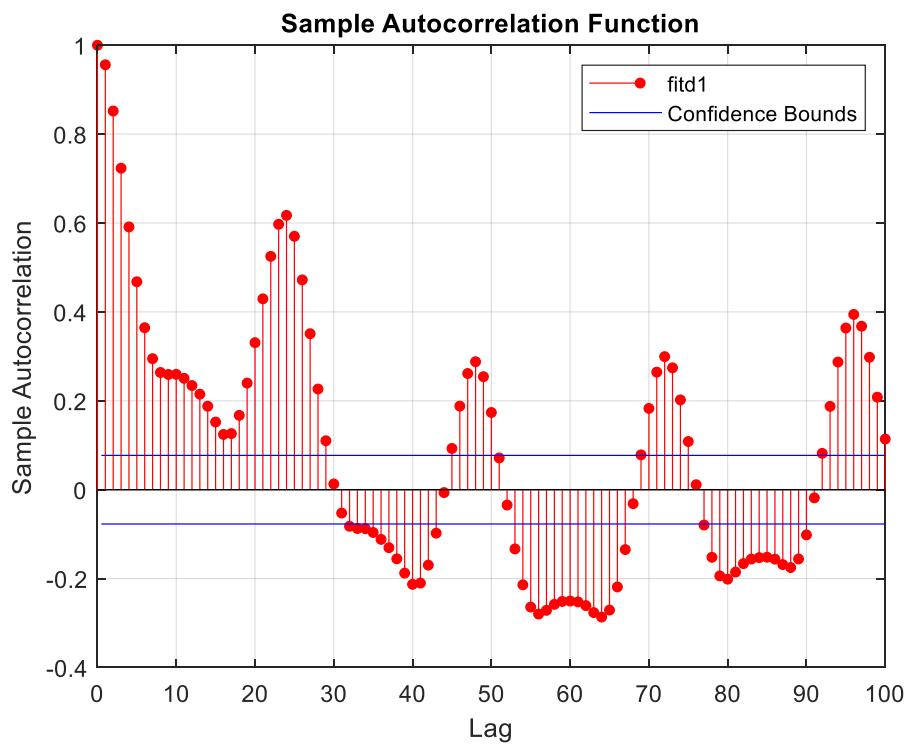
The initial autocorrelation function without differencing is depicted in Figure 37. We can see a high level of seasonality on a 24-hour basis, which is why we chose 24 as the seasonal difference denoted by parameter S. The ACF plot approaches zero very slowly

after seasonal differencing, as shown in Figure 38. This implies that additional differentiation is required, which the non-seasonal component can provide. The new ACF plot approaches zero relatively quickly, as shown in Figure 39, indicating that  $d$  can be set to one for nonseasonal differencing. Only the first lag appears to be highly significant, as shown in the exact figure; thus,  $q$  is set to one. The high  $Q$  values of 24 and 48 were chosen because they corresponded to the points with the highest auto correlation.

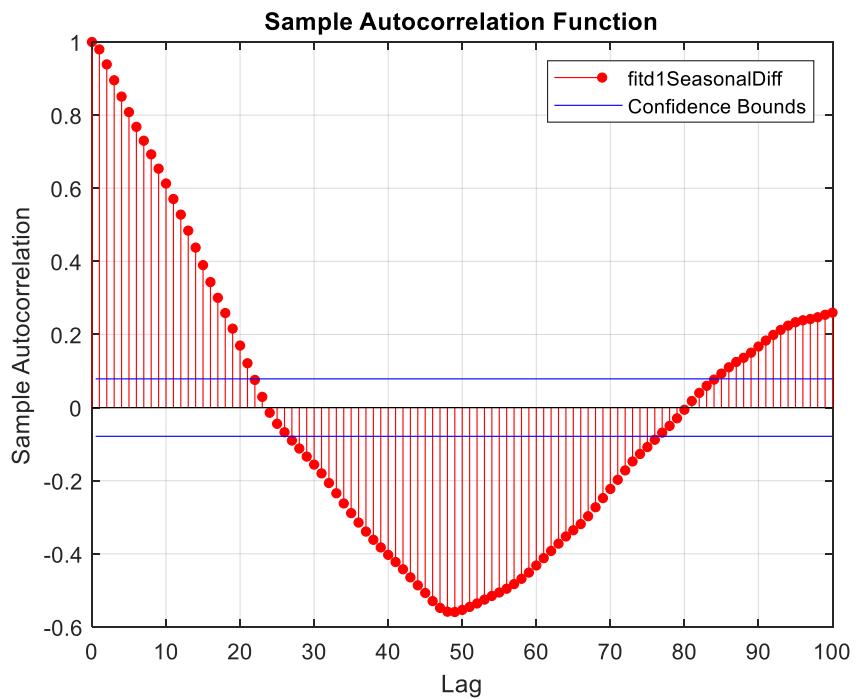
Figure 40's PACF plot shows the values of our small  $p$  and large  $P$  lags. We set  $p$  to 1 and 2 because we see a significant lag at 1 and 2 that is greater than the lag at subsequent ones. We considered lags of 24 and 168 for large  $P$ ; to simplify our model, we only included the two with the highest partial autocorrelation.



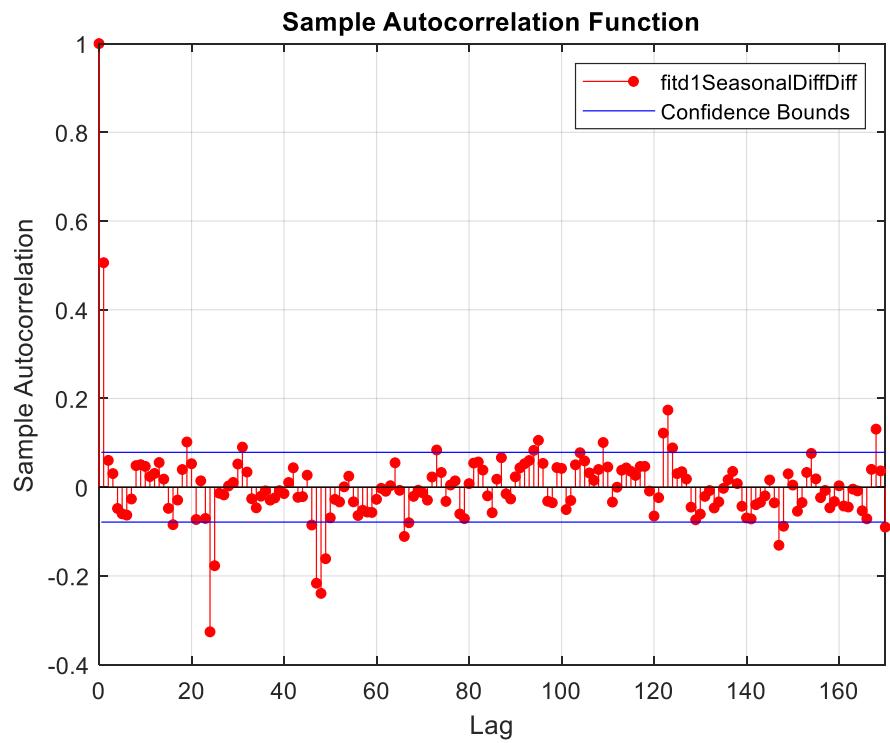
**Figure 36 - Excerpt from the Saint John Dataset**



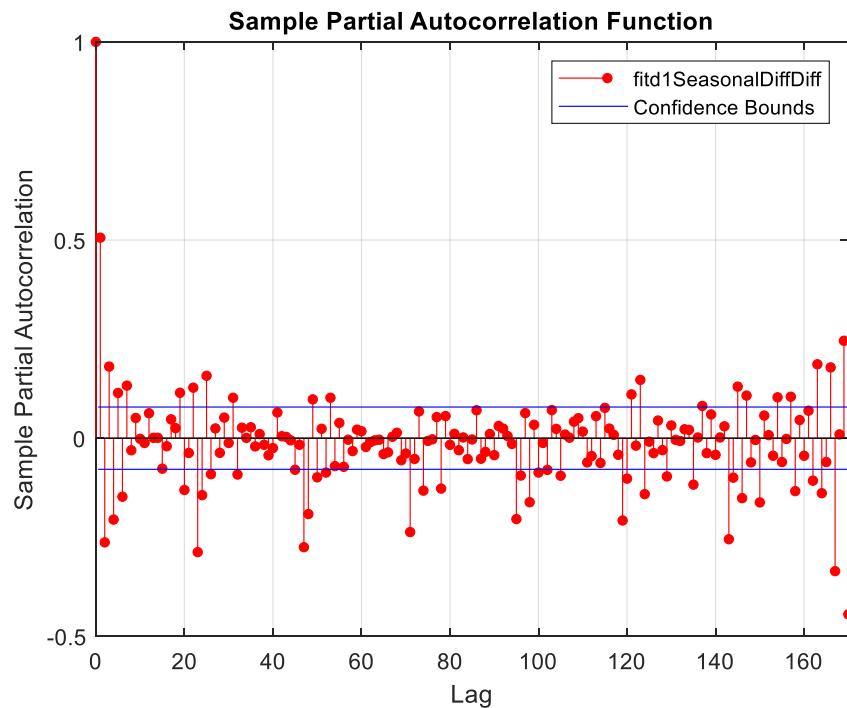
**Figure 37 - Plot of the Initial Auto Correlation – Saint John Dataset**



**Figure 38 - ACF Plot Following Seasonal Differencing – Saint John Dataset**



**Figure 39 - ACF Plot After Seasonal and Non-Seasonal Differencing – Saint John Dataset**



**Figure 40 - PACF Plot After Seasonal and Non-Seasonal Differencing – Saint John Dataset**

## Appendix B

### 1 Metrics for Overall Accuracy

#### 1.1 The Toronto Dataset's Overall Performance Metrics

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
<b>MAPE (%)</b>	2.16	2.54	2.30	3.75	4.00	6.09
<b>MBE (MW)</b>	5.41	2.86	4.04	10.81	-0.01	1.67
<b>MAE (MW)</b>	125.73	148.01	134.87	214.88	231.59	350.36
<b>RMSE (MW)</b>	189.76	219.57	201.32	293.94	321.58	488.07
<b>STD (MW)</b>	189.69	219.57	201.29	293.75	321.59	488.10

Table 17 – The Overall Performance Metrics – Toronto Dataset

#### 1.2 The Ottawa Dataset's Overall Performance Metrics

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
<b>MAPE (%)</b>	2.72	3.44	3.09	4.78	4.98	7.33
<b>MBE (MW)</b>	-0.69	0.09	-0.70	-4.29	4.69	1.05
<b>MAE (MW)</b>	27.70	35.10	31.59	49.05	50.96	75.79
<b>RMSE (MW)</b>	37.13	46.82	41.93	65.77	68.58	102.83
<b>STD (MW)</b>	37.13	46.82	41.92	65.63	68.42	102.83

Table 18 - The Overall Performance Metrics – Ottawa Dataset

#### 1.3 The Saint John Dataset's Overall Performance Metrics

Metrics	CNN	LSTM	ANN	MLR	SARIMAX	SNF
<b>MAPE (%)</b>	3.89	4.55	4.33	6.11	5.33	9.39
<b>MBE (MW)</b>	-0.08	0.32	0.19	-0.65	-0.18	0.08
<b>MAE (MW)</b>	4.71	5.43	5.08	7.18	6.30	10.96
<b>RMSE (MW)</b>	8.06	8.95	8.29	11.06	10.82	16.90
<b>STD (MW)</b>	8.06	8.95	8.29	11.04	10.82	16.91

Table 19 - The Overall Performance Metrics – Saint John Dataset

## 2 Other Forecasters' Box Plots of the Error Distribution

### 2.1 The Toronto Dataset

#### 2.1.1 The Hourly Performance

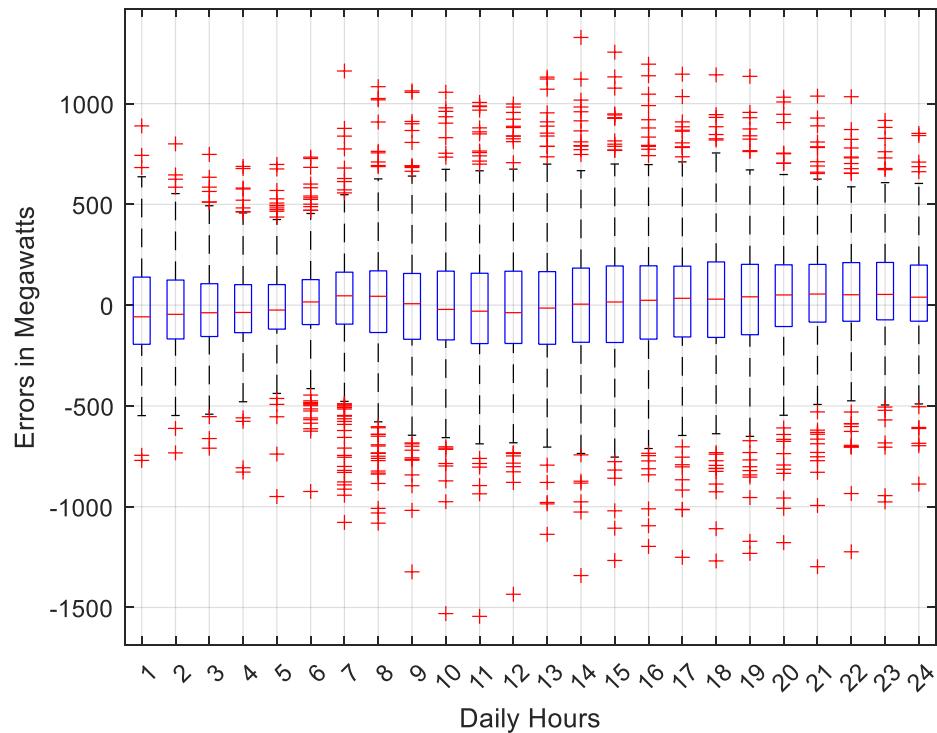
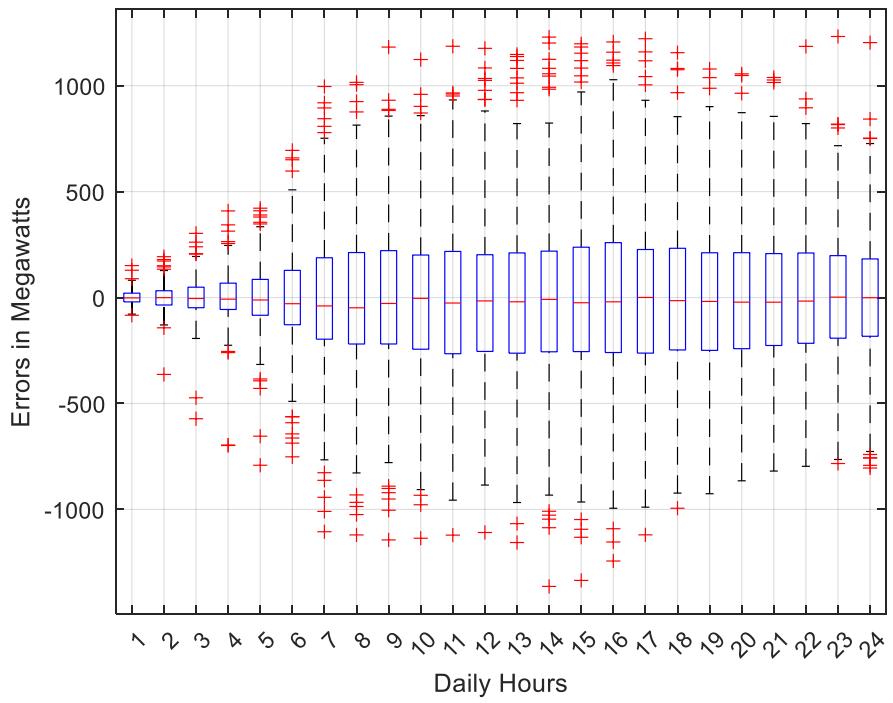
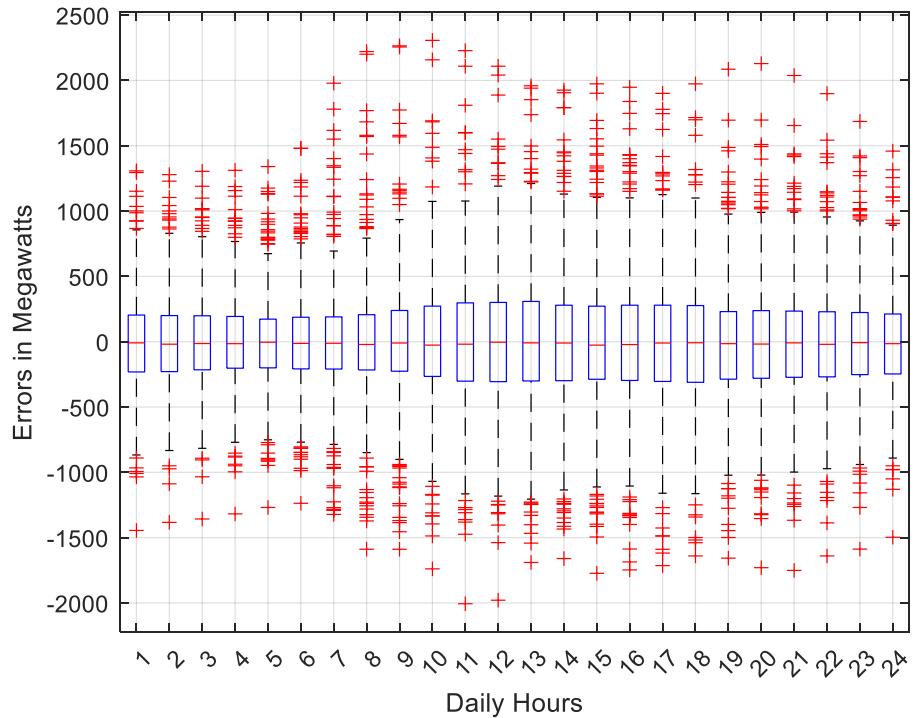


Figure 41 - Hourly Error Distribution for the MLR Forecaster – Toronto Dataset

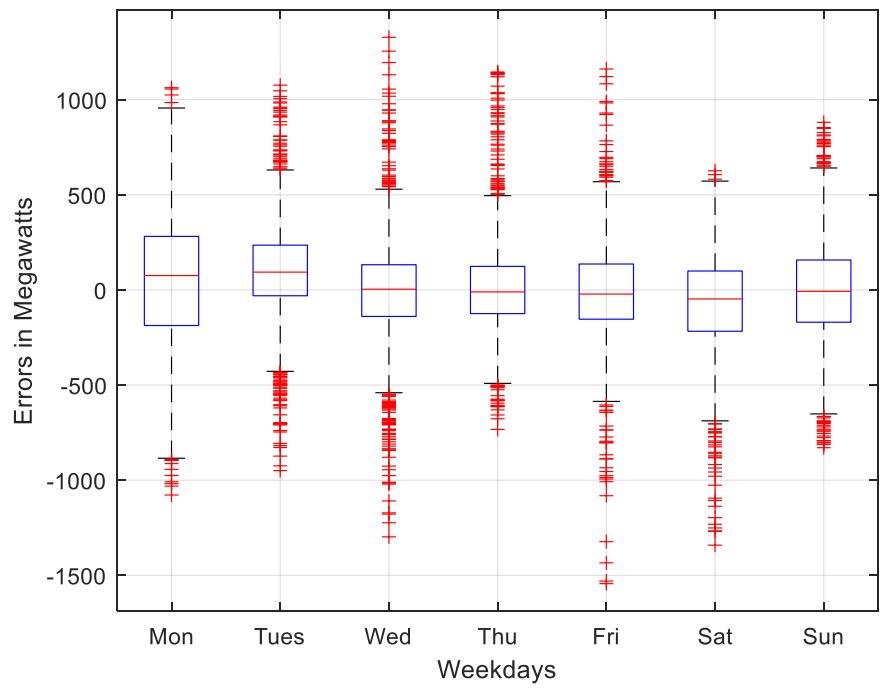


**Figure 42 - Hourly Error Distribution for the SARIMAX Forecaster – Toronto Dataset**

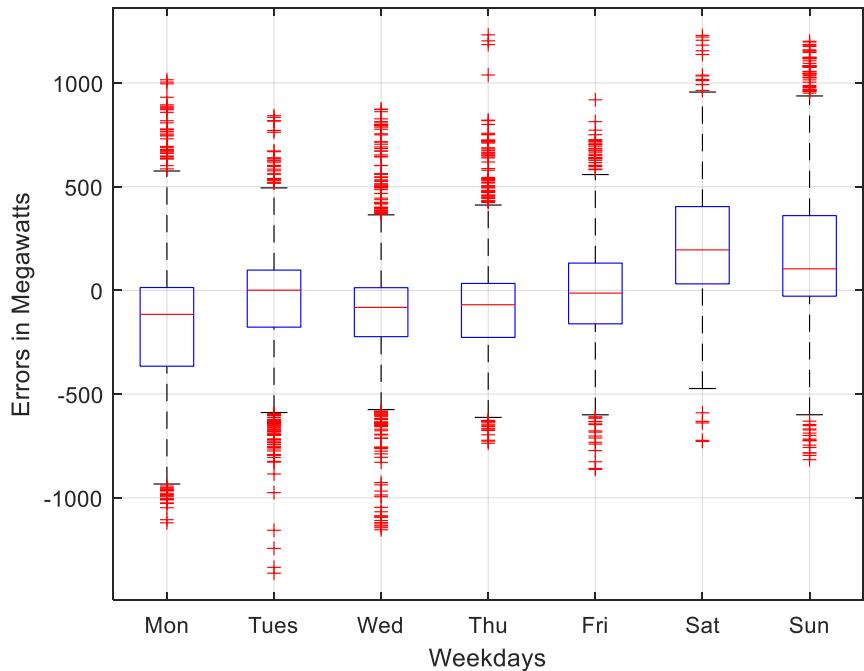


**Figure 43 - Hourly Error Distribution for the SNF Forecaster – Toronto Dataset**

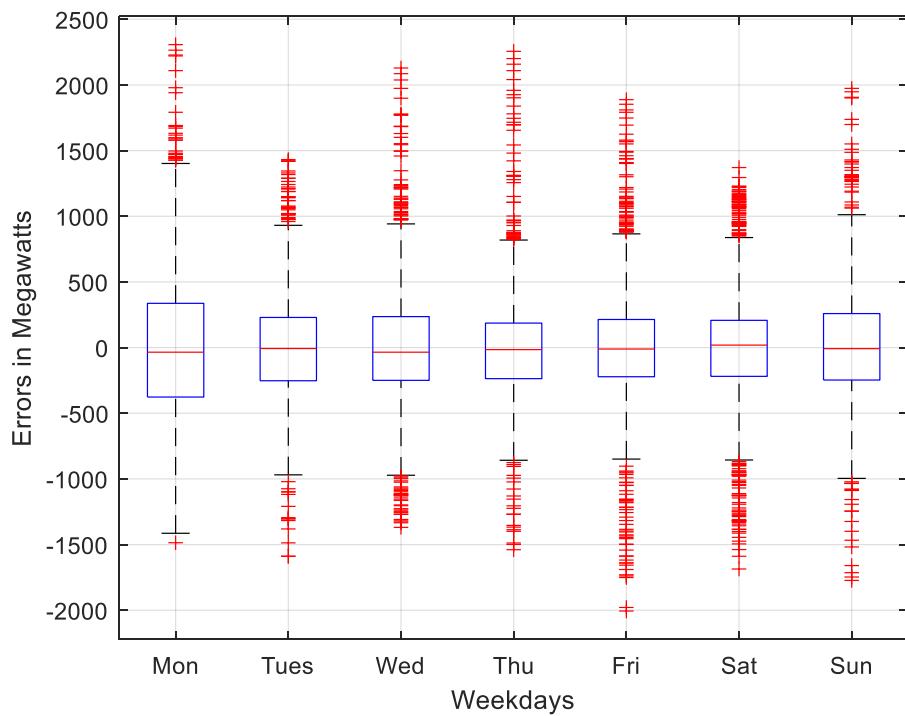
## 2.1.2 The Daily Performance



**Figure 44 - Daily Error Distribution for the MLR Forecaster – Toronto Dataset**

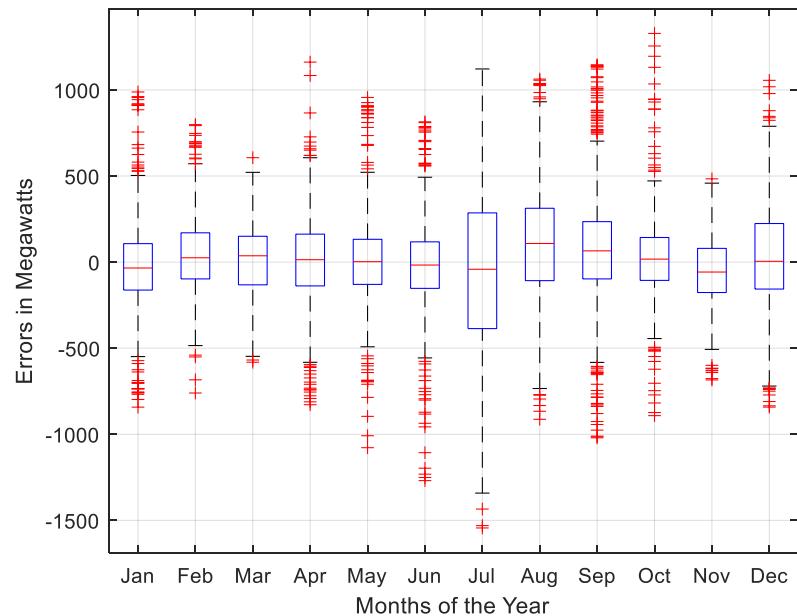


**Figure 45 - Daily Error Distribution for the SARIMAX Forecaster – Toronto Dataset**

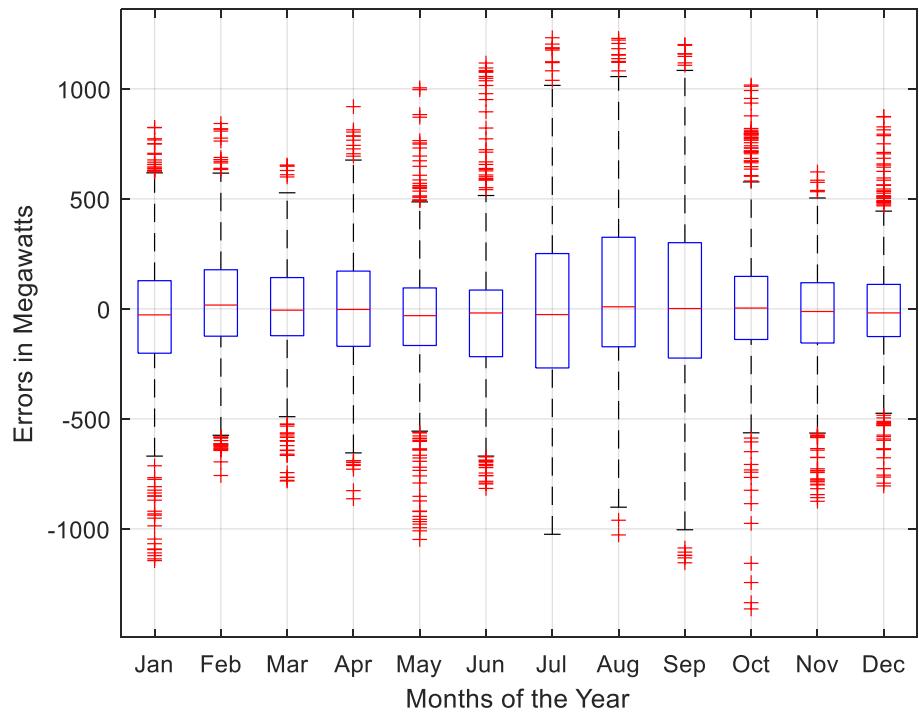


**Figure 46 - Daily Error Distribution for the SNF Forecaster – Toronto Dataset**

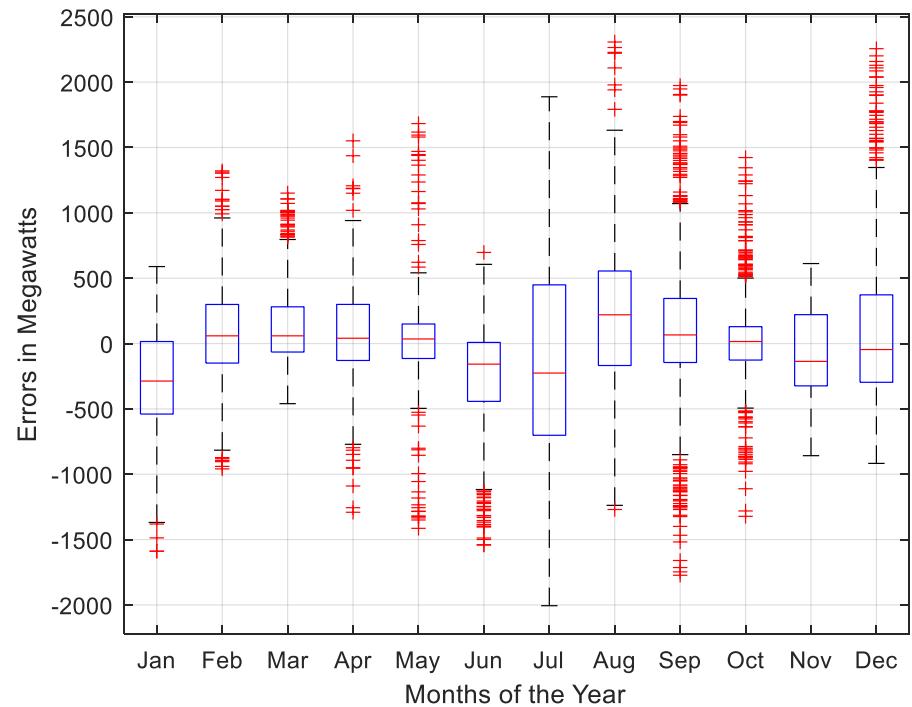
### 2.1.3 The Monthly Performance



**Figure 47 - Monthly Error Distribution for MLR Forecaster – Toronto Dataset**



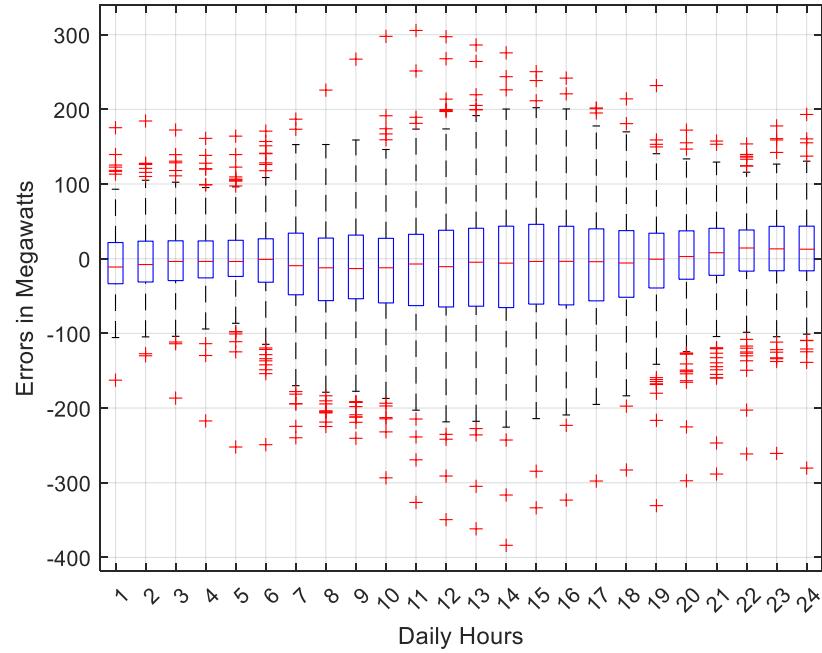
**Figure 48 - Monthly Error Distribution for SARIMAX Forecaster– Toronto Dataset**



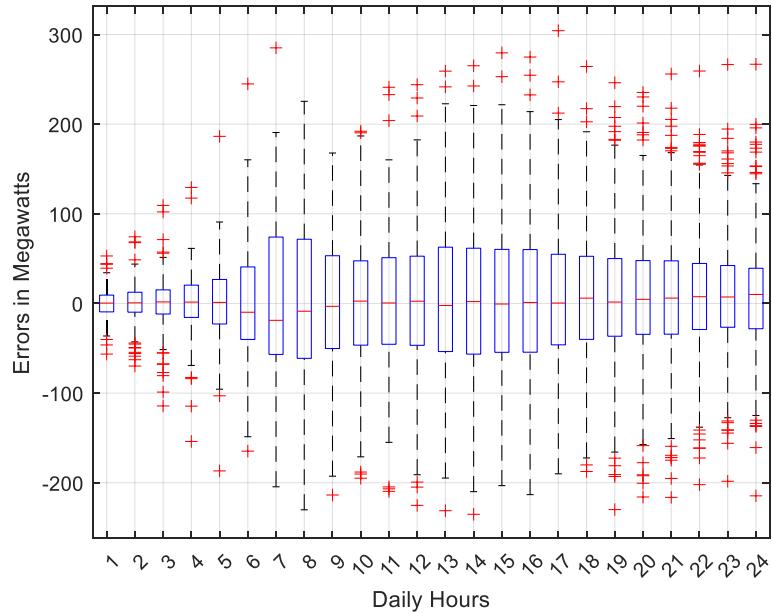
**Figure 49 - Monthly Error Distribution for SNF Forecaster– Toronto Dataset**

## 2.2 The Ottawa Dataset

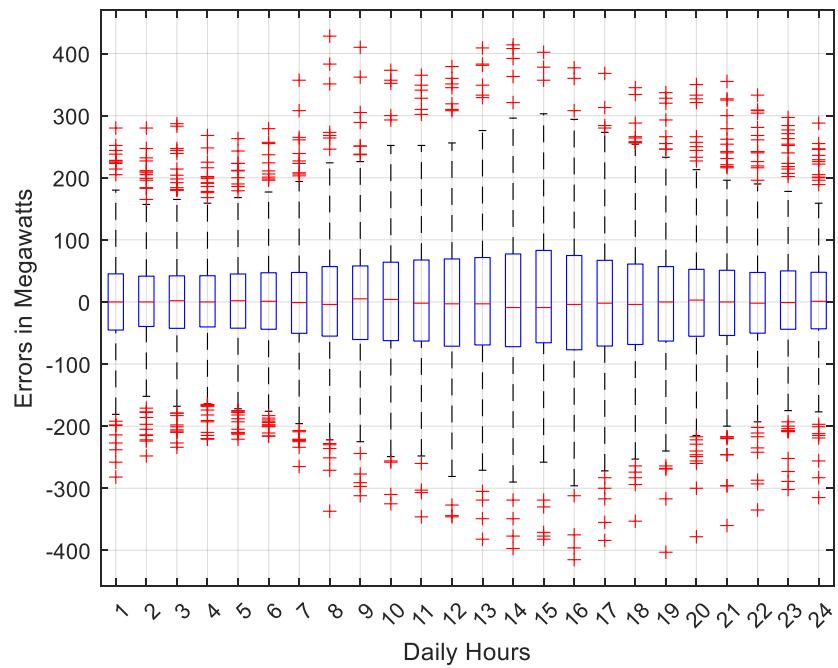
### 2.2.1 The Hourly Performance



**Figure 50 - Hourly Error Distribution for the MLR Forecaster – Ottawa Dataset**

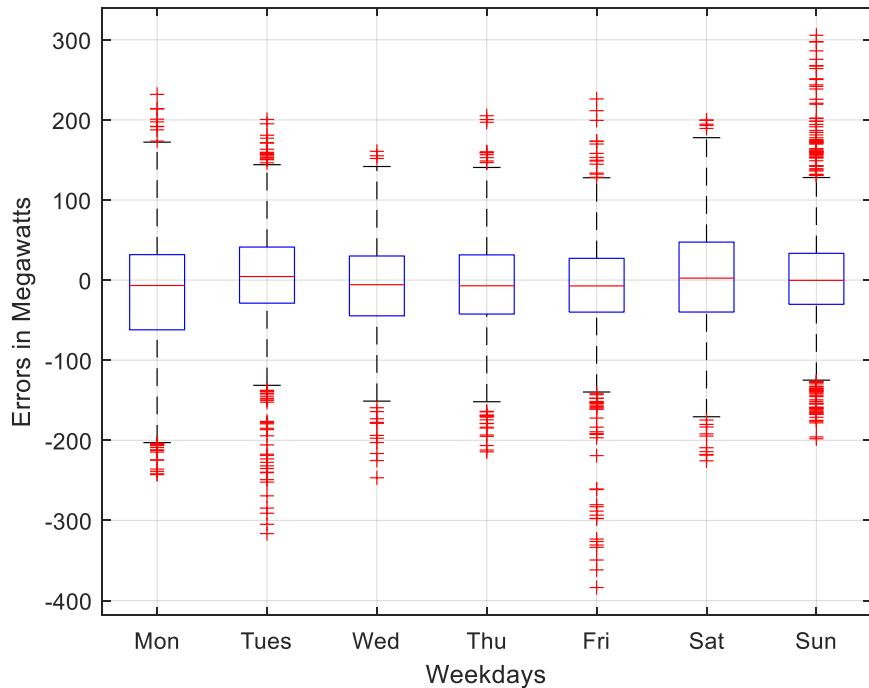


**Figure 51 - Hourly Error Distribution for the SARIMAX Forecaster – Ottawa Dataset**

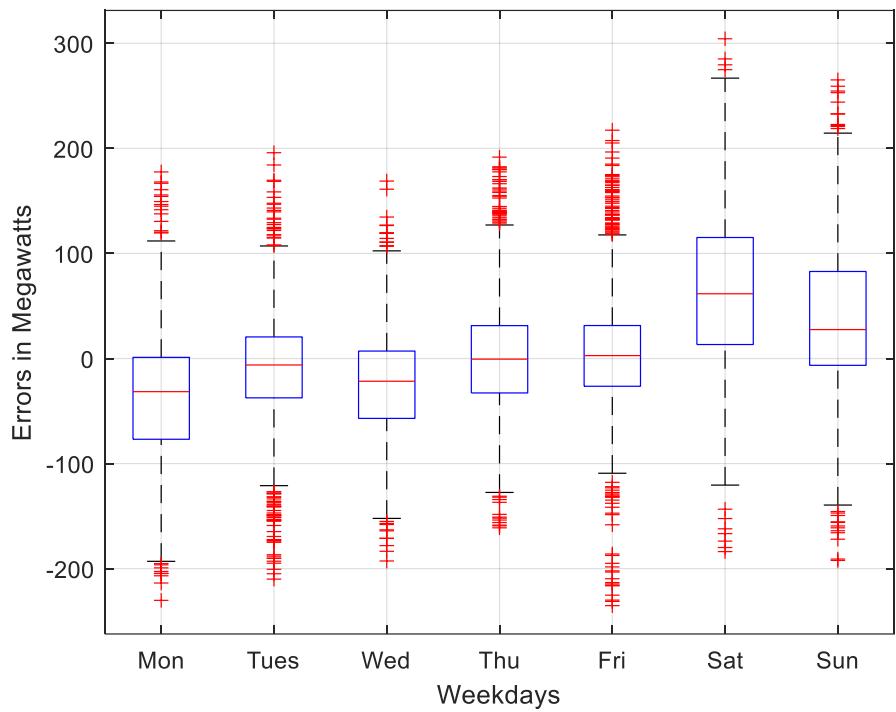


**Figure 52 - Hourly Error Distribution for the SNF Forecaster – Ottawa Dataset**

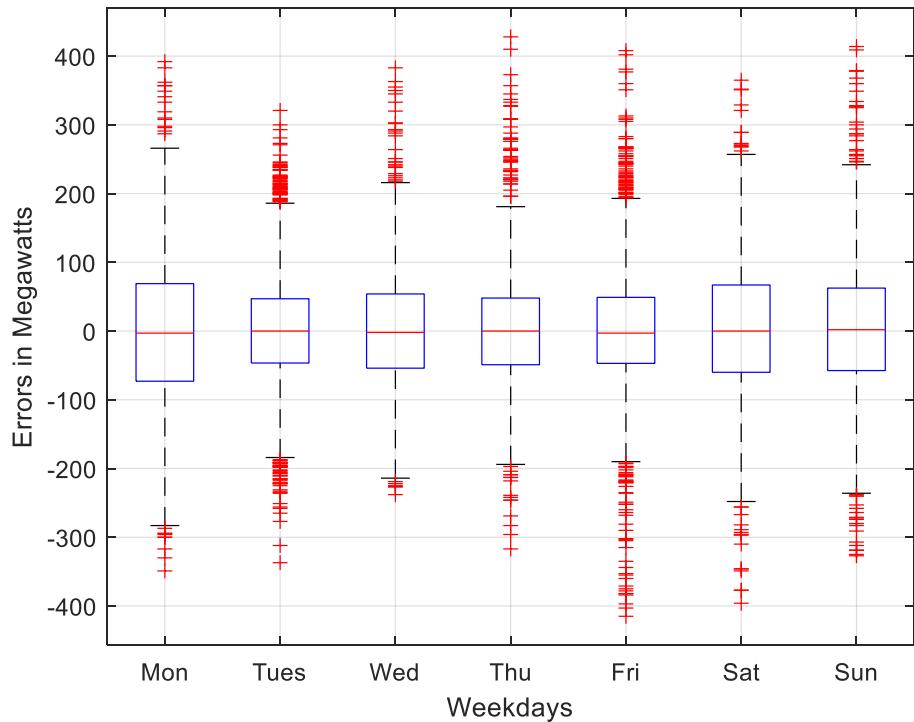
### 2.2.2 The Daily Performance



**Figure 53 - Daily Error Distribution for the MLR Forecaster – Ottawa Dataset**

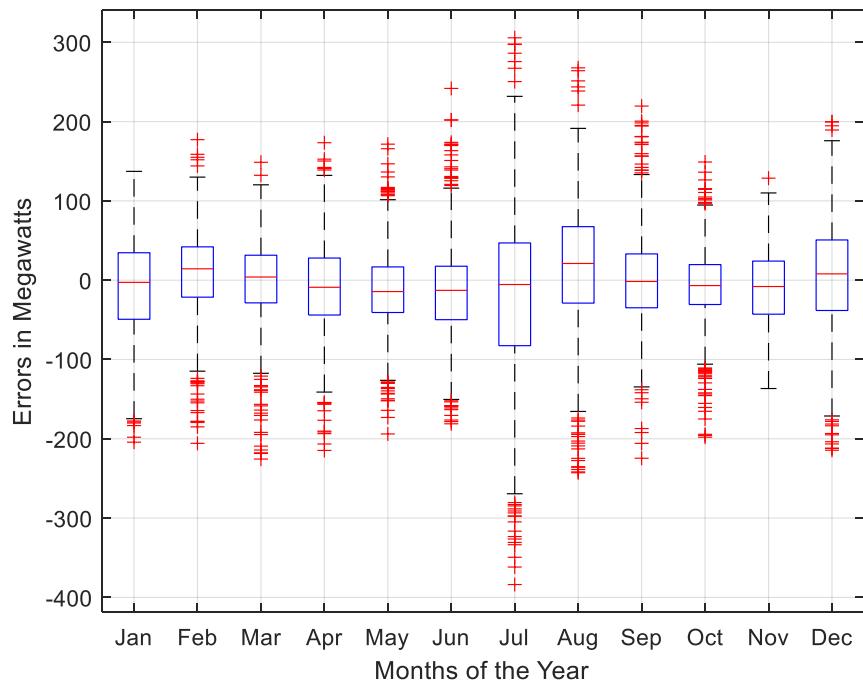


**Figure 54 - Daily Error Distribution for the SARIMAX Forecaster – Ottawa Dataset**

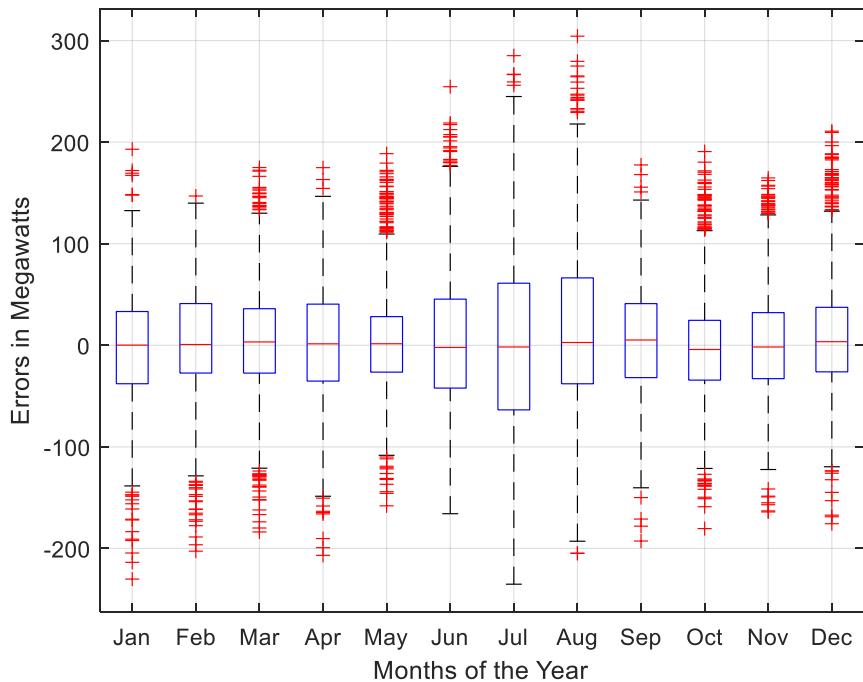


**Figure 55 - Daily Error Distribution for the SNF Forecaster – Ottawa Dataset**

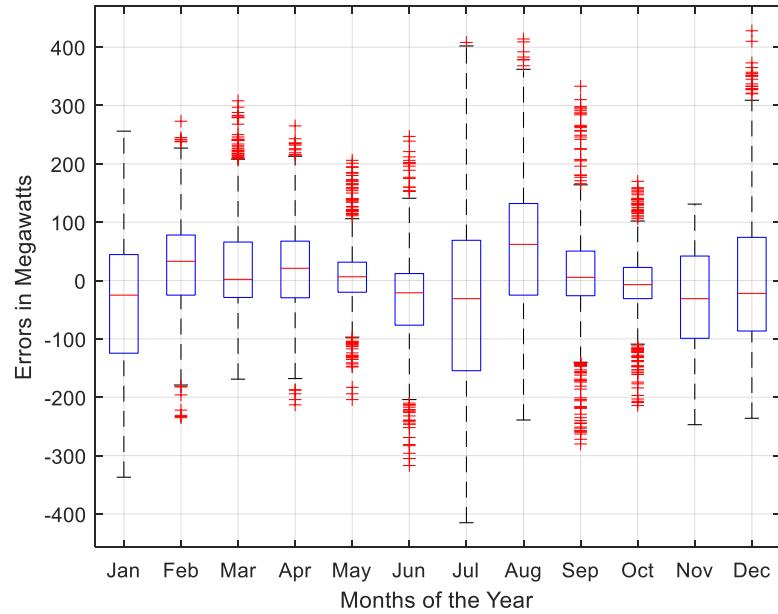
### 2.2.3 The Monthly Performance



**Figure 56 - Monthly Error Distribution for MLR Forecaster – Ottawa Dataset**



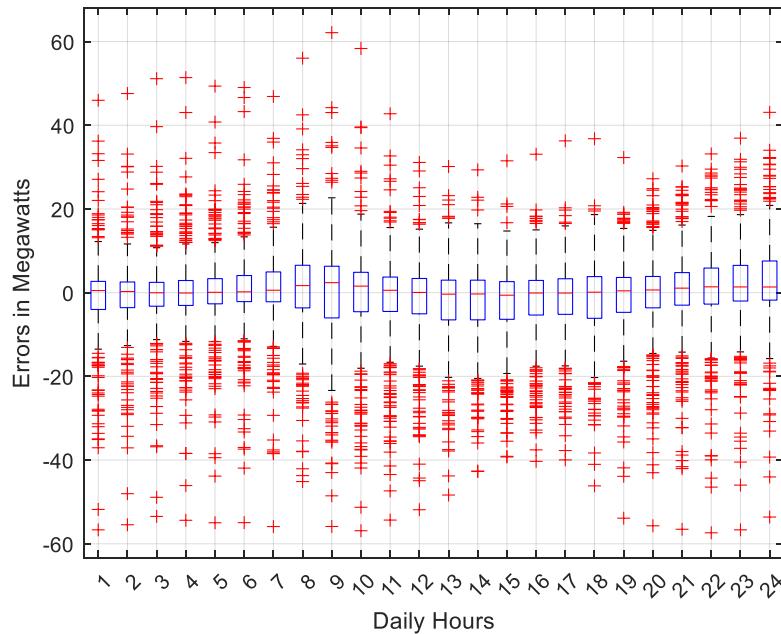
**Figure 57 - Monthly Error Distribution for SARIMAX Forecaster – Ottawa Dataset**



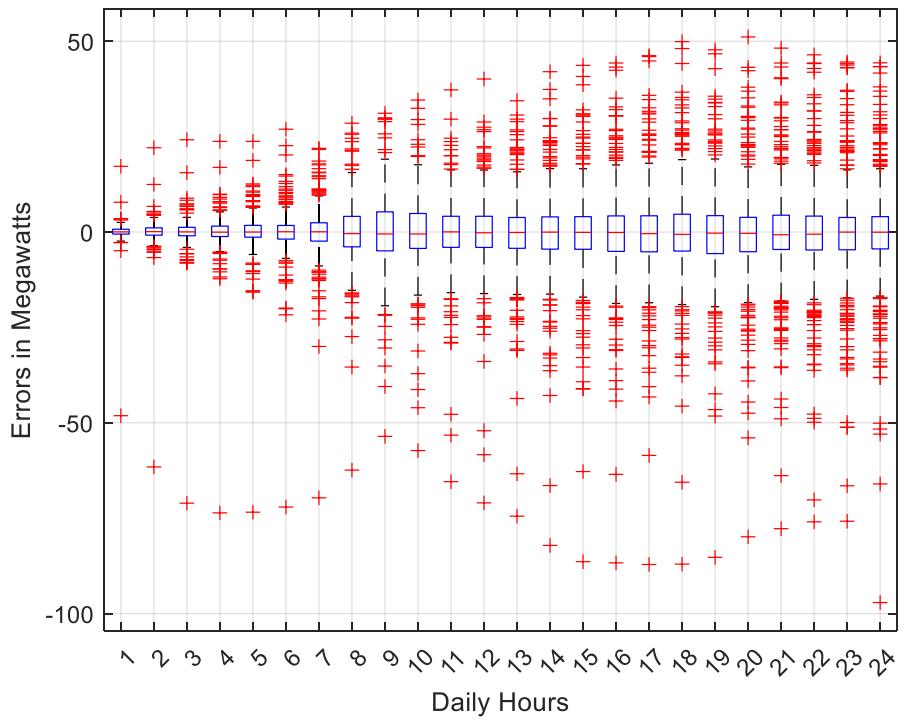
**Figure 58 - Monthly Error Distribution for SNF Forecaster – Ottawa Dataset**

## 2.3 The Saint John Dataset

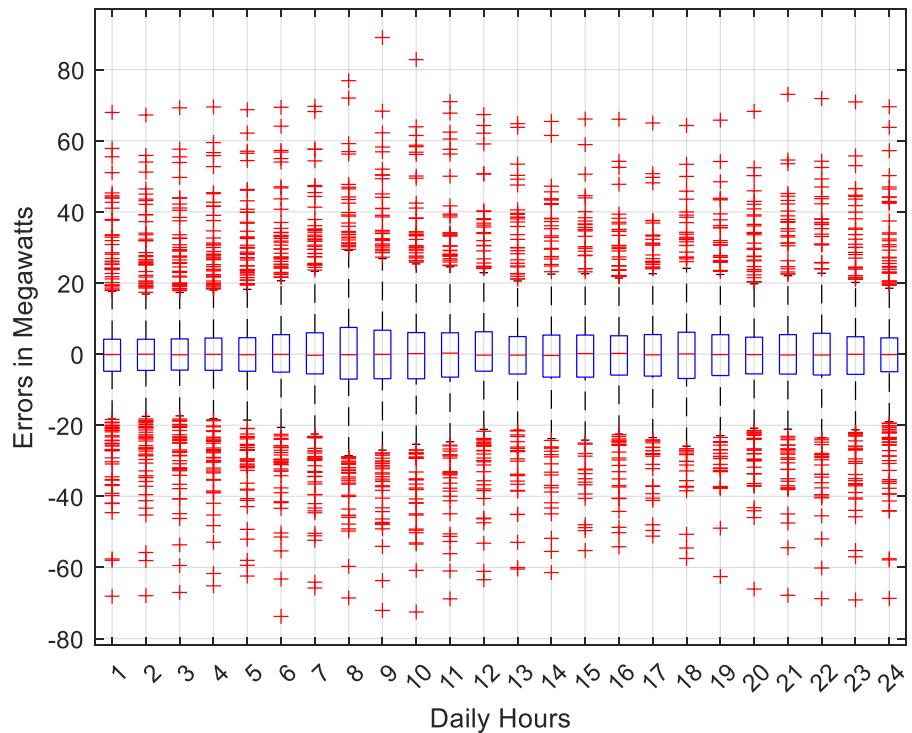
### 2.3.1 The Hourly Performance



**Figure 59 - Hourly Error Distribution for the MLR Forecaster – Saint John Dataset**



**Figure 60 - Hourly Error Distribution for the SARIMAX Forecaster – Saint John Dataset**



**Figure 61 - Hourly Error Distribution for the SNF Forecaster – Saint John Dataset**

### 2.3.2 The Daily Performance

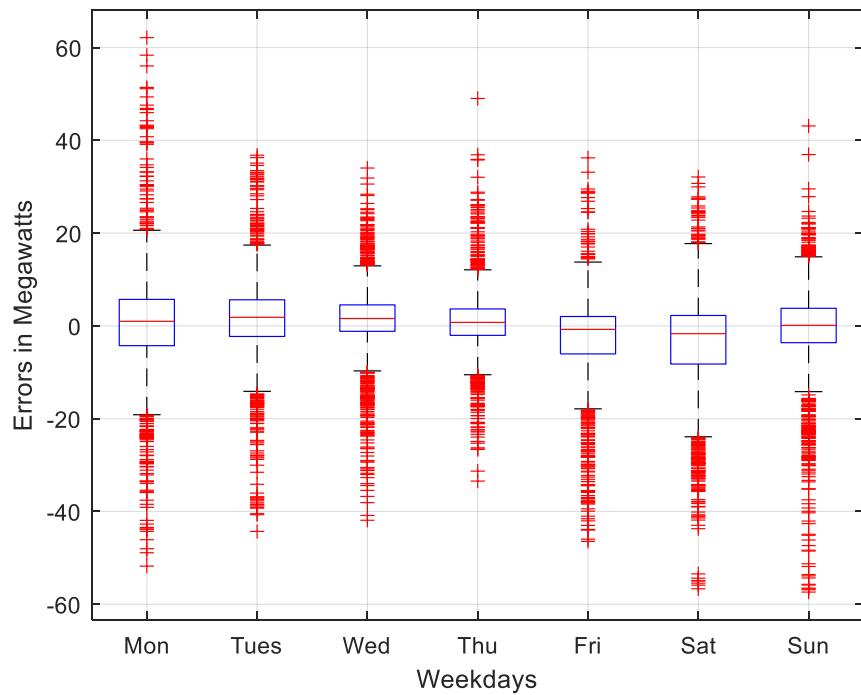


Figure 62 - Daily Error Distribution for the MLR Forecaster – Saint John Dataset

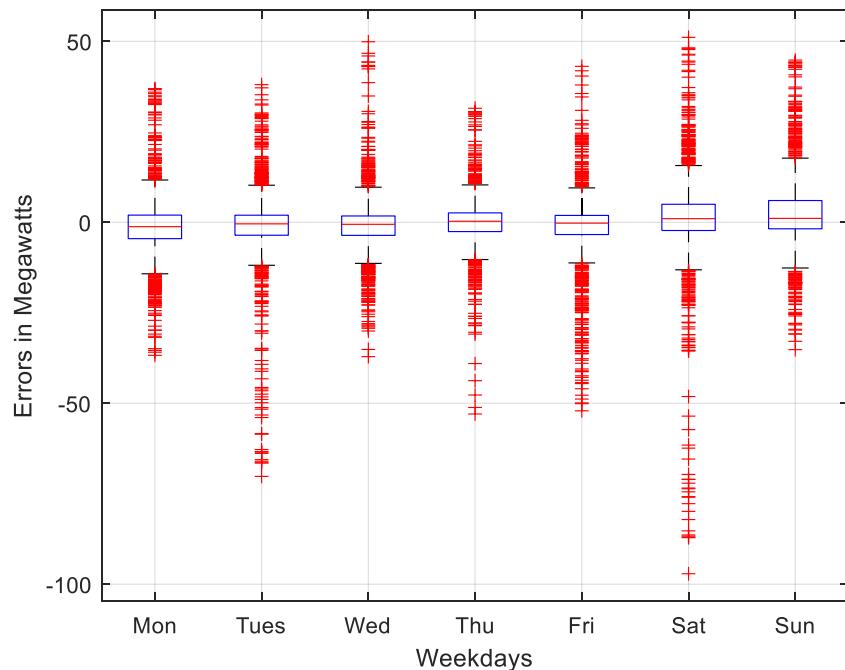
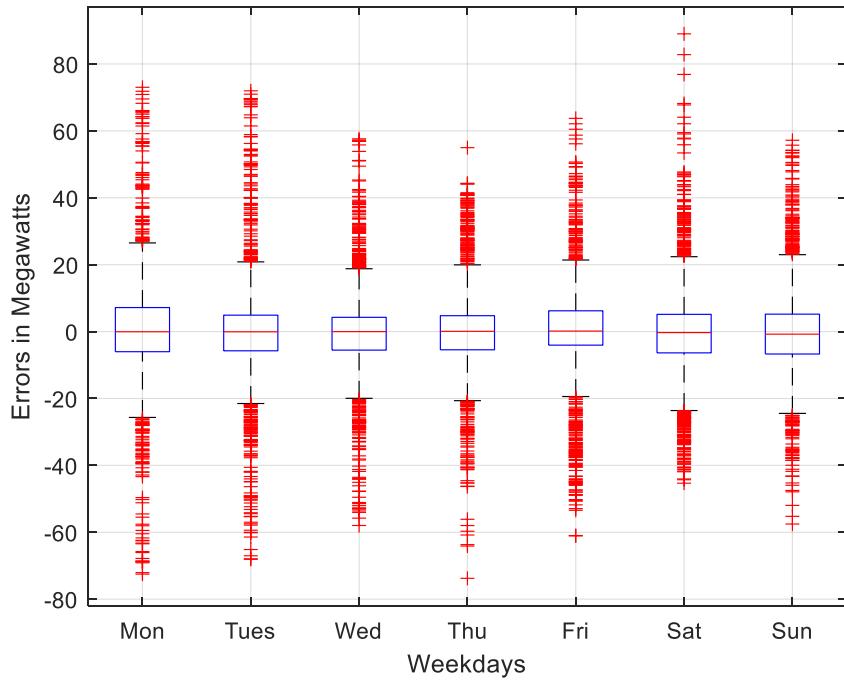
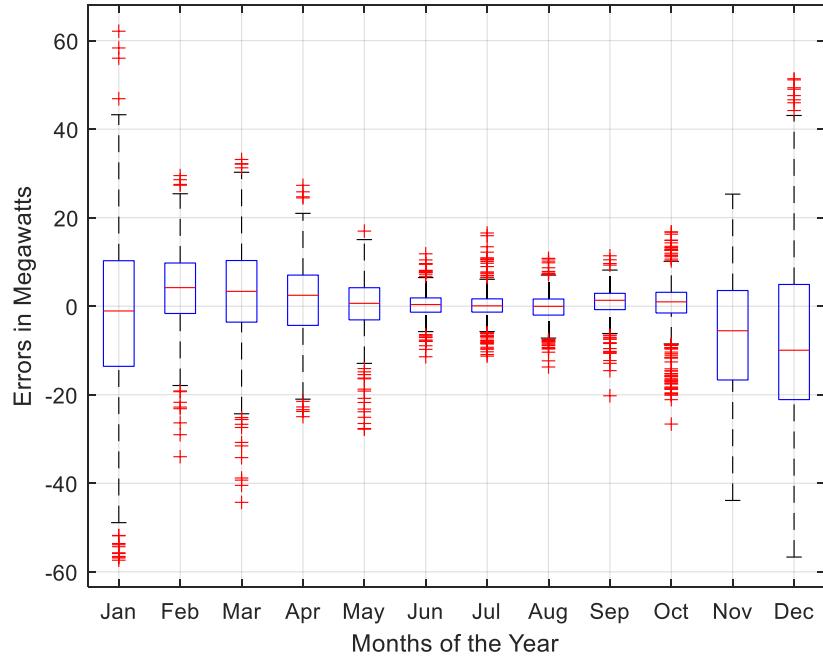


Figure 63 - Daily Error Distribution for the SARIMAX Forecaster – Saint John Dataset

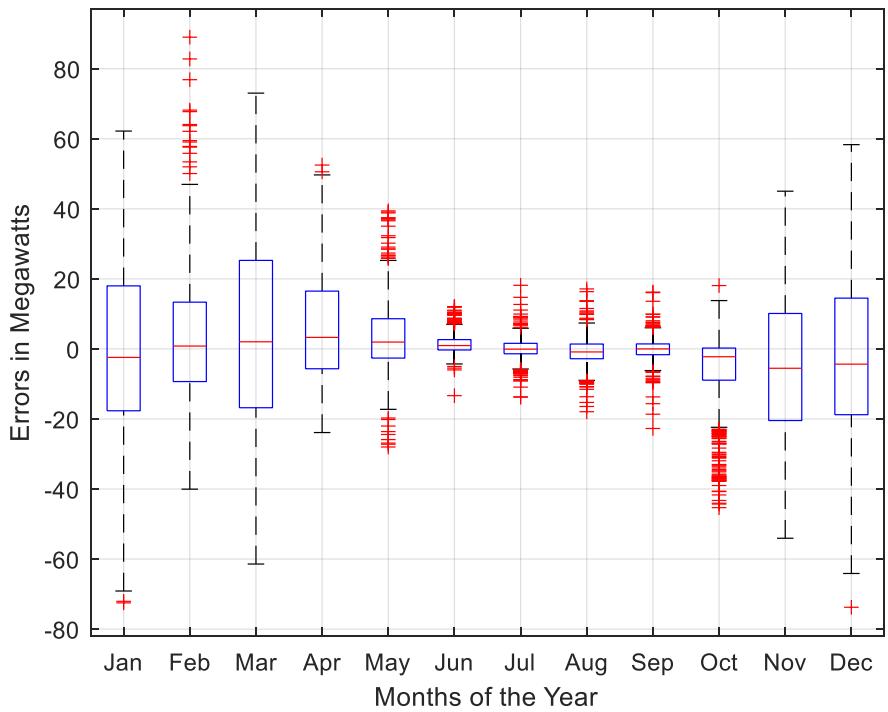


**Figure 64 - Daily Error Distribution for the SNF Forecaster – Saint John Dataset**

### 2.3.3 The Monthly Performance



**Figure 65 - Monthly Error Distribution for MLR Forecaster – Saint John Dataset**



**Figure 66 - Monthly Error Distribution for SNF Forecaster – Saint John Dataset**

## **Curriculum Vitae**

**Candidate's full name:** Tolulope Oluwaseun Olugbenga

**Universities attended:** University of Debrecen, Debrecen, Hungary

**Degree:** BSc. in Computer Science Engineering, 2018

**Publications:** None

**Conference Presentations:** None