Kurdistan Region Government/Iraq
Presidency of the Council of Ministers
Ministry of Higher Education & Scientific Research
Erbil Polytechnic University
Technical Engineering College
Information System Engineering Department

Load Forecasting to Erbil Governorate Based on Machine Learning Techniques

A Thesis
Submitted to the Council of the College of Technical Engineering at Erbil Polytechnic University, in Partial Fulfillment of the Requirements for the Degree of Master of Information System Engineering

by
Zhala Jameel Hamad
B.Sc. Software Engineering- Koya University- 2011

Supervised by
Dr. Ismael Abdulrahman

Erbil- Kurdistan Region
October 2022

بسم الله الرحمن الرحيم

(أَنزَلَ اللَّهُ عَلَيْكَ الْكِتَابَ وَالْحِكْمَةَ وَعَلَّمَكَ مَا لَمْ تَكُن تَعْلَمُ
وَكَانَ فَضْلُ اللَّهِ عَلَيْكَ عَظِيمًا)

سورة النساء - آية 113

Declaration

I declare that the Master Thesis entitled: “Load Forecasting to Erbil Governorate Based on Machine Learning Techniques”, represents my work which was done after registration for the degree of master. All works contained within this thesis are my independent research and have not been submitted for awarding any previous application for a degree or professional qualifications.

Signature:
Name: **Zhala Jameel Hamad**
Date: /   /2022

Linguistic Review

I, hereby, certify that this thesis, titled “Load Forecasting to Erbil Governorate Based on Machine Learning Techniques” has been read and checked. After indicating all the grammatical and spelling mistakes, the thesis was given again to the candidate to make adequate corrections. After the second reading, I found that the candidate corrected the indicated mistakes. Therefore, I certify that this thesis is free from mistakes.

Signature:

Name: **Dr. Himdad A. Muhammad**

Date: 30 /August / 2022

Supervisor Certificate

This thesis has been written under my supervision and has been submitted for the award of the degree of master of Information System Engineering with my approval as supervisor.

Signature:

Name: **Dr. Ismael Abdulrahman**

Date: / / 2022

I confirm that all requirements have been fulfilled.

Signature:

Name: **Dr. Roojwan Sc. Hawezi**

Head of the Department of Information System Engineering

Date: / / 2022

I confirm that all requirements have been fulfilled.

Postgraduate Office

Signature:

Name:

Date: / / 2022

Examining Committee Certification

We certify that we have read this thesis: “Load Forecasting to Erbil Governorate Based on Machine Learning Techniques”, and as an examining committee, examined the student (**Zhala Jameel Hamad**) in its content and what is related to it. We approve that it meets the standards of a thesis for the degree of master in Information System Engineering.

Signature
Name: **Asst. Prof. Dr. Amen Salih**
Chairman
Date: / / 2022

Signature
Name: **Asst. Prof. Dr. Abas Mohammed**
Member
Date: / / 2022

Signature
Name: **Asst. Prof. Dr. Bzar Khidir**
Member
Date: / / 2022

Signature
Name: **Dr. Esmael Abdulrahman**
Supervisor and Member
Date: / / 2022

Approved by:
Dean of the College of Erbil Technical Engineering
Signature
Name:
Date: / / 2022

Dedication

I dedicate this work to:

- My dear parents, especially my father
- My husband and my children

Acknowledgments

First of all, I would like to start by thanking Almighty Allah for giving me the opportunity, determination, and strength to do my research.

I am incredibly grateful to Dr. Ismael Abdulrahman, my supervisor, who guided me to write this thesis from the beginning until the final step. Words cannot express my gratitude to him. For such assistance, I always owe a special debt.

I'm thankful to Eng. Ramazan Mohamad (Head of Erbil Control Center) and Eng. Mohammed Faraj (Head of the Operation Department in Kurdistan Regional Control Center) and Eng. Ayad Hejran (Advisor of the Smart Meter Project) for helping me to obtain the data required for my study.

