Deep Learning Techniques for Forecasting Electrical Loads

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: (name, degree, department/field), Chair

(name, degree, department/field)

(continue as required)

This thesis is accepted by the

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

December 2021

© Tolulope Oluwaseun Olugbenga, 2021

ABSTRACT - 150 words

For over a century, load forecasting has been used to plan and operate electric grids. Load aggregators, power marketers, independent system operators, regulatory commissions, industrial/commercial firms all benefit from load forecasting for revenue projection, energy trading, rate design, and other purposes. Load forecasting has been extensively studied in the literature [1]–[9]. Forecasting load has been done using statistical and machine learning (ML) techniques, and the line between the two is becoming increasingly blurred [1]. However, machine learning algorithms are more intelligent and can adapt to complex relationships between load and other influencing factors (weather) [10]. Deep learning approaches like recurrent neural networks (RNN) [11], long-short-term memory networks (LSTM) [12], and 1-D convolution neural networks (CNN) [3], [7] have recently caught the attention of researchers in this field. This study compares deep learning forecasters to some conventional forecasters used by utilities to see if deep learning better suits their needs.

DEDICATION

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

ACKNOWLEDGEMENTS

Without a doubt, this is one of the most challenging journeys I have ever undertaken. It has demonstrated how much I can grow and achieve when I believe in myself and put in the effort. 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. Then, after hearing what other students had to say about how difficult a thesis master's degree is, I began to doubt my abilities and became concerned that I would not be able to finish it.

I want to express my gratitude to my supervisors, Dr. Dawn MacIsaac and Dr. Julian Cardenas; without them, I would not have completed this program. I appreciate your patience and encouraging words, which reminded me that anything is possible and that all I need to do is keep going. I would also like to thank my family for always being there for me and constantly motivating me to finish this program.

If I am completely honest, the person who first entered the lab in December 2018 would not be able to complete this degree. In order to finish, I needed to improve both personally and intellectually. 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 short, this has been an educational experience and a game of physical and cognitive development. If I had to do it all over again, I would 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 - 150 words ii](#_Toc82948985)

[DEDICATION iii](#_Toc82948986)

[ACKNOWLEDGEMENTS iv](#_Toc82948987)

[Table of Contents v](#_Toc82948988)

[List of Tables viii](#_Toc82948989)

[List of Figures ix](#_Toc82948990)

[List of Abbreviations x](#_Toc82948991)

[1 Introduction 1](#_Toc82948992)

[1.1 Objectives and Contributions 3](#_Toc82948993)

[1.2 Outline of the Thesis 4](#_Toc82948994)

[2 Overview of Load Forecasting 5](#_Toc82948995)

[2.1 Factors that affect the load demand 5](#_Toc82948996)

[2.1.1 Historical load 5](#_Toc82948997)

[2.1.2 Economic Factors 6](#_Toc82948998)

[2.1.3 Chronological Factors 6](#_Toc82948999)

[2.1.4 Meteorological Factors 7](#_Toc82949000)

[2.1.5 Random Factors 9](#_Toc82949001)

[2.2 Load Forecasting Horizons 9](#_Toc82949002)

[2.2.1 Very Short-Term Load Forecasting (VSTLF) 10](#_Toc82949003)

[2.2.2 Short Term Load Forecasting (STLF) 11](#_Toc82949004)

[2.2.3 Medium-Term Load Forecasting (MTLF) 11](#_Toc82949005)

[2.2.4 Long Term Load Forecasting (LTLF) 12](#_Toc82949006)

[2.3 Overview of Load Forecasting Techniques 13](#_Toc82949007)

[2.3.1 Statistical and Machine Learning Techniques 13](#_Toc82949008)

[2.3.2 Deep Learning Techniques 14](#_Toc82949009)

[2.3.3 The Myth of Finding the One Size Fits All Technique 15](#_Toc82949010)

[2.4 Description of the Benchmark Techniques 16](#_Toc82949011)

[2.4.1 The Seasonal Naïve Forecaster (SNF) 16](#_Toc82949012)

[2.4.2 The Multiple Linear Regression Forecaster (MLR) 17](#_Toc82949013)

[2.4.3 The Auto-Regressive Integrated Moving Average Forecaster (ARIMA) 19](#_Toc82949014)

[2.4.4 Artificial Neural Networks (ANNs) 21](#_Toc82949015)

[3 Investigation 29](#_Toc82949016)

[3.1 The Deep Learning Techniques 30](#_Toc82949017)

[3.1.1 The Long Short Term Memory Forecaster (LSTM) 34](#_Toc82949018)

[3.1.2 The Convolutional Neural Network Forecaster (CNN) 37](#_Toc82949019)

[3.2 Specifications for the Algorithms 40](#_Toc82949020)

[3.2.1 The MLR Forecaster 41](#_Toc82949021)

[3.2.2 The ANNSTLF-G3 Forecaster 41](#_Toc82949022)

[3.2.3 The LSTM Forecaster 42](#_Toc82949023)

[3.2.4 The CNN Forecaster 42](#_Toc82949024)

[3.3 How our Results Were Analyzed 43](#_Toc82949025)

[4 Results and Discussion 44](#_Toc82949026)

[4.1 Performance Metrics 44](#_Toc82949027)

[5 Conclusion 47](#_Toc82949028)

[5.1 Contributions 47](#_Toc82949029)

[5.2 Future Work 47](#_Toc82949030)

[Bibliography 48](#_Toc82949031)

[Appendix Title 68](#_Toc82949032)

[Glossary 69](#_Toc82949033)

Curriculum Vitae

List of Tables

[Table 1 44](#_Toc82949034)

List of Figures

[Figure 1 - An artificial neuron’s workflow 22](#_Toc82949035)

[Figure 2 - Examples of the most frequently used ANN activation functions [111] 23](#_Toc82949036)

[Figure 3 - The Structure of a simple feed-forward ANN [112] 24](#_Toc82949037)

[Figure 4 - The Block Diagram of the third generation ANNSTLF [37] 27](#_Toc82949038)

[Figure 5 – A simple network versus a deep learning network [130] 31](#_Toc82949039)

[Figure 6 - The Long Short-Term Memory Structure [147] 35](#_Toc82949040)

[Figure 7 - An Architecture of a one-dimensional CNN for time series data [162] 38](#_Toc82949041)

[Figure 8 – The Rectified Linear Unit Activation Function [163] 39](#_Toc82949042)

[Figure 9 – Examples of Max and Average Pooling [164] 39](#_Toc82949043)

[Figure 10 – The structure of the BLF and CLF network 42](#_Toc82949044)

List of Abbreviations

|  |  |
| --- | --- |
| SNF – Seasonal Naïve Forecaster | SVM – Support Vector Machine |
| MLR – Multiple Linear Regression | MAE – Mean Absolute Error |
| ARIMA – Auto-Regressive Integrated Moving Average | MAPE – Mean Absolute Percentage Error |
| ML - Machine Learning | MBE – Mean Biased Error |
| RNN – Recurrent Neural Networks | MSE – Mean Squared Error |
| LSTM – Long Short-Term Memory networks | RMSE – Root Mean Square Error |
| CNN – Convolutional Neural Networks | SD – Standard Deviation |
| ANN – Artificial Neural Networks | UNB – University of New Brunswick |
| BLF – Base Load Forecaster | MLP – Multilayer Perceptron |
| CLF – Change in Load Forecaster | MW – Mega-Watt |
| RLS – Recursive Least Squares |  |
| STLF – Short Term Load Forecasting |  |
| MTLF – Medium-Term Load Forecasting |  |
| LTLF – Long Term Load Forecasting |  |
| VMD - Variational Mode Decomposition |  |
| GRU – Gated Neural Networks |  |
| MLP – Multilayer Perceptron |  |
| ReLU – Rectified Linear Unit |  |

# 1 Introduction

Load forecasting is a critical component of electric utility design, planning, and operation; it has played a vital role in the power industry for over a century [1], [3], [5], [7], [11]–[14]. Load forecasting is a critical building component for power system operators to ensure the network is continuously operating and managed safely and efficiently. An important objective for load forecasting is to ensure that consumers have an adequate energy supply to maintain the balance of supply and demand. However, load forecasting can also be helpful 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], [2]. These organizations use load forecasting in power systems planning/operations, revenue projection, rate design, energy trading, and other activities [3]–[5].

Over the last decade, there has been a surge in the adoption of renewable energy and distributed generation sources and the advancement and implementation of smart grids and buildings to meet growing energy demands effectively. To integrate the preceding without causing system disruptions, it is necessary to have reliable load forecasting across multiple time horizons [15]. Electric load forecasting is well studied [1], [7]–[9], and most current research focuses on developing more accurate forecasts. The demand patterns used to drive modern technologies are complex due to the deregulation of energy markets and several random factors, often governed by human behaviour, which must be considered to predict future electricity demand. Therefore, developing a forecasting model appropriate for a particular power network is not a simple task [4]–[6]. From a financial standpoint, over-forecasting, or forecasting more power than required, results in the start-up of an excessive number of generating units, resulting in over-production and unnecessary expense. Conversely, underestimating the required demand is a result of a higher load than expected and results in electricity deficit. When this occurs, the system operator is forced to purchase potentially pricey peaking power to cover the difference at significantly higher than the market price. Both situations result in suboptimal generation scheduling and present technical challenges to the operator.

In February 2008, the Electric Reliability Council of Texas documented a power system incident that prompted them to respond to a big ramp down of wind generation by ramping up evening demand more quickly than predicted to maintain load/generation balance [16]. They drew on reserve power and relieved the load of power consumers who agreed to act as temporary curtailment loads during the event. After-incident analysis indicates that more precise forecasting of generation and demand may easily have averted the need for an emergency reaction. In September 2011, a heatwave in South Korea increased the electricity demand significantly. Due to a lack of available energy to satisfy the uptick caused by the heatwave, South Korea's power supply was disrupted for nearly 1.5 million people [17]. Although these scenarios are uncommon, they provide extreme illustrations of the potential repercussions of imbalance and, hence, the significant importance of accurate load forecasting.

In load forecasting, statistical and machine learning (ML) techniques have been applied. Deep learning techniques have gained popularity in recent years due to their ability to better interpret complex relationships in the data [3], [18]. The purpose of this work is to compare deep learning forecasting techniques against some conventional forecasters in use by specific utilities to determine if deep learning can better suit their needs. Overall accuracy and accuracy in peak detection were compared. The peak demand forecasts are critical for securing adequate generation, transmission, and distribution capacity. Accurate peak forecasts improve capital expenditure, decision making and system reliability.

## 1.1 Objectives and Contributions

This section includes a list of the objectives and contributions we hope to accomplish through this work.

* Deep learning techniques are considered, as they have demonstrated remarkable results when applied to various problems.
* Two types of models are implemented that have been determined to have the potential to produce promising results in load forecasting: CNNs and LSTMs.
* This contributes to the maturation of the ongoing debate over deep learning methods that have not been thoroughly tested in load forecasting.
* We will investigate the added value provided by the deep learning algorithms in comparison to conventional forecasters. We will compare both overall and peak detection accuracy.
* We plan to test deep learning techniques for their adaptability to external factors like annual increases in power demand or temperature shifts.
* We intend to develop forecasters that can adapt to complex data relationships without explicit user intervention.
* Lastly, because we used publicly available data, this work will be reproducible and serve as a benchmark for future research within and beyond our smart-grid team.

## 1.2 Outline of the Thesis

// To be filled

# 2 Overview of Load Forecasting

## 2.1 Factors that affect the load demand

Different factors can affect load demand, such as the region in question, the type of customers in the region, weather factors (e.g., temperature), the time of the day, day of the week, and other unpredictable factors (i.e., coronavirus outbreak). These factors can be classified into five distinct categories: historical load, economic, chronological, meteorological, and random.

### 2.1.1 Historical load

Because load at any given hour is reliant on the load at the preceding hour [19], historical load data is utilized as input to short-term load forecasting models. Additionally, the load may have connections with historical load at various lags, such as the prior week or load at the same time yesterday. As a result, the literature makes use of a variety of input variables. For example, Houimli et al. [20] forecasted the subsequent 48 half-hourly loads using the previous 48 half-hourly loads, Park et al. [21] used the previous two hours of load data to forecast the next hour, Bakirtzis et al. [22] used the previous two days of hourly load data to forecast the next day's hourly load, and Velasco et al. [23] used the previous 24 hours of load data to forecast the next. As demonstrated, the literature uses a wide variety of historical load input data, but there is agreement that some sort of historical load input is necessary.

### 2.1.2 Economic Factors

Economic factors include capital expenditure in the facility's infrastructure by constructing new buildings, laboratories, and experiments that increase the load to the electric grid. The location's funding profile dictates what equipment, processes, and experiments can be conducted and when. Demand pricing and demand management programs influence how customers use electricity during system peaking [24]. Economic factors will have little effect on short-term load forecasting because they usually affect consumption patterns over a more extended period [1]. However, economic factors might serve as an impetus for examining a system's load pattern and adopting load reduction strategies.

### 2.1.3 Chronological Factors

Seasonal, weekly, and daily cycles and holidays can influence load [24], [25]. Autumn and spring often have a lower load. Weekdays differ from weekends, with weekends having a lighter load [19], [25]. In general, public holidays have a lower load than weekdays and are more comparable to weekends. The load on days immediately preceding and following the holiday is also impacted [24]. Because time influences how electricity is used, it is incorporated into load forecast models using calendar data [26]. Some, or all, of the patterns, might be considered. The pattern of weekday-weekend/holiday can be explained by establishing distinct models for each category [27]. Alternatively, many indicator variables could be used [2], [23]. Weekends and holidays are particularly difficult for studies that do not differentiate between these days [28]. Other patterns are accounted for using variables such as the hour of the day, the day of the week, the month, and the week number [2], [20], [23]

### 2.1.4 Meteorological Factors

The most frequently used and most significant weather variable is temperature [19], [21], [24], [29], [30]. The majority of load forecasting models incorporate one or more temperature-related variables [31]. The relationship between temperature and load is non-linear. According to Hong and Shahidehpour [32], temperature factors alone can account for more than 70% of the variability in load. This nonlinear relationship contributes to the widespread use of nonlinear approaches for load forecasting [1], [19]. Since the early 1930s, the relation between temperature and load has been recognized [19].

Humidity, solar irradiance, wind speed, barometric pressure, and precipitation are other weather variables that might alter the electric hourly load profile. Days with high humidity require cooling equipment to operate at a higher duty cycle to remove surplus moisture from the conditioned air [33]. Long periods of high sun irradiation will radiantly heat the interiors of buildings, requiring cooling systems to run longer and with less diversity. Precipitation has a propensity to chill the air, hence decreasing the cooling load [24]. Wind speed and barometric pressure can also influence the hourly load profile and frequently do so in conjunction with other variables such as precipitation. Wind speeds may amplify the effect of low temperatures, resulting in a greater wind chill index as well as increased demand. Wind speeds greater than 15 mph generate renewable energy, reducing the reliance on central sources of generation.

Janicki provides an in-depth description of the many types of meteorological variables that are utilized in load forecasting, as well as instances of their application in the literature [34]. The efficacy of these variables in forecasting load varies according to geographic location, industry, and regional climate. Friedrich and Afshari [35] discovered that incorporating four meteorological variables (temperature, specific humidity, wind speed, and sun irradiation) produced more accurate findings than relying solely on temperature. This was not a direct comparison because the models were of two distinct types (ANN vs. transfer function). Taylor and Buizza [36] employ a variety of meteorological factors. They modify the model's temperature, wind speed, and cloud cover to employ effective temperature, wind cooling power, and lighting. They do not make comparisons to a model based solely on temperature. Khotanzad et al. [37] use an effective temperature to adjust for humidity and wind speed. Specific studies concentrating exclusively on temperature imply that additional weather variables could be incorporated to improve forecasts [2], [21].

For load forecasting, the location of the weather data input must be determined. Forecasting loads can be pretty location-specific, such as forecasting for individual buildings or local regions. This form of forecasting is possible due to the availability of smart grid data, although system load forecasting is still required. Utilizing weather data for a significant load center is one approach of selecting weather data (e.g., Toronto used for Ontario). Weather stations located throughout a region can also be averaged to provide a single input variable [38]. Additionally, weather station selection can be used to discover which stations are the most accurate predictors of load [39]–[41]. Distributed or multi-region forecasting is a technique for anticipating load by utilizing meteorological data from different locations [42], which is particularly useful in vast geographic areas.

### 2.1.5 Random Factors

Random factors affecting the electrical load profile are other random disruptions in the load pattern that cannot be described by the preceding factors [24]. These disturbances might include considerable loads that operate on an ad hoc basis, making prediction impossible [43]. Other disruptions, such as extensive employee absenteeism (due to illness, severe weather, etc.) and planned or unforeseen power system outages can substantially impact the facility's load profile.

## 2.2 Load Forecasting Horizons

Electricity demand can be assessed periodically - hourly, daily, weekly, monthly, or yearly and forecasting can be applied to various horizons: very short-term load forecasting (VSTLF, <1-day), short-term load forecasting (STLF, <2-weeks), medium-term load forecasting (MTLF <3-years), and long-term load forecasting (LTLF >3years) [10]. Short-term forecasting has been the focus in most current research, concentrating on horizons of less than two weeks [1], [10], [44]. Disparities in time horizons have implications for the models and methodologies used in forecasting, and for the available and selected input data. The operator must examine the most suited model type and the critical external factors that must be considered to obtain the most accurate forecast [29].

### 2.2.1 Very Short-Term Load Forecasting (VSTLF)

VSTLF generates forecasts for loads up to one day in the future. Throughout the power industry, utilities and grid operators typically use such forecasts for real-time scheduling of electricity generation, load frequency control, and demand response. Very short-term load forecasts are also critical to retailers, power marketers, and trading firms' operations. VSTLF contributes to the immediate balancing of supply and demand. Trading in power markets is another application that relies on this type of forecasting. VSTLF is frequently viewed as a subproblem of short-term load forecasting (STLF) because both can use weather forecasts as forecasting period inputs. However, to achieve high accuracy over a very short time horizon, it is necessary to recognize the practical distinction between VSTLF and STLF.

From a modelling perspective, VSTLF models can incorporate lagged load as an independent variable in addition to those commonly used in STLF, such as weather and calendar variables (ex…). VSTLF, from an implementation standpoint, requires the model to be estimated quickly to produce the forecast on time. Additionally, the short lead time complicates the data collection process [45]. The VSTLF literature has primarily focused on the modelling aspect. Researchers have experimented with a variety of techniques for forecasting the next few minutes to hours' load. Liu et al. compared five VSTLF techniques in [46]. Charytoniuk and Chen proposed an approach based on using a set of ANNs to model load dynamics rather than actual loads [47]. Taylor evaluated various methods for VSTLF using minute-by-minute observations of British electricity demand [48].

### 2.2.2 Short Term Load Forecasting (STLF)

According to Mandal et al. [49], STLF is a critical instrument in a utility system’s day-to-day operations and planning activities, such as energy transactions, hydrothermal coordination, unit commitment, calculating load flows, economic dispatch, security analysis, fuel scheduling, unit maintenance, and making decisions to avoid overloading. STLF estimates load up to two weeks in advance. STLF is a complex process influenced by various factors, including economic conditions, time of day, season, weather. The electricity demand is determined by meteorological variables, human social activities, and industrial activities.

This area has become increasingly important in recent years due to two main factors: deregulation of the power systems, which introduces new challenges to the forecasting problem, and the fact that no two utilities are identical, necessitating a detailed case study analysis of the various geographical, meteorological, load type, and social factors affecting load demand [50]. Hippert et al. [19] explain that forecasting short-term load becomes complicated when the load at a given hour is dependent not only on the load at the previous hour but also on the load at the same hour on previous days and the load at the same hour on the same denomination day in the previous week [51]. Additionally, the forecaster must model the relationship between the load and other variables such as weather, leisure activities, and so on.

### 2.2.3 Medium-Term Load Forecasting (MTLF)

MTLF is another type of load forecasting which operates on a longer timescale, ranging from two weeks to three years. The MTLF guides decisions about network operations, schedule maintenance, fuel procurement for power plants, capacity planning and infrastructure development, and financial budgeting [52]. Additionally, MTLF enables a company to forecast load demand over a longer time, which can aid in negotiations with other companies. Demographic and economic factors influence MTLF. MTLF typically produces the daily peak and average load [53], [54]. MTLF and STLF have a close association; long-term decision-making must be integrated into short-term decision-making. This coordination between different decision-making levels is critical to ensure that specific operational objectives that develop in the medium term are explicitly considered in the short term [55]. Additionally, coordination between decision-making levels has become critical for generation businesses seeking to boost their profitability.

### 2.2.4 Long Term Load Forecasting (LTLF)

LTLF is the final type of load forecasting. LTLF covers a period of more than three years. LTLF is required for planning purposes, such as constructing new power plants, expanding the transmission system, and electric utility expansion planning. There are indicators affecting LTLF in terms of demographic and economic development. The population growth, industrial expansion, local area development, gross domestic product, and annual energy consumption in the past are all factors to consider. Annual peak load demand and annual energy demand for the years ahead are the outputs of the LTLF [56]. Although these longer-term forecasts receive less attention than their more visible short-term counterparts, their inaccuracy has significant financial consequences. It may result in either wasted investment in new generation facilities or a shortage of supply capability when there is an under-forecast.

## 2.3 Overview of Load Forecasting Techniques

### 2.3.1 Statistical and Machine Learning Techniques

Statistical techniques and machine learning (ML) have both been used to forecast load, and with the widespread adoption of data science, the line between these two approaches is becoming increasingly ambiguous [1]. Examples of statistical techniques applied to electrical load forecasting include multiple linear regression analysis [57], [58] exponential smoothing [59], [60], and auto-regressive integrated moving average (ARIMA) modelling [61], [62]. On the other hand, ML algorithms are more intelligent and can be better, as they provide the capacity to learn and adapt to the non-linear and complex relationships between load and other influencing factors (e.g., weather, time of day) automatically [10]. Examples are Artificial Neural Networks (ANNs) [63][64], Fuzzy Regression Models [65], [66], Support Vector Machines [67], Gradient Boosting Machines [68]; they have all been applied to electrical load forecasting.

The authors of [8] discussed many regression-based approaches for STLF [69]. Another study [70] examined various Multiple Linear Regression (MLR) algorithms for load forecasting. The disadvantage of MLR techniques is that they require external factors such as temperature and time of day. The ARIMA model is the most frequently used among all regression models since it consistently produces good prediction results; for example, the author in [71] combined ARIMA and Box-Jenkins methods to do hourly forecasting.

### 2.3.2 Deep Learning Techniques

Deep learning approaches have had remarkable success in the last few years at handling complex sequential data [72], [73]. As a result, deep learning approaches have been effectively used to load forecasting applications, where they have been shown to outperform a variety of benchmark models, including simple ANNs and standard statistical time series methods such as ARIMA [74]. With improved computational power, more datasets, and the granularity of available data, deep learning models are expected to dominate the load forecasting field. Deep learning approaches like the recurrent neural network (RNN) [11], long-short-term-memory network (LSTM) [12], and the 1-D convolution neural network (CNN) [3], [7] have become enticing to researchers in this field, primarily because of their ability to learn about temporal dependencies in data inputs, and their ability to quickly adapt to abrupt changes in load patterns, as they occur.

The authors of [75] examined seven distinct models using three real-world data sets and demonstrated that deep learning methods could be employed in load forecasting applications in place of more traditional mathematical techniques such as ARIMA. The authors of [13] offered a novel parallel model that is a combination of convolutional neural networks (CNN) and recurrent neural networks (RNN). Additionally, because RNNs employ control theory in their structure, they can determine the relationship between old and new data, making them an appealing network for load forecasting applications in recent years. Regarding how RNNs work, the authors in [76] conducted an appropriate study on these networks. Similar to [13], the authors of [77] presented a mix of long short-term memory (LSTM) and convolutional neural networks (CNN). The proposed model’s performance in load forecasting was more stable than that of other machine learning techniques. Similarly, the authors of [64] suggested a new Deep-Energy model that combines a 1-D CNN for feature extraction with a fully connected network for forecasting future load data. They forecasted the data for the next three days using an hourly electricity consumption data set from the United States [78]. During the training phase, data from the preceding seven days was used. They compared the proposed model’s performance to five different machine learning techniques using the Root Mean Square Error (RMSE) and Mean Absolute Percent Error (MAPE) metrics. The findings indicated that the Deep-Energy model could make accurate short-term load predictions than the other models.

In another paper [79], the authors presented a new model that incorporates three algorithms: Variational Mode Decomposition (VMD), Convolutional Neural Networks (CNN), and Gated Neural Networks (GRU), and named it SEPNet. This model was created to forecast hourly power prices, and to assess it, hourly data from the city of New York, USA was used. The data set included hourly electricity prices from 2015 to 2018. Compared to other models such as LSTM, CNN, and VMD-CNN, the SEPNet model fared better, improving the RMSE and MAPE by 25% and 19%, respectively. Additionally, several writers, for example [80], employed ANNs to forecast other load data types, such as photovoltaic system output data. They proposed a robust CNN-based model named PVPNet and assessed it using daily data from 2015.

### 2.3.3 The Myth of Finding the One Size Fits All Technique

Tao Hong spoke about the myth of finding the best technique [1]. He concluded that it is essential that researchers and users know that a universally best technique does not exist. The approach applied to load forecast should be based on forecasting needs and the dataset being analyzed. It is not likely that one approach will be helpful in all load forecasting scenarios. Different algorithms perform better or worse with different datasets. Furthermore, forecast errors differ significantly for different utilities, utility zones, different horizons, etc.

## 2.4 Description of the Benchmark Techniques

Many publications lack detailed information about their experimental set-ups, making conducting direct comparisons with reported results challenging. The benchmark algorithms proposed for this work have been selected because they are relevant and because they are sufficiently well documented to reproduce [1], [4]–[6], [81]–[83].

### 2.4.1 The Seasonal Naïve Forecaster (SNF)

The naïve forecaster is a simple forecaster based on a random walk model [84]; it has often been implemented as a ground-level benchmark for developing more sophisticated forecasters [81], [85]–[87]. It is used to demonstrate how much value is added by forecasters under comparison – when a naïve forecaster outperforms a more complex forecasting model, we know that the complex model offers little value. Bracale [86] et al. point out 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 forms the basis of the naive forecaster. The Seasonal Naïve Forecaster (SNF) improves the naïve forecaster by considering seasonal trends [88]. The SNF can be expressed by the simple mathematical relationship shown in (1):

where  is the time series, and is the seasonal period (for hourly data, m=24 if we take the hourly sample from the day before). The naive formula takes the last observed value as the future value, while the seasonal naive formula takes the value from the previous season. In this work, we used the previous week’s lag for the SNF forecaster. The SNF forecaster is excellent for making short-term forecasts of variables that are generally stable or vary consistently. However, it is highly ineffective at forecasting time series data that fluctuate significantly or are susceptible to irregular elements such as temperature [85].

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

Multiple linear regression (MLR) is one of the most commonly used statistical techniques for load forecasting [44], [58], [65], [70], [81], [89]–[93]. MLR forecasters model the relationships between a continuous dependent variable and one or more independent variables. An MLR with two independent variables can be expressed mathematically as:

In the case of load forecasting,  is the load, and  are independent variables such as temperature and time-of-day, s are coefficients estimated, and is an error term. The error term  is typically assumed to have a mean of zero and a constant variance [58]. MLR models are fitted such that the sum-of-squares of differences of actual and forecasted values are minimized. MLRs’ accuracy is determined mainly by the relationships between the data and the independent variables included. Amral et al. state in [94] that multi-linear regression models for short-term load forecasting are relatively simple to develop and maintain. Moreover, MLRs primary shortcoming is its reliance on the accuracy of previously recorded load and temperature data, which considerably impacts the predicted output. Increasing the number of relevant independent variables generally improves predictive accuracy, but it eventually reaches a point where the improvement is negligible. However, while MLRs can simulate non-linear relationships, they cannot do so without explicit user specifications [95]. Additionally, MLRs are incapable of intelligently learning and adapting to data changes caused by newer factors.

#### 2.4.2.1 Assumptions of the MLR forecaster

1. The dependent variable and each of the independent variables should have a linear relationship.
2. Correlations between any of the independent variables are low. Multicollinearity exists when various variables are correlated with each other and with the dependent variable. When independent variables exhibit multicollinearity, obtaining the variable that contributes to the variance in the dependent variable can be difficult.
3. The residuals have a constant variance. The magnitude of our forecast error does not change much while the independent variable’s value changes.
4. Observations are autonomous. The MLR model presupposes that all observations are independent of one another; in other words, the residuals values are also independent of one another.
5. The data is normally distributed.

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

In 1970, Box and Jenkins proposed the autoregressive integrated moving average (ARIMA) forecaster. Therefore it is also known as the Box-Jenkins model [96]. The ARIMA forecaster is arguably one of the most popular and commonly utilized statistical forecasting techniques for load forecasting [97]. The ARIMA model seeks to explain data by utilizing time-series data on previous values and making linear regression predictions. It allows regression techniques to be applied to non-stationary data. If the data contains a trend, it is said to be non-stationary.

As the name implies, this family of techniques consists of three main components: a) an “autoregression” portion that models the series’ relationship with its past values (lagged values); b) a “moving average” portion that model the forecast as a function of past forecast errors (lagged forecast errors); and c) an “integrated” portion that makes the series stationary. A lag feature is a fancy phrase for a variable that holds data from earlier time steps. Lags are essential in time series research because of a phenomenon known as autocorrelation. For example, if we want to anticipate the demand for today t, we can use the demand from yesterday t-1 as a feature.

The term “AR” in ARIMA stands for autoregression, suggesting that the model is dependent on the relationship between the present values of the data and their previous values. In other words, it indicates that the data has been regressed against its previous values. The letter “I” stands for integrated, indicating that the data is stationary. Stationary data is time-series data that has been stabilized by subtracting the observations from the prior values. The term “MA” refers to a moving average model, which indicates that the model’s forecast or outcome is linearly related to its historical values [97]. This implies that forecasting errors are linear functions of previous errors.

Each AR, I, and MA part is included in the model as parameters p, d, q, respectively. Specific integer values are assigned to the parameters to denote the ARIMA model type. The ARIMA model is denoted by ARIMA (p, d, q). The parameter p denotes the number of autoregressive terms or “lag observations”; it is also called the “lag order” because it influences the model’s output by giving lagged data points. The parameter d is the degree of differentiation; it specifies how many times the lagging indicators have been subtracted from the data to make it stationary. Differencing is required since linear regression models work better when applied to stationary signals [93], [98]. The parameter q denotes the model’s forecast error and is often referred to as the size of the moving average window. The result is an estimate based on a linear combination of weighted differentiated lagged values and lagged errors as delineated in (3) [99]–[101]:

Here  is estimated to account for the average change between consecutive observations, the lag operator is the nth differentiated lag value of the time series, is the nth lag error of the time series. Generally, the error terms are assumed to be independent, uniformly distributed variables taken from a normal distribution with a mean of zero. and respectively are the parameters of the autoregressive and moving average parts; they represent the nth coefficients of that lag term estimated by the model to minimize the error.

In [102], Fernandez et al. forecasted energy load for non-residential buildings using an ARIMA model, a polynomial model, a neural network model, and a support vector machine model. The study analyzed energy consumption data from Spain’s University of Deusto in Donostia-San Sebastian. The goal was to forecast six days in advance at hourly intervals. Compared to the other models, the ARIMA model had the highest accuracy among all the models. Additionally, the authors noted that the ARIMA model runs 200 times quicker than the Support Vector Machine model because of the lower number of parameters. The ARIMA model is mainly dependent on the quality of historical data and data differencing. It is critical to verify that data collection was reliable and extensive for the model to produce accurate results and forecasts. While ARIMA models can be accurate and dependable under the right conditions and with sufficient data, one of the model’s primary disadvantages is that the parameters (p, d, q) must be manually set. These numbers may vary slightly among datasets and forecast horizons. Therefore, finding the best accurate fit can be a lengthy trial-and-error process.

### 2.4.4 Artificial Neural Networks (ANNs)

Human brains are uniquely capable of comprehending the context of real-world situations in ways that machines cannot. Neural networks were designed to overcome this issue. Artificial neural networks were founded on the work of McCulloch and Pitts in 1943 [103], who developed a binary unit whose value is determined by the linear sum of the network's weighted inputs. Another seminal development occurred in 1949 when Hebb [104] proposed a learning rule stating that neuronal connections are adaptable and can be reinforced through the frequent activation of a neuron by another. The feed-forward network was invented by Rosenblatt's [105] perceptron network with signals connected in a single direction. Backpropagation [106] was a significant development that enabled multilayer perceptron network training.

An artificial neural network is an effort to imitate the network of neurons that comprise the human brain to enable the computer to learn and make decisions in a similar way to humans [6], [107]–[109]. ANNs are built by programming conventional computers to act like interconnected brain cells. ANNs are modelled like the human brain in that it learns the relationship between inputs and outputs via experience. A neural network is made up of neurons. The primary neuronal workflow can be separated into the following components, as illustrated in Figure 1. A neuron gets two inputs x1 and x2, each of which has a unique weight, w1 and w2, reflecting its relative importance. Each neuron calculates the weighted total of those inputs and adds a bias b that is unique to it. Following that, the result is subjected to the activation function. Finally, the output of this activation function is the neuron’s final output.

Diagram

Description automatically generated

Figure - An artificial neuron’s workflow

The activation functions of an ANN are critical because they enable the solution of non-linear problems. Figure 2 shows some frequently used activation functions. The activation function has to be non-decreasing but differentiable [2], [110], as the backpropagation algorithm computes the error function's gradient. Linear transfer functions are typically used on neurons in the output layer, whereas tanh transfer functions are typically used on neurons in the hidden layer. If the neuron’s workflow does not include activation functions, an ANN will perform similarly to a linear regression model [110]. Neurons in an ANN can be classified into three layers, as seen in Figure 3: input, hidden, and output.

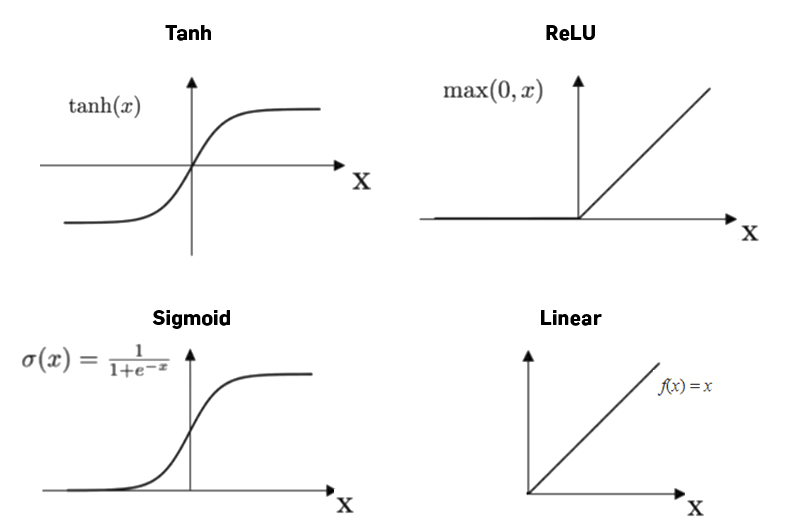


Figure - Examples of the most frequently used ANN activation functions [111]

The multilayer perceptron (MLP) is one of the most well-known and widely used artificial neural networks due to its universal approximation capability and ability to scale well with input dimensions. It is a feed-forward network architecture with one or more hidden layers of connected neurons, as seen in figure 3. Each layer is connected in a single direction, and there are no connections between layers; thus, the term feed-forward. The network is presented with model inputs in the first layer, and the neuron inputs in subsequent layers are the outputs of all preceding layer neurons. The MLP is trained using a supervised learning algorithm in which each sample vector represents a set of inputs with desired outputs. The function that maps the inputs to the outputs should be written in such a way that it is generalizable to previously unseen inputs [103]–[106].

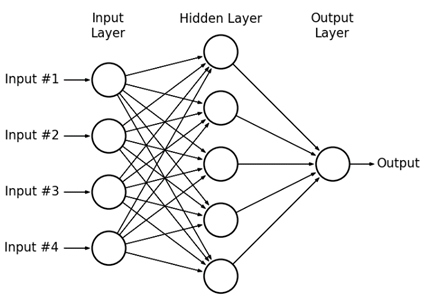


Figure - The Structure of a simple feed-forward ANN [112]

The number of layers in an ANN is calculated as the sum of hidden and output layers. The input layer is nothing more than a vector representation of the input. Similarly, the number of neurons may be estimated by deleting the input layer and counting only the neurons in the hidden and output layers [2], [113]. An ANN’s operation can be characterized in terms of cycles, each of which consists of two phases: forward propagation and back-propagation. The forward propagation phase proceeds from left to right, intending to produce an output for specific input. When the final output of an ANN is produced, it is critical to calculating the error or how far off the target value is from the final output; this is the cost or loss function’s assignment. A Mean Square Error (MSE) is one of the available metrics in a regression problem. In this scenario, an optimization procedure such as gradient descent assists in minimizing this loss function by guiding it in the appropriate direction toward the function minimum.

Training an ANN entails fine-tuning its parameters: weights and biases. An ANN begins by randomly generating those parameters. After calculating the cost function, the network parameters are modified according to an optimization technique during the back-propagation phase [9]. Typically, training instances are handled in batches, which speeds up the training operation due to the time required to calculate the gradient descent and update the network’s parameters. The back-propagation phase proceeds from right to left. Back-propagation is a technique for training a network that involves computing the error signal at the output then propagating it back through the network layers. Using the chain rule, it calculates gradients for every hidden and output layer neuron to determine their error sensitivity.

The capacity for generalization is one of the critical properties of ANNs [18]; this means that the ANN should not remember the data on which it was trained but rather grasp, extract, and learn the patterns, trends, and dependencies to perform well with new, previously unseen data [6]. Neural networks have produced excellent results in load forecasting [1], [6], [9], [68], [91], [107]. They have gained popularity due to their capacity to discover complex and non-linear correlations from historical data, which is extremely difficult to do using statistical techniques. Adya and Collopy [114] draw two major findings from their evaluation: they demonstrated that neural networks have the capacity for prediction and that neural network research must be validated through comparisons to alternative methodologies. Zhang et al. [115] evaluated the use of neural networks in load forecasting and demonstrated that while neural networks could deal with huge amounts of historical load data with non-linear features, they neglected the linear relationship between the data. In [116] and [117], Papalexopoulos et al. developed a neural network-based approach in addition to a regression-based approach. Both models were validated using training data from 1986 to 1990 on peak and hourly loads for 1991. It was shown that the ANN model enhanced forecasting accuracy for both peak load and hourly forecasts.

#### 2.4.4.1 How Artificial Neural Networks Handle Overtraining

Tweaking the weights and biases, increasing the hidden layer's size from a single neuron to many neurons, and increasing the number of explanatory factors will continuously enhance a network's capacity to predict the training dataset correctly. This will eventually result in over-training and a decline in the network's capacity to generalize to previously unseen data. In other words, the algorithm has discovered properties of the training data that are absent from the test set. The cross-validation technique is used to avoid overtraining. The training set is separated into two distinct halves. For example, if three years of data are provided, they are separated into two years and one-year sets.

The first set is used to train the ANN, while the second set is used to validate the learned model after a few hundred runs over the training data. The validation set’s error is evaluated. Typically, as the number of runs over the training set increases, this error reduces until the ANN is over-trained, as shown by a rise in this error. As a result, when the error on the validation set begins to grow, the training is stopped. This approach generates the necessary number of epochs throughout the training set. After that, the whole three years of data is used to re-train the MLP with this number of epochs.

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

One of the most popular ML-based load forecasters is the ANNSTLF [1], [82], [93]. The ANNSTLF-G3 has improved prediction accuracy and generated economic benefits for over a dozen utilities [95], [118]. Some publications have named ANNSTLF-G3 the best forecaster for short-term load forecasting [1], [93]. The configuration of this load forecaster has undergone a few revisions since it was first proposed [119], [120], and we will discuss the third-generation design (G3) [37], which uses two shallow multi-layer feed-forward ANNs together with a recursive least squares (RLS) combiner to predict short-term load. The figure below shows the block diagram of the system:

Diagram, schematic

Description automatically generated

Figure - The Block Diagram of the third generation ANNSTLF [37]

Both ANN blocks are multi-layer perceptron trained with the error back-propagation algorithm. The base-load forecaster (BLF) is trained to forecast regular next-day load, while the change-load forecaster (CLF) is trained to forecast changes in the load demand from one day to the next. The two ANN forecasters complement one another since the BLF places a stronger emphasis on normal load patterns, while the CLF places a stronger emphasis on yesterday’s load. Combining these two independent forecasts improves accuracy. This is especially true in instances of abrupt load changes brought about by weather changes. The BLF has a proclivity for a delayed response to sudden changes in load.

On the other hand, because the CLF uses yesterday’s load as a baseline and forecasts future changes in that load, it responds more quickly to changing conditions. It is argued that the CLF forecaster allows the model to rapidly adapt to abrupt changes in temperature [93], [121], [122]. Both blocks are presented with the same 79 inputs (see Figure 4) and output a 24x1 vector representing hourly forecasts. The CLF sums predicted changes with actual last-day values to produce its output. The final forecast is based on a weighted average of each block’s outputs, with the weights adaptively determined using an RLS algorithm. ANNSTLF-G3 forecasts are typically produced one day at a time (24 hrs.). To extend the forecast horizon beyond one day, the previous day’s forecast load is substituted for the actual load to produce the next day’s load forecast. According to the paper’s authors, the ANNSTLF-G3 algorithm performs best when the hidden layer contains between 30 and 60 neurons and is trained using two to three years of data.

# 3 Investigation

This work aims to determine whether deep learning approaches can improve forecasting accuracy for data sets by comparing the accuracy of deep learning forecasters to some of the current forecasters used by utilities. This work will focus on STLF horizons. []

A significant strength of this work is implementing the research to three distinct datasets, which provides substantial data for evaluating the methods and a solid foundation for this study's conclusions. Two sets come from an Independent Electrical System Operator in Ontario and have been included because the data is publicly available, which helps with the reproducibility of this work. One set is from Ottawa [123], and the other is from Toronto [123], and they both consist of city-wide load aggregation measurements taken hourly, spanning ten years from 2010-2019.

The third set comes from Saint john Energy, a municipally-owned utility reseller. This data is included because the work proposed here supports efforts in a larger Smart Grid Technologies project underway at UNB, which partners with that utility reseller. The Saint John Energy data set is smaller than the others, spanning about 3.5 years, from 2018 to the present, but otherwise matches the hourly measurements with city-wide Saint John load aggregates.

In some parts of this work, weather data (temperature) obtained from Environment Canada [124] will augment the time-series data. The training dataset for both the Toronto and Ottawa datasets are from the years 2010-2018, while the testing dataset is from 2019. The training dataset for the Saint John dataset is from the years 2018 - 2020, while the testing dataset is from January 2021 until present. Megawatts were used to quantify the load demand.

A Hampel filter was used to find and replace outliers in the datasets. The filter is a sliding window with a variable width that glides across the time series. The filter determines the median and standard deviation for each window. If a point in the window is more than three standard deviations from the window's median, the Hampel filter flags it as an outlier and substitutes it with the window's median [125]. []

Four benchmark forecasters will be used for comparison: a Seasonal Naïve forecaster, a Multiple Linear Regression (MLR) forecaster, an Auto-Regressive Integrated Moving Average (ARIMA) forecaster, and a forecaster based on a shallow Artificial Neural Network (ANN). These benchmark algorithms have been available for many years and have been implemented and used by researchers and utilities [1], [4]–[6], [81]–[83]. We began by implementing each of the benchmark algorithms. Then, the CNN and the LSTM forecasters were implemented. Finally, the performance of the deep learning forecasters was assessed by comparing them against the performance of the benchmark algorithms, using the data sets available. Section 3.2 contains the details of the implementation.

## 3.1 The Deep Learning Techniques

According to Yann Lecun and colleagues, “deep learning enables computer models built of many processing layers to learn representations of data at different levels of abstraction” [126]. Although this kind of model has existed for a long period, it has only recently garnered widespread popularity. A pivotal moment in developing deep learning occurred in 2006 when Geoffrey Hinton et al. [127] proved that a greedy layer-wise pre-training strategy enabled the construction of deep belief networks (DBNs). Geoffrey Hinton’s quantum leap in inventing an effective neural network training algorithm paved the way for deep learning implementations [128]. Previously, the utility of deep architectures had been limited by their proclivity to become stuck in suboptimal solutions. Since then, research has established that the same idea holds for different types of networks [129], ushering in the deep learning era.

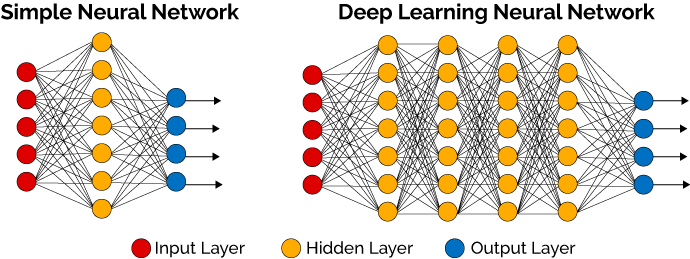


Figure – A simple network versus a deep learning network [130]

Goodfellow et al. [131] examine the concept of expanding network depth to improve generalization error rather than merely increasing the size of a single hidden layer. Numerous neurons may be required to accurately represent input data in a one-layered network, which is solved more effectively by shifting to a deeper design. Deep learning's success has been linked to the numerous levels of representation introduced by multiple layers [132]. By providing more powerful outputs than conventional benchmark neural networks, deep learning has revolutionized research fields like image processing and sequence learning. Their popularity is primarily due to their success in resolving a wide variety of previously believed unsolvable issues using shallow networks. Although the concept of ‘deep learning’ has been bandied about for decades, it was frequently dismissed as a fanciful notion rather than a feasible technology. This was primarily due to three constraints: (i) insufficient training data, (ii) insufficient processing power, and (iii) insufficient efficient training algorithms [133]. With improvements in the semiconductor industry resulting in powerful graphics processing units (GPUs) and the rising digitization of the world, these limits have been overcome.

Deep learning models have grown in popularity during the last several years in fields such as computer vision, speech recognition, machine translation, and board game programs, where they have delivered results comparable to expert human performance, if not beyond it [134]. Reinforcement learning has also benefited from the revolution in deep learning. Mnih et al. [135] experimented with demonstrating that deep networks may be trained to perform at a professional human level when playing computer games. The significant benefit of deep learning models over traditional models is that they acquire high-level features incrementally from data, eliminating the Need for topic knowledge and time-consuming feature extraction [136]. The primary reason for utilizing deep learning models in this study is that they are superior to the conventional models in terms of their ability to (i) learn extremely non-linear relationships and (ii) learn shared uncertainties.

Deep learning is a class of networks that encompasses a variety of architectures. Deep neural networks, recurrent neural networks, long short-term memory networks, deep belief networks, and convolutional neural networks are the most prevalent. In discussion, the terms deep learning and neural networks are frequently used interchangeably, which can be confusing. As a result, it is worth emphasizing that the term “deep” refers to the number of layers in a neural network. As illustrated in Figure 5, a neural network with more than three layers—including the inputs and outputs—is referred to as a deep learning technique. A neural network with only two or three layers is referred to as a simple neural network. The annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a competition for computer vision. For the first time, a Convolutional Neural Network (CNN) won this competition in 2012, substantially lowering the error rate from 26.1 percent to 15.3 percent [137]. Since then, deep learning models have reduced the error rate to 3.57 percent by employing residual nets with a depth of up to 152 [138]. The authors ascribed the network's strong performance to its depth. Other competitions have been won by CNNs, including the ICDAR Chinese handwriting competition [139], the ISBI image segmentation competition [140], and the MICCAI Grand Challenge on cancer detection from medical pictures [141].

Deep learning has demonstrated favourable results in a variety of domains, not just image identification. They have also had a significant impact on speech recognition. Dahl et al. [142] and Seide et al. [143] transcribed voice data using DBNs. Dahl et al. [142] discovered that increasing the depth of their model from one to eight hidden layers consistently increased performance. Additionally, their approach outperformed earlier models generated for the same dataset by around 2%. Abdel-Hamid et al. [144] utilized CNNs for speech recognition. In comparison to deep belief HMMs, a CNN lowered the error rate on a benchmark phone call dataset by 6% – 10%. Deng et al. [145] predicted the same dataset using an ensemble deep learning approach, which improved the accuracy of single CNNs by around 1%.

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

Recurrent Neural Network (RNNs) introduced memory into neural networks, which helps to model sequential data. RNNs have been successfully applied in machine translation, speech synthesis, and time series prediction [146]. Typically, back-propagation or real-time recurrent learning algorithms are used to train RNNs. These training methods expose traditional RNNs to vanishing gradient issues, reducing their effectiveness when dealing with large data sets [7], [14], [108]. The LSTM is an RNN created to fix vanishing gradient problems and store information for long periods. Its memory cell configuration helps retain information more than any deep neural network currently available [108]. LSTM is perhaps the most well-known deep learning architecture for time series forecasting, which is built specifically to remember past data to retrieve it at a suitable time in the future to produce the output prediction.

LSTMs are a classification of recurrent neural networks that can learn the order of dependencies between elements in a sequence. LSTMs overcome the issue of vanishing gradients using gates that regulate the input flow, making them excellent for dealing with time series data with lengthy temporal dependencies. Unlike a conventional recurrent unit, which overwrites its memory at each time step, the LSTM unit can select whether to retain existing memory via the introduced gates. The LSTM provides a model capable of storing information for an extended period and better control of gradients. Munem[108] et al. argue that LSTM is better than other deep neural networks because of its memory cell configuration.

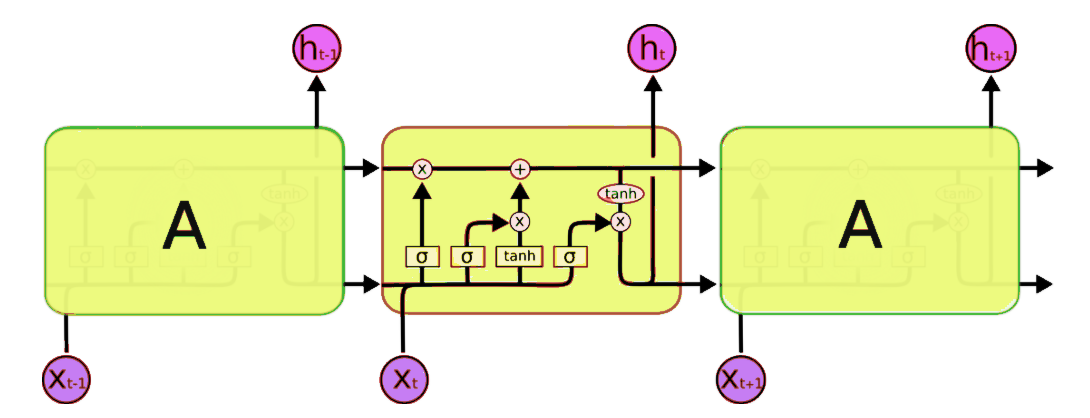


Figure - The Long Short-Term Memory Structure [147]

This repeating module in standard RNNs will have a relatively simple structure, such as a single tanh layer. LSTMs have the structure as well, although the repeating module is structured differently. Rather than a single neural network layer, there are four that interact in a unique way. Each memory block is composed of the following: a memory cell, an input gate, a forget gate, and an output gate. Each line in Figure 6 represents a whole vector, from one node’s output to the inputs of others. The pink circles denote operations performed at the point level, such as vector addition. The tiny yellow boxes represent layers of learned neural networks. Concatenation occurs when two lines merge, whereas forking occurs when a line’s content is replicated, and the copies are sent to various locations.

The key to LSTMs is the cell state, represented by the horizontal line running across the diagram’s top. It maintains the integrity of data travelling through it. By adequately regulating gates, the LSTMs can remove or add information to the cell state. Gates typically allow information to pass through on an optional basis. They are constructed using a sigmoid neural network layer and pointwise multiplication. The sigmoid layer generates values between 0 and 1, indicating how much of each element should be allowed to pass through. A number of zero indicates that “everything is forgotten,” whereas a value of one indicates that “everything is retained.” Three gates protect and govern the cell state in an LSTM [7].

The initial stage in LSTM is for a sigmoid layer dubbed the “forget gate layer” to decide what information should be discarded from the cell state. It examines the preceding hidden layer and input and returns a number between 0 and 1 for each number in the cell state. The following step is to decide what new information will be stored in the cell state by merging two pieces to make a state update. The first is that a sigmoid layer known as the “input gate layer” determines which values need to be updated. The second is that a tanh layer generates a vector of new candidate values that could be inserted into the state. Following that, multiplying the old state by forgetting the items and adding the new candidate’s values to update the old cell state into the new cell state.

Finally, the net executes the output, a filtered version of our cell state [148], [149]. First, a sigmoid layer uses the cell state to execute outputs. Then we run the cell state through tanh and multiply it by the output of the sigmoid gate to output only the sections we want. There will be no improvement in the state cell memory if the input gate value is minimal and close to zero. In a network model, stacked LSTM can be implemented by using multiple LSTM layers [7]. The technique of forgetting and retaining information within a cell makes LSTM perfect for dealing with sequential data. Bouktif et al. [150] are one of the authors who applied the LSTM to load forecasting. They examined half-hourly French electricity demand from 2008 to 2016. 70% of the data has been used to train the model, while 30% was used as the test set. A genetic algorithm was used to determine the optimal time lags to include in the input vector and the appropriate amount of stacked LSTM layers. The final structure used six LSTM layers with 100, 60, and 50 cells, and 100-time lags were transmitted into the input layer. The test data revealed a mean absolute error of 250 MW and a root mean square error of 341 MW. []

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

In recent years, Convolutional Neural Networks (CNNs) have gained the attention of researchers studying load forecasting [3], [13], [74], [151], [152]. CNNs are a type of deep learning network used for data processing with a grid-like topology [3], [146], [153]. This can comprise time series and image data, which can be viewed as a one-dimensional and two-dimensional data grid, respectively [3], [153]–[155]. CNN is like the ANN in that it is a feed-forward neural network designed to mimic human neurons [3], [126]. They have been successfully applied in computer vision, audio processing, activity recognition, natural language processing, drug discovery, video recognition, and time series forecasting, among other applications [7], [156]–[161]. In load forecasting, CNNs are known to boost the power of the ANN because they have deeper layers and have model parameters such as a receptive field length and dilation, which can help interpret load data better [7], [75]. In at least one of its layers, CNN employs a particular linear mathematical technique called convolution [146].

Convolution is performed in CNNs by repeatedly applying filters or kernels to the input data to build a feature map. CNNs are used to extract a large number of features. As a result, a CNN may perform the convolution process multiple times in each network's convolution layers. The number of times the convolution process is performed is determined by the number of filters in the layer, which the operator can specify. Each kernel will focus on a distinct feature of the input data. The convolutional layer performs three distinct actions. The feature map is created because of the first procedure mentioned above.

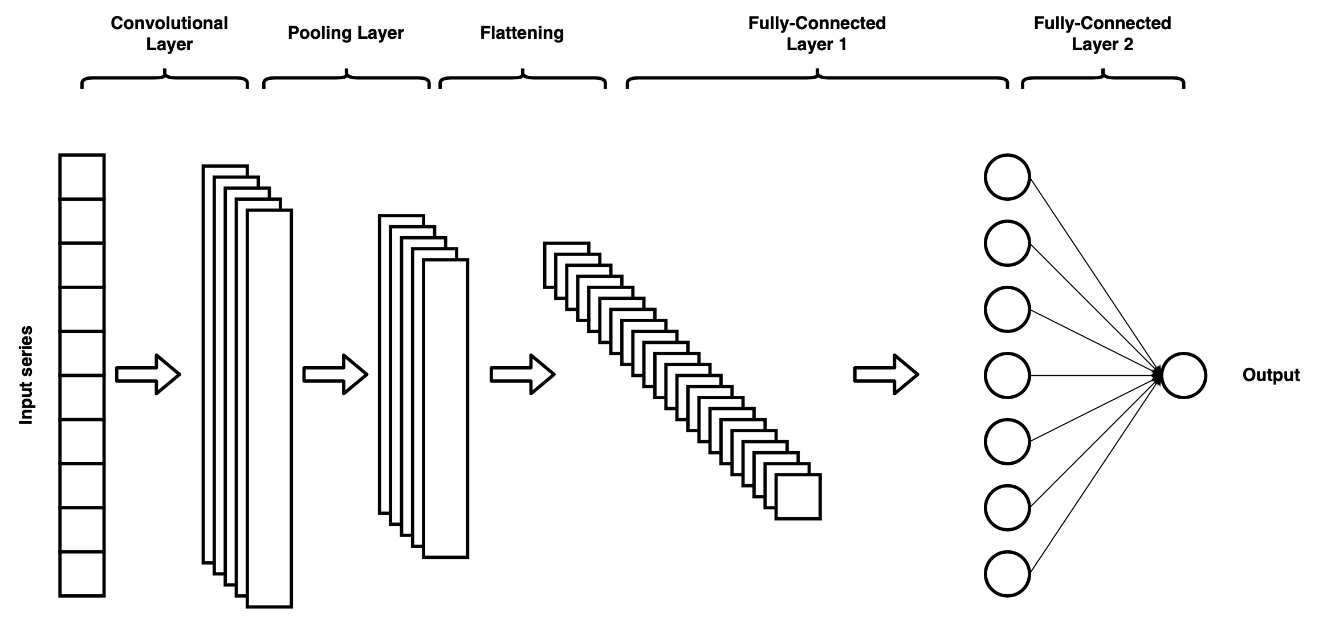


Figure - An Architecture of a one-dimensional CNN for time series data [162]

The second stage involves activating the elements in the feature map using a non-linear activation function, most commonly a ReLU or rectified linear activation function. Convolution is a linear process in and of itself. The rectified linear unit (ReLU) activation functions utilized in the convolutional layers introduce non-linearity. ReLU is a linear piecewise function. Because they behave similarly to linear functions, they are simple to create and train. When non-linear activation functions are utilized, propagating errors through multiple layers of a network frequently results in the so-called "vanishing gradient" problem, which inhibits deep networks from learning effectively. This is overcome by employing an activation function with similar qualities to that of a linear function. Similar to the sigmoid activation functions, the ReLU activation function squashes the inputs z, clamping negative values to zero, as shown in figure 8.



Figure – The Rectified Linear Unit Activation Function [163]

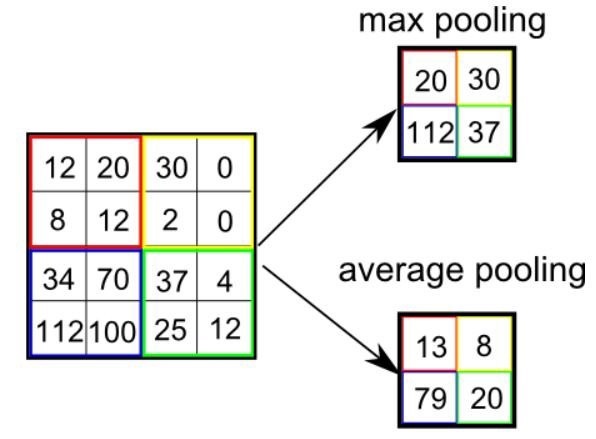


Figure – Examples of Max and Average Pooling [164]

The third stage employs a pooling procedure to smooth and minimize the dimensions of the resulting feature map. The max-pooling method is commonly used; it returns an array of the maximum output values within the previous layer’s rectangle neighbourhood [146]. Pooling layers are used to lower the size of the previous layer's output. A single value represents the pool from the output of a specified pool of neighbouring neurons from the preceding layer. In other words, the pooling layer aggregates the responses from individual areas into a single value. For instance, as seen in Figure 9, a max-pooling operation keeps the highest value inside a region as the item to pass through to the next layer. As a result, the following layer processes fewer inputs, increasing computing efficiency.

One or more convolutional layers may be present in the CNN network. After the convolutional layers generate their outputs, the hidden or fully connected layers receive them. The output layer is positioned immediately after the hidden layer and serves the same purpose as an output layer in a typical neural network. When Amaradinghe[3] et al. compared the CNN with the LSTM, SVM, ANN, and other algorithms for individual building level load forecasting, they concluded that CNN is a viable technique that produces accurate load forecasts.

## 3.2 Specifications for the Algorithms

The seasonal naive method was straightforward to implement; we used the previous week's hourly lag as the current hour's value. Additionally, we repeated the preceding step for each hour of our test set. In the ARIMA model, these hyperparameters were used for the Toronto, Ottawa, and Saint John datasets, respectively: (24, 2, 25), (23, 2, 24), and (24, 2, 24).

### 3.2.1 The MLR Forecaster

The MLR forecaster was developed using ten independent variables, also referred to as inputs, and one target variable, which is the actual demand at a particular hour. The independent variables are Temperature, Hour of the day, Month of the year, day of the week – Sunday is the first day of the week, Weekend Indicator – one or zero, Maximum hourly demand from the previous day, Minimum hourly demand from the previous day, Average hourly demand from the previous day, Hourly lag from the previous day, Hourly lag from the previous week.

### 3.2.2 The ANNSTLF-G3 Forecaster

The resilient back-propagation algorithm is used to train the BLF and CLF networks. According to the MATLAB handbook, this training method is effective and is frequently used for pattern recognition problems [165]. Additionally, we observed that this training method outperformed the Levenberg-Marquardt back-propagation method. The hidden layer is comprised of 60 neurons.   In the hidden and output layers, the activation function is a hyperbolic tangent sigmoid transfer function. We observed improved results when we changed the activation function of the output layer from linear to tangent sigmoid. The training data was divided into two groups, 80% used for training and 20% for validation. The RLS combiner has an initial weight for each hour for both the BLF and CLF outputs; after each iteration, it automatically updates the weights for each hour based on the algorithm’s calculation.

A picture containing text, clock

Description automatically generated

Figure – The structure of the BLF and CLF network

### 3.2.3 The LSTM Forecaster

Other researchers on the smart-grid team at UNB have used the LSTM algorithm for load forecasting, but only with the Saint John dataset. We took the present implementation and altered it to meet our datasets and input feature sets. Additionally, because the ANNSTLF structure was recognized as the best forecaster for short-term load forecasting [1], [93]; our approach mimics the ANNSTLF structure by creating a Base Load Forecaster, Change in the Load Forecaster, and RLS combiner; while using the LSTM algorithm in place of the ANN. The architecture will have the same inputs and structure as the ANNSTLF, but the BLF and CLF algorithms will be trained using LSTMs. We were interested in seeing if this adjustment could improve forecasting performance.

// More details to be added after the implementation

### 3.2.4 The CNN Forecaster

We implemented the CNN algorithm similarly to the LSTM using the ANNSTLF structure. The architecture of the CNNs used in this study consists of six layers: an input layer, a convolutional layer, a rectified linear unit activation layer (relu), a max-pooling layer, a fully connected layer, and a regression output layer. The adam optimization training algorithm was used to train the CNNs. Additional information about this algorithm can be found here [166].

## 3.3 How our Results Were Analyzed

// To be filled

# 4 Results and Discussion

## 4.1 Performance Metrics

This study will compare all forecasters’ performance across all forecasters and subsets of the forecasts such as weekdays, weekends, mornings, or evenings. It will assist us in identifying instances where forecasters perform better or worse than expected. The performance will be evaluated according to accuracy in forecast values and accuracy in peak load localization. Table 1 delineates the main error measures used to quantify accuracy:

|  |  |
| --- | --- |
|  |  |
|  |  |

Table

Mean Absolute Error (MAE) is the simplest way to measure forecast error [18], but because it is an absolute measure, it does not provide a way to compare measurements across forecast scenarios of different scales. For this reason, Mean Absolute Percent Error (MAPE) is commonly used [1] since the interpretation of comparisons is straightforward. The MAPE value indicates the magnitude of the forecasted values’ error in percentage terms; it is also the most frequently used load forecasting metric [1]. However, as is the case with demand forecasting, MAPE returns undefined values when the actuals are zero. It produces extreme values when the actuals are close to zero and penalizes negative errors (when forecasts exceed actuals) more severely than positive errors. This is because the percentage error for too low forecasts cannot surpass 100%, while there is no maximum limit to overly high forecasts [1], [167]. Both the MAE and MAPE risk underestimating the effect of significant but infrequent errors. By focusing exclusively on the mean, we run the danger of being blindsided by a colossal error. To accommodate for severe, unusual errors, we incorporated the Root Mean Square Error (RMSE) [64]. With the RMSE, when we square the errors before computing the mean and then take the square root, we get an error size measure favouring significant but rare errors above the mean. However, RMSE is not scaled to the original error, so it is more difficult to interpret.

To fully capture bias and precision, Mean Biased Error (MBE) and standard deviation (SD) can also be used [168], [169]. MBE denotes the mean error of all forecasts across the entire forecast horizon; it quantifies the model’s overall bias and determines if the model produces over-or under-estimation (MBE > 0 or MBE < 0). Since a positive error on one pair can compensate for a negative error on another, a forecast model can attain a very low bias while remaining imprecise. However, evaluating our forecast precision solely based on the MBE value will be insufficient; but a significantly biased forecast already indicates that something is amiss with the model.

Finally, standard deviation indicates the spread of errors by quantifying how far apart individual errors are from the mean error.  The standard deviation is calculated using the total number of values in the time series; it is one of the most accurate measures of dispersion. The standard deviation is less impacted than other measurements by irregularities in the time series.  In comparison to other measures of dispersion, the standard deviation is more difficult to compute and interpret. The standard deviation is also strongly influenced by extreme values in the time series. However, while these metrics have their limits, they are simple instruments for assessing forecast accuracy.

//More Details to Be Added Later

# 5 Conclusion

## 5.1 Contributions

This research will assess the value added by deep learning algorithms (like CNN and LTSM) by comparing their performance to traditional forecasters. We aim to explore deep learning approaches to see if they are more adaptive to changes in extraneous factors like annual increases in power demand or temperature shifts. The goal is to develop forecasters that can adapt to complex data relationships without explicit user intervention. This is an important goal for our Smart-Grid Project industry collaborators, and we will provide an analysis tuned explicitly to their data set. Furthermore, because we have also included analysis on data that is publicly available, this work will be reproducible, making it a valuable comparison point for future research within and beyond our smart-grid team.

## 5.2 Future Work

// To be filled

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] 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.

[3] K. Amarasinghe, D. L. Marino, and M. Manic, “Deep neural networks for energy load forecasting,” 2017, doi: 10.1109/ISIE.2017.8001465.

[4] 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.

[5] 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.

[6] 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.

[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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] W. He, “Load Forecasting via Deep Neural Networks,” 2017, doi: 10.1016/j.procs.2017.11.374.

[14] 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.

[15] 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.

[16] 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.

[17] “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 (accessed Sep. 17, 2021).

[18] 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.

[19] 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.

[20] 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.

[21] 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 Transadions Power Syst.*, 1991.

[22] A. G. Bakirtzis, V. Petridis, S. J. Klartzis, M. C. Alexiadis, and A. H. Maissis, “A neural network short term load forecasting model for the greek power system,” *IEEE Trans. Power Syst.*, 1996, doi: 10.1109/59.496166.

[23] 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.

[24] G. Gross and F. D. Galiana, “SHORT-TERM LOAD FORECASTING.,” *Proc. IEEE*, 1987, doi: 10.1109/PROC.1987.13927.

[25] A. Muñoz, E. F. Sánchez-Úbeda, A. Cruz, and J. Marín, “Short-term Forecasting in Power Systems: A Guided Tour,” 2010.

[26] 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.

[27] D. Srinivasan and M. A. Lee, “Survey of hybrid fuzzy neural approaches to electric load forecasting,” 1995, doi: 10.1109/icsmc.1995.538416.

[28] 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.

[29] 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.

[30] T. Hong, “Short Term Electric Load Forecasting dissertation,” *3442639*, 2010.

[31] J. Foster, “Electric load forecasting with increased embedded renewable generation,” Queen’s University, 2020.

[32] T. Hong and M. Shahidehpour, “Load Forecasting Case Study,” *U.S. Dep. Energy*, 2015.

[33] E. Taylor, “Short-term Electrical Load Forecasting for an Institutional/Industrial Power System Using an Artificial Neural Network,” The University of Tennessee, Knoxville, 2013.

[34] M. JANICKI, “Methods of weather variables introduction into short-term electric load forecasting models - a review,” *PRZEGLĄD ELEKTROTECHNICZNY*, 2017, doi: 10.15199/48.2017.04.18.

[35] 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.

[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] E. L. Taylor, “Short-term Electrical Load Forecasting for an Institutional/ Industrial Power System Using an Artificial Neural Network,” University of Tennessee, 2013.

[44] 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.

[45] 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.

[46] K. Liu, “Comparison of very short-term load forecasting techniques,” *IEEE Trans. Power Syst.*, 1996, doi: 10.1109/59.496169.

[47] W. Charyloniuk and M. S. Chen, “Very short-term load forecasting using artificial neural networks,” *IEEE Trans. Power Syst.*, 2000, doi: 10.1109/59.852131.

[48] 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.

[49] P. Mandal, T. Senjyu, 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.

[50] 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.

[51] 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.

[52] S. Dwijayanti, “Short Term Load Forecasting Using a Neural Network Based Time Series Approach,” Oklahoma State University, 2013.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] S. Kumar, S. Mishra, and S. Gupta, “Short term load forecasting using ANN and multiple linear regression,” 2016, doi: 10.1109/CICT.2016.44.

[58] 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.

[59] P. Ji, D. Xiong, P. Wang, and J. Chen, “A study on exponential smoothing model for load forecasting,” 2012, doi: 10.1109/APPEEC.2012.6307555.

[60] J. F. Rendon-Sanchez and L. M. de Menezes, “Structural combination of seasonal exponential smoothing forecasts applied to load forecasting,” *Eur. J. Oper. Res.*, 2019, doi: 10.1016/j.ejor.2018.12.013.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] M. Hanmandlu and B. K. Chauhan, “Load forecasting using hybrid models,” *IEEE Trans. Power Syst.*, 2011, doi: 10.1109/TPWRS.2010.2048585.

[67] A. Yang, W. Li, and X. Yang, “Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines,” *Knowledge-Based Syst.*, 2019, doi: 10.1016/j.knosys.2018.08.027.

[68] V. Mayrink and H. S. Hippert, “A hybrid method using Exponential Smoothing and Gradient Boosting for electrical short-term load forecasting,” 2017, doi: 10.1109/LA-CCI.2016.7885697.

[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] M. Baccouche, F. Mamalet, and C. Wolf, “（RGB)Sequential deep learning for human action recognition,” *Int. Work. Hum. Behav. Underst.*, 2011.

[73] 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.

[74] M. Vos, C. Bender-Saebelkampf, and S. Albayrak, “Residential Short-Term Load Forecasting Using Convolutional Neural Networks,” 2018, doi: 10.1109/SmartGridComm.2018.8587494.

[75] 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.

[76] C. Gallicchio, A. Micheli, and L. Pedrelli, “Design of deep echo state networks,” *Neural Networks*, 2018, doi: 10.1016/j.neunet.2018.08.002.

[77] 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.

[78] B. Farsi, “On Short-Term Load Forecasting Using Machine Learning Techniques,” Concordia University, 2020.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

[87] M. Rana and I. Koprinska, “Forecasting electricity load with advanced wavelet neural networks,” *Neurocomputing*, 2016, doi: 10.1016/j.neucom.2015.12.004.

[88] 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.

[89] 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.

[90] M. Abuella and B. Chowdhury, “Solar power probabilistic forecasting by using multiple linear regression analysis,” 2015, doi: 10.1109/SECON.2015.7132869.

[91] 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.

[92] X. Sun, Z. Ouyang, and D. Yue, “Short-term load forecasting based on multivariate linear regression,” 2017, doi: 10.1109/EI2.2017.8245401.

[93] R. Weron, *Modeling and forecasting electricity loads and prices: A statistical approach*. wiley, 2006.

[94] N. Amral, C. S. Özveren, and D. King, “Short term load forecasting using multiple linear regression,” 2007, doi: 10.1109/UPEC.2007.4469121.

[95] T. Hong, “Short Term Electric Load Forecasting,” North Carolina State University, 2010.

[96] E. Stellwagen and L. Tashman, “ARIMA : The Models of Box and Jenkins,” *Foresight Int. J. Appl. Forecast.*, 2013.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] A. Shadkam, “Using SARIMAX to forecast electricity demand and consumption in university buildings,” The University of British Columbia, 2020.

[102] I. Fernández, C. E. Borges, and Y. K. Penya, “Efficient building load forecasting,” 2011, doi: 10.1109/ETFA.2011.6059103.

[103] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, 1943, doi: 10.1007/BF02478259.

[104] D. O. Hebb, “The first stage of perception: growth of the assembly,” *Organ. Behav.*, 1949, doi: 10.1016/0301-0082(84)90021-2.

[105] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychol. Rev.*, 1958, doi: 10.1037/h0042519.

[106] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, 1986, doi: 10.1038/323533a0.

[107] 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.

[108] 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.

[109] 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.

[110] A. Si. Walia, “Activation functions and it’s types-Which is better?,” *Towards Data Science*, 2017. .

[111] “Activation Function - AI Wiki,” 2019. https://docs.paperspace.com/machine-learning/wiki/activation-function (accessed Sep. 18, 2021).

[112] “Artificial Neural Network (ANN) with Practical Implementation | by Amir Ali | Wavy AI Research Foundation | Medium,” 2019. https://medium.com/machine-learning-researcher/artificial-neural-network-ann-4481fa33d85a (accessed Sep. 10, 2021).

[113] C. L. COCIANU and H. GRIGORYAN, “An Artificial Neural Network for Data Forecasting Purposes,” *Inform. Econ.*, 2015, doi: 10.12948/issn14531305/19.2.2015.04.

[114] M. Adya and F. Collopy, “How effective are neural networks at forecasting and prediction? A review and evaluation,” *J. Forecast.*, 1998, doi: 10.1002/(sici)1099-131x(1998090)17:5/6<481::aid-for709>3.0.co;2-q.

[115] Zhang, G., E. Patuwo, and M. Y. Hu, “Forecasting with Artificial neural networds,” *Int. J. Forecast.*, 1998.

[116] 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.

[117] 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.

[118] B. F. Hobbs, “Analysis of the value for unit commitment of improved load forecasts,” *IEEE Trans. Power Syst.*, 1999, doi: 10.1109/59.801894.

[119] 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.

[120] 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.

[121] 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.

[122] P. R. J. Campbell and K. Adamson, “Methodologies for load forecasting,” 2006, doi: 10.1109/IS.2006.348523.

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

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

[125] 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.

[126] G. H. Yann LeCun, Yoshua Bengio, “Deep learning (2015), Y. LeCun, Y. Bengio and G. Hinton,” *Nature*, 2015.

[127] 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.

[128] S. Suresh, “An Analysis of Short-term Load Forecasting on Residential Buildings Using Deep Learning Models,” Virginia Polytechnic Institute and State University, Blacksburg, 2020.

[129] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” 2007, doi: 10.7551/mitpress/7503.003.0024.

[130] “What is the difference between Machine Learning and Deep Learning | by Neeraj Kumar | Medium,” 2017. https://medium.com/@Say2neeraj/what-is-the-difference-between-machine-learning-and-deep-learning-5795e4415be9 (accessed Sep. 18, 2021).

[131] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2015.

[132] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” 2013, doi: 10.1109/ICASSP.2013.6638947.

[133] 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.

[134] D. Silver, J. Schrittwieser, K. Simonyan, I. A.- Nature, and U. 2017, “Mastering the game of Go without human knowledge,” *Nature*. 2016.

[135] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, 2015, doi: 10.1038/nature14236.

[136] 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.

[137] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” 2012.

[138] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016, doi: 10.1109/CVPR.2016.90.

[139] 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.

[140] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” 2012.

[141] 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.

[142] G. E. Dahl, M. Ranzato, A. R. Mohamed, and G. Hinton, “Phone recognition with the mean-covariance restricted Boltzmann machine,” 2010.

[143] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using Context-Dependent Deep Neural Networks,” 2011, doi: 10.21437/interspeech.2011-169.

[144] 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.

[145] L. Deng and J. C. Platt, “Ensemble deep learning for speech recognition,” 2014, doi: 10.21437/interspeech.2014-433.

[146] B. Y. Goodfellow I., “Courville A-Deep learning-MIT (2016),” *Nature*, 2016.

[147] “Long Short Term Memory | Architecture Of LSTM,” 2017. https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/ (accessed Aug. 30, 2021).

[148] P. P. Phyo, “Deep Learning for Short-term Electricity Load Forecasting,” Sirindhorn International Institute of Technology, 2018.

[149] C. Olah, “Understanding LSTM Networks [Blog],” *Web Page*, 2015.

[150] 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.

[151] 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.

[152] I. Koprinska, D. Wu, and Z. Wang, “Convolutional Neural Networks for Energy Time Series Forecasting,” 2018, doi: 10.1109/IJCNN.2018.8489399.

[153] 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.

[154] 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.

[155] 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.

[156] M. Imani and H. Ghassemian, “Sequence to Image Transform Based Convolutional Neural Network for Load Forecasting,” 2019, doi: 10.1109/IranianCEE.2019.8786456.

[157] 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.

[158] 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.

[159] 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.

[160] E. Gawehn, J. A. Hiss, and G. Schneider, “Deep Learning in Drug Discovery,” *Molecular Informatics*. 2016, doi: 10.1002/minf.201501008.

[161] 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.

[162] “Convolutional neural networks for time series forecasting | Python for Finance Cookbook,” 2020. https://subscription.packtpub.com/book/data/9781789618518/10/ch10lvl1sec63/convolutional-neural-networks-for-time-series-forecasting (accessed Aug. 30, 2021).

[163] “ReLU : Not a Differentiable Function: Why used in Gradient Based Optimization? and Other Generalizations of ReLU. | by Kanchan Sarkar | Medium,” 2018. https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization-7fef3a4cecec (accessed Sep. 17, 2021).

[164] “What is max pooling in convolutional neural networks? - Quora,” 2017. https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks (accessed Sep. 17, 2021).

[165] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox TM 7 User ’ s Guide*. 2010.

[166] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, “Adam Optimization Algorithm for Wide and Deep Neural Network,” *Knowl. Eng. Data Sci.*, 2019, doi: 10.17977/um018v2i12019p41-46.

[167] 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.

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

[169] 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.

Appendix Title

Text begins here. To add additional chapters, simply start a new page and format title as the ‘Appendix’ style. Remove this text.

Glossary

Start writing here; remove page if there is no content

**Curriculum Vitae**

Candidate’s full name: Tolulope Oluwaseun Olugbenga

Universities attended:

BSc in Computer Science Engineering, University of Debrecen, 2018

Publications: None

Conference Presentations: None