My appreciation and gratitude go to the head of the department and all academic staff members who taught me during my master's courses at (Department of Information System Engineering, Erbil Technical Engineering College, Erbil Polytechnic University) for their kindness.

Finally, I want to thank my supportive husband (Dlshad Ahmad) for his patience, encouragement, and assistance during my study.

ABSTRACT

Load forecasting is a nonlinear problem and complex task that plays a crucial role in power system planning, operation, and control. The complexity, ambiguity, and wide range of factors influencing the prediction make the load forecasting problem difficult. It is considered a type of time series problem that needs a special solution. A recent study proposed a deep learning approach called historical data augmentation (HDA) to enhance the accuracy of the load forecasting model by dividing the input data into several yearly sub-datasets. When the original data is associated with high time step changes from one year to another, the approach was not found as effective as it should be for a long-term forecasting. Because the time-series information is disconnected by the approach between the end of one-year sub-data and the beginning of the next-year sub-data. Alternatively, this study proposes using a two-year sub-dataset to connect the two ends of the yearly sub-sets. A correlation analysis is conducted to show how the yearly datasets are correlated to each other.

Several inputs are considered in the model to increase the model generalization, including load demand profile, weather information, and some important categorical data such as weekday and weekend data that are embedded using the one-hot encoding technique. In addition, a Simulink-based program is introduced to simulate the problem which has the advantage of visualizing the algorithm. The deep learning methods used in this study are the long short-term memory (LSTM) and gated recurrent unit (GRU) neural networks which have been increasingly employed in recent years for time series and sequence problems. The proposed model is applied to the Kurdistan regional load demands and compared with classical methods of data inputting, demonstrating improvements in both the model accuracy and training time. Also, it showed that an OTSAF forecasting structure works better in terms of accuracy than an MTSAF forecasting structure.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Full Name
AI	Artificial Intelligent
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
GRU	Gated Recurrent Unit
HAD	Historical Data Augmentation
LSTM	Long Short-Term Memory
LTLF	Long-Term Load Forecasting
ML	Machine Learning
MTLF	Medium-Term Load Forecasting
MTSAF	Multi-Time Step Ahead Forecasting
MW	Megawatt
OTSAF	One Time Step Ahead Forecasting
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
STLF	Short-Term Load Forecasting

CHAPTER ONE: INTRODUCTION

1.1 Overview

Electricity is an essential element utilized in daily life and one of the most popular economic factors. A continuous supply of electricity to the load side is necessary for modern power systems. For this purpose, an accurate method of estimating load demand in the present and the future with the least amount of error is needed. To achieve this goal, scientists and academics have been working to employ the most effective and efficient technique known as load forecasting for predicting future electricity demand (Al Mamun et al., 2020). Load forecasting predicts future load demands by analyzing historical data and finding dependency patterns of its time-step observations (Kwon et al., 2020). Historical data is the most significant element in a forecasting model. The model should first comprehend the pattern of electrical load data consumption in order to be trained. It has many applications in power system operation and planning including demand response, scheduling, unit commitment, energy trading, system planning, and energy policy (Jacob et al., 2020).

Accurate load forecasting helps power companies and decision-makers to achieve a balance between supply and demand, prevent power interruptions due to load shedding, and avoid excess reserve of power generation. Demand forecasting reduces power generation costs and aids in creating a well-organized power system utility, which is vital due to the high cost of power generation (Kim et al., 2019).

The lack of a consistent electrical supply is one of the main obstacles to Iraq's economic development. Although grid-based power capacity has grown significantly over the past few years, it is still far from enough to fulfill the growing demand (Mohammed, 2018). This issue has an influence on

Kurdistan as well, as ongoing growth has led to more demand. There are three provinces in the Kurdistan Region: Erbil, Dohuk, and Sulaimani shown in Fig.1.1. Erbil is a populated area with a number of businesses, industries, and government offices. However, as a result of modern society and technological advancements, electrical loads change from day to day. Power system overload and shortage are results of rising demand from industrial, residential, and commercial sectors. Figure 1.2, shows the difference between annual peak demand and average supply from Kurdistan Region. It is obvious that the demand has increased in recent years (Taherifard, 2019).

Various Machine Learning (ML) based approaches are frequently utilized by various power and energy utility companies to forecast the amount of energy required to achieve stability between generation and demand. In the present study, LSTM and GRU approaches based on DL are proposed. The Erbil load dataset is used to evaluate each suggested model.

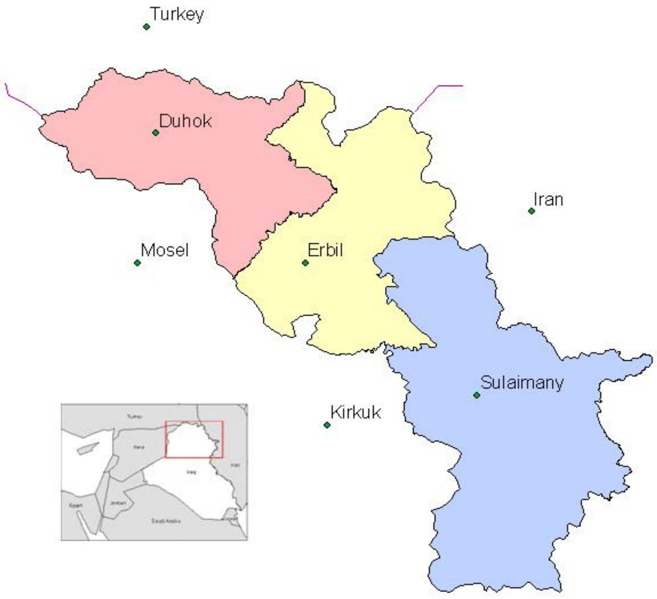
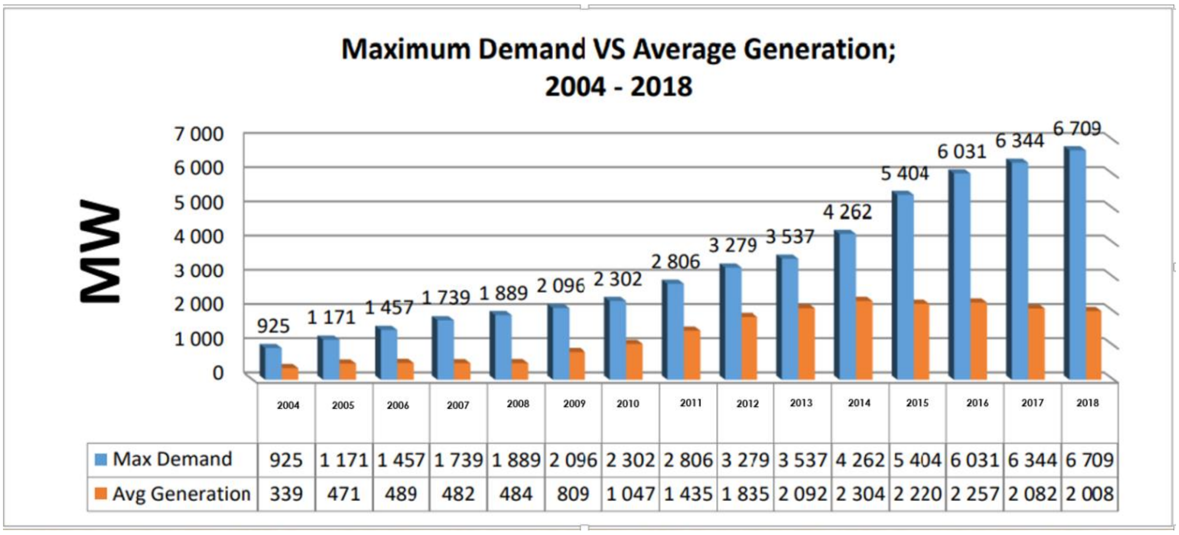


Figure 1.1 Map of Kurdistan Region (Hamad and Abdulrahman, 2022).



	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
■ Max Demand	925	1 171	1 457	1 739	1 889	2 096	2 302	2 806	3 279	3 537	4 262	5 404	6 031	6 344	6 709
■ Avg Generation	339	471	489	482	484	809	1 047	1 435	1 835	2 092	2 304	2 220	2 257	2 082	2 008

Figure 1.2 Difference between demand and generation in Kurdistan Region from 2004 to 2018 (Taherifard, 2019).

1.2 Problem Statement

Because there is no way of storing and creating electrical energy at the same time, the amount of electricity generated should be balanced with the amount used by users. Electricity providers are required to produce power that is balanced among consumer services, distribution, transmission, and generation. Load forecasting can assist in resolving these issues and reducing additional generating and end-user costs. Reliable forecast findings for electrical load prediction models are essential for the utility's generation and transmission plans as well as other economic factors.

The load forecasting studies conducted in the Kurdistan region that were accessible, revealed that many of them used statistical techniques, while others used simple ML approaches. Statistical techniques have limited accuracy and uncertainty when dealing with highly nonlinear systems. Contrarily, ML approaches such as artificial neural networks (ANN), DL, and RNN are more accurate and perform better.

Since load forecasting is a time-series issue, sequence-based and time-series issues can be solved with RNN. It is a feed-forward multi-neural network with additional feedback cycles from previous time steps used to store temporal information as internal states. A recurrent network adds a memory state to learn the sequence order of input data and extracts the dependencies among the input observations. However, almost all RNNs are nowadays replaced with LSTM or GRU to solve major shortcomings in the RNNs: vanishing and exploding gradients. When the RNN weights are updated, it quickly results in either too small changes in the weights (vanishing gradient) or too large changes (exploding). The result is a short-term memory which is extremely hard for the RNN to learn and determine the dependencies among observations from earlier time steps to the later ones**.** Consequently, replacing RNN with LSTM and GRU allows for handling longer data sequences.

To improve the accuracy of the load forecasting model, we divide the input data into multiple-year sub-datasets and employ a DL method termed historical data augmentation (HDA). This method was not found to be as effective as it should be for long-term forecasting of Erbil data (explained in section 4.2). Since it disconnects the time-series information at the end of one year and the beginning of the next. To connect the two ends of the yearly sub-sets, this study suggests using a two-year sub-dataset.

1.3 Research Motivation

There is currently a critical shortage of electricity production and reserves in the Kurdistan region. Population growth as well as the expansion of the industrial and commercial sectors need a continuously high level of power supply. Governments and power providers are under pressure to find a way to solve these issues. Load forecasting is used to forecast future power usage over a certain period. If the model predicts a load pattern close to the actual, responsible entities can make cost-effective decisions depending on the expected values. Furthermore, a predictive model allows power companies to manage scheduling, such as maintenance activities, as well as enhance energy efficiency. Generally, load forecasting lowers costs, improves the reliability of the system, and maximizes the use of available energy resources.

Load forecasting is a time-series problem. Modeling time series with an RNN is a powerful technique. It utilizes an internal state to recall data over time and is useful for load forecasting. While learning from long data sequences is difficult. When the gradient gets lower and smaller, the weight updates become irrelevant, which implies no meaningful learning is taking place. As alternatives, LSTM and GRU can move information over length sequences and solve this issue. In addition, the historical data which appears to be sequential was collected every day for six years. LSTM and GRU are employed to estimate the daily load, weekly load, and 365-day load. As a result, this study emphasizes DL methods for predicting load for the province of Erbil.

1.4 Research Objectives

The main objective of this research is to propose a load forecasting model for the Erbil province. This study concentrated on the following objectives:

- To apply different DL techniques in the context of regression forecasting based on historical data of average daily load demand.
- To conduct load forecasting with factors including temperature, weekdays, and weekends. Considering factors with load data, to enhance the accuracy.
- To propose a reform in the forecasting input data to obtain a better performance and solve certain complex problems that the one-year data-augmentation approach fails to predict accurately.
- To introduce the Simulink model of prediction in order to fill the gap in the current programs used for load forecasting.

1.5 Outline of Thesis

The thesis is divided into five chapters, each of which is described as follows:

Chapter 1 Introduction of load forecasting model for Erbil province utilizing deep learning algorithms based on historical data of daily average load. It also shows research problems, as well as the thesis objectives and outline.

Chapter 2 Covers the related work, which provides literature discussing load forecasting methods. In addition, the definition of load forecasting and the key factors that influence load forecasting are explained. Statistical and machine learning load forecasting techniques are also discussed.

Chapter 3 Describes the methodology of the thesis. The architecture of load forecasting techniques employed in this study and the functionalities of each technique are explained.

Chapter 4 presents the results of applying the proposed approach to the Kurdistan region. The scatter plots in this chapter show how the dataset for one year is correlated to another yearly dataset of the same time-series sequence. The simulations developed as a result of the research are also included. The results of the weekly load forecast using smoothed data are displayed at the end of the chapter.

Chapter 5 Summarizes the thesis conclusions and suggestions for future work.

CHAPTER TWO: BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

This chapter gives a background theory of load forecasting with literature related to the thesis's work and discusses various forecasting approaches. Differences and similarities among studies have been highlighted. Over the years, there has been a rise in interest in load forecasting research, with multiple studies on load forecasting systems. Techniques of load forecasting are also addressed. While there are two types of load forecasting techniques: statistical and machine learning, only studies relating to ML, particularly DL, will be covered in this chapter. When training a neural network, some factors must be taken into account; these factors were also mentioned.

2.2 Load Forecasting

A forecast is the prediction of an upcoming event or series of events, and producing accurate forecasts often isn't easy. It is a complex issue that affects a wide range of sectors, including industry and business, government, finance, environmental sciences, medical, sociology, politics, and economics. Forecasting is important because it serves as a source of input in many management processes (Soliman and Al-Kandari, 2010). The electricity sector is a vital aspect of society that has a big impact on people's lives. To avoid wasting energy resources, power energy should not be supplied over demand. Furthermore, there shouldn't be a shortage of it because that could lead to outages in some locations. For keeping the balance between supply and demand for electricity, load forecasting is necessary (He, 2017).

Knowing future load requirements assists utility businesses in planning, making financially viable decisions, and reducing risk. Forecasting is also used to make decisions about future generating and transmission investments. It aids in preparing resources like fuels necessary to operate the generation side and other resources necessary to ensure that consumers have uninterrupted power. This demonstrates cost-effective electricity generation and distribution. Load forecasting assists in planning a future generation plant's size, location, capacity, and type. Also, it offers an overview of the costs of transmission and distribution infrastructure (Al Mamun et al., 2020).

Time-series data is used in the majority of forecasting situations. A time series is a collection of values collected over a period of time and organized sequentially. Moreover, load forecasting can be seen as a time series problem (Martínez-Álvarez et al., 2015). A sequence of random variables, $y_1, y_2, y_3, \ldots, y_n$ can be defined as an observed time series. Where y_1 signifies the series value at the first time step, $y_2$indicates the series value at the second time step, and so on until period n (Montgomery et al., 2015). It indicates that time series is made up of n sequential data derived from a random variable. Predicting a random variable in a time series entails obtaining knowledge about the random variable to be forecasted by looking at previous random variables. To forecast y_{1+i} at moment n, we used the observational data $y_1, y_2, \ldots, y_{n-1}$ for $i > 0$. This refers to the function $y_{n(i)}$ that gives us useful information about y_{n+i} in terms of previous data.

Various methods for predicting future values based on previous observations have been developed. When data is available and a pattern in the data is predicted to persist into the future, these strategies are appropriate to utilize.

The input and output of the system can be used to classify time series forecasting. Univariate and multivariate models can be differentiated based on input. Multi-step/single-step models can be distinguished based on the output. The multivariate model combines extra time-series variables such as weather or calendar data to achieve the forecasting objective, whereas univariate models produce forecasts purely based on univariate data like previous load data. Models that anticipate forwards in time only for one time step are referred to as single-step forecasting, while multi-step forecasting makes forecasts up to a specific timescale (Rana and Rahman, 2020).

2.3 Load Forecasting Types

Depending on its application, load forecasting can be classified into very–short-term load forecasting (VSTLF), short-term load forecasting (STLF), medium-term load forecasting (MTLF), and long-term load forecasting (LTLF), as shown in Fig 2.1. VSTLF is used in the problems of demand response and real-time operation that requires a time horizon of a few minutes to several hours ahead. It is rarely mentioned in studies because it is very short. Forecasting the load demand from one day to several days ahead called STLF; is essential for a utility's daily operations, such as unit commitment and load control. One week forecasting to several weeks ahead is known as MTLF. These two types of forecasting cover the majority of load-forecasting studies in the literature and are mainly used in scheduling, unit commitment, and energy marketing. Lastly, LTLF refers to the forecasting with a time frame of up to several years ahead and it is useful for planning and energy-trading purposes (Farsi et al., 2021, Eskandari et al., 2021).

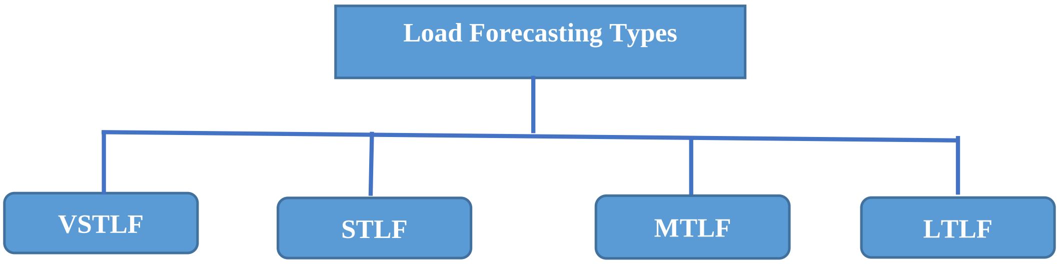


Figure 2.1 Load forecasting types

2.4 Load Forecasting Techniques

The use of different forecasting techniques for load forecasting by different researchers is an important point to note from all of those studies. Although all of the strategies utilized in those studies can reliably forecast load, some forecasting techniques outperform others in various conditions. Forecasting models employ a variety of methodologies that can be classified as statistical, artificial intelligence (AI), or hybrid. Statistical methods necessitate the development of a mathematical model that depicts the connection between the end load and other input variables. These were the earliest techniques utilized, and they are still relevant today. Statistical techniques are relatively fast, easy to set up, and computationally inexpensive. However, they suffer from uncertainty and low accuracy with high nonlinear systems. Time series analysis, exponential smoothing, and regression approaches are commonly used (López et al., 2018). These models have the advantage of producing accurate findings under normal circumstances. But they have limited accuracy or cannot produce satisfactory results when dealing with nonlinearly related variables (Talaat et al., 2020). Autoregressive integrated moving average (ARIMA) is the most natural technique for forecasting load among the traditional time series models. The unpredictable nature of the load time series can be expressed well by ARIMA processes. It does not have any issues with modeling several seasonal cycles or adding exogenous variables. ARIMA model drawback is that they can only represent linear correlations between variables (Dudek, 2016). The intricacy of certain nonlinear data patterns, as well as the excessive amount of computational options leading to long solution durations, are among the explanations.

Therefore, artificial neural networks (ANN) and intelligent ML techniques offer a promising and appealing alternative to this type of challenge. The relationship between the input and output variables, which can be difficult and complex to derive from the mathematical formulation, is the basis for the ANN's model strength. Increased processing capacity has made forecasting easier in various power system management applications, from load forecasting to security assessment and problem diagnostics. As a result, AI approaches have proven to be effective in reducing prediction errors (Kuster et al., 2017).

In AI-based methods for nonlinear time series issues, neural network techniques are the most prevalent. As a branch of ML, DL evolved from ANN.

Deep learning, which primarily refers to the multi-layer network with strong feature learning capabilities, has recently drawn a lot of attention for electrical load-interval forecasting. It has three key characteristics: big-data training, excellent generalization ability, and unsupervised feature learning (Dong et al., 2021). Because most DL algorithms imply a neural network topology, they are sometimes referred to as DNN. However, because of the back propagation (BP) approach, neural networks do not operate effectively when there are several hidden layers. It takes a long time, and because of the random initialization, it occasionally produces a bad local minimum and slow convergence. Deep architectures are the best solution for solving difficult ML challenges (Wei et al., 2019). Therefore, DL models are employed.

2.5 Factors that Impact the Accuracy of Load Forecasting

A comprehensive understanding of the system's properties is necessary in order to develop an effective and efficient forecasting model. Many factors affect consumer load behavior as well as overall losses in the transmission systems. These factors include time considerations, weather, the economy, population, epidemic, and war. Some factors have a long-term impact, while others have a short-term impact. The most essential factors will be discussed below.

One of the most significant independent factors in load forecasting is the weather. The weather impacts residential and agricultural customers, it can also change the load profile of industrial users. Weather forecasts and other elements are used in load forecasting models to estimate future loads and save operational costs. Weather is frequently considered a tipping point that can create system unreliability by reducing the efficient power supply. Weather data can be temperature, wind speed, rainfall, humidity, etc. (Liu et al., 2016). Among them, temperature is a critical meteorological variable that has a considerable impact on load demand. It has an impact on the generators' unit commitment status. Three input variables are represented by the temperature data points. When the temperature is included in the inputs, the minimum and maximum values are also gathered in addition to the current load demand inputs (Reddy, 2018).

Time is also a significant factor. The hour of the day, the day of the week (weekday, weekend), and the month of the year all influence the load consumption. Peak loads occur in the morning and evening during the winter. While during summer, increasing load intensity in the afternoon hours correlates to the use of air conditioning. Therefore, the load curve is periodic. Holidays and special days have a different load pattern than regular days (Lusis et al., 2017). Since electricity has become a need in people's daily lives, it has become a commodity. As a result, the state economy has an impact on electricity demand. Economic factors play a larger role in long-term forecasting but can also affect the total load in short-term forecasting. In comparison to the daily load curve of underdeveloped countries, the daily load curve of developed countries reveals various patterns (Zhao et al., 2021). The educational level, annual income, price, and other economic factors influence the building load. Low energy costs and a large annual income will further encourage the resident to use the device frequently (Ma et al., 2017).

2.6 Literatures Review Based on Deep Learning

Recently, deep learning has been receiving special attention because of its capacity to capture data behavior when dealing with complicated non-linear patterns and massive amounts of data (Bouktif et al., 2018). Numerous studies have been conducted on DL approaches, which are at the top of innovation in load forecasting systems. Their effectiveness as reliable predictions has been proven by researchers, as long as a dataset of acceptable quality and quantity is provided and the appropriate parameters are determined (Jin et al., 2021). Among DL approaches, the LSTM and GRU are quite well to time series data and have good adaptability (Tang et al., 2019). They outperform other ML algorithms in many ways, especially in terms of storing sequential data for long-term forecasting. (Rajagukguk et al., 2020).

The literature review was divided into three time periods: Firstly: Short-term forecasting can help with load flow estimation and decision-making to prevent overloading (Khan et al., 2013). Most recent studies use LSTM as the main DNN or a hybrid model. They suggested multi-features, and characteristics for extracting relevant data from previous data to construct a better STLF or VSTLF network. Secondly, medium-term forecasting, it is necessary to develop a perfect schedule for generating plants in order to improve the effectiveness of fuel supply in power plants (Askari and Keynia, 2020). While, long-term load forecasting refers to forecasting with a time frame of up to several years ahead and it is helpful for planning and energy-trading purposes (Xie et al., 2015). There are many variables to consider when forecasting over a long period of time, which adds complexity and reduces forecast confidence. As a result, there is little research on this time period (Khuntia et al., 2016).

For predicting non-residential consumer load in (Jiao et al., 2018), LSTM is suggested. A sizable amount of energy consumption is composed of non-residential consumers, such as commercial and industrial users. Using multiple correlated sequence information, the k-means algorithm assesses non-residential consumer daily load curves, categorizes, and extracts their energy consumption patterns.

The RNN with LSTM cells was proposed by (Agrawal et al., 2018) as a core of the model. The model Forecasts power usage over five years at the hourly resolution. As well, a new deep supervised learning model based on LSTM was introduced by (Tan et al., 2019) to evaluate ultra-short-term industrial power consumption. Depending on the bias-variance tradeoff, they developed a novel loss function that includes peak demand forecasting inaccuracy. This loss function aids model learning through mixing two types of error average error across all samples and maximum error across various sample distribution and making a tradeoff between both errors. Potentially ensuring that the model performs well in each situation.

In (Bouktif et al., 2019), LSTM and GRU models with single and multi-sequences have been proposed to forecast daily, weekly, monthly, and yearly load. They showed that by feeding both models numerous temporal sequences as inputs, they were able to learn critical information reliably over extended timescales. In addition, it decreased forecast error by over 15%.

Also, in (Dong and Grumbach, 2019) the hybrid of LSTM and GRU is proposed on distribution feeders. Compared to traditional models, the proposed method with the inclusion of a virtual feeder displayed higher performance for both summer and winter forecasts. Many aspects have been considered (max demand from last year, max commercial load percentage, max residential load percentage, max temperature, max temperature change, net load change for a large number of customers).

Authors (Tang et al., 2019, Sehovac and Grolinger, 2020) showed that when LSTM and GRU are introduced to the network, experimental results indicate that the model has an improved accuracy rate. Also (Wu et al., 2019) illustrated that GRU's computation speed is faster than LSTM, and its accuracy is higher. Further, taking into consideration past electricity costs improves accuracy. The multi-sequence approach suggested by (Bouktif et al., 2019, Lai et al., 2020) was found to be more resistant to time fluctuations than machine learning and single-sequence models output. However (Lai et al., 2020), used DNN and historical data augmentation (DNN–HDA). The approach divides the incoming data into numerous sequences, each representing a single year. When data is separated into several parts, information concerning the relationship between the end of one portion and the start of the following portion is lost. As illustrated in this research, this may not be an issue for some data and load forecasting situations. When the nature of the data changes and involves high uncertainty and volatility in the time step information. It struggles to anticipate future load demand, particularly for long-term forecasting.

Regarding forecast power usage in residential buildings, researchers (Sajjad et al., 2020) presented a hybrid CNN-GRU model. According to the representative features' extraction possibilities of CNNs and the efficient gated design of multi-layered GRU, the suggested model is a great replacement for earlier hybrid models in terms of performance and computational complexity efficiency. Day-ahead load forecasting is carried out for normal days (excluding special days) by (Kwon et al., 2020). The fully connected (FC) layer is utilized as input for the forecasting day, with an LSTM layer extracting features from previous data. While (Alhussein et al., 2020, ,nShao et al., 2020, Shang et al., 2021, Farsi et al., 2021, Rafi et al., 2021), used CNN layers to extract features from the input data, and LSTM layers to learn sequences. The proposed methodologies are executed with their hidden properties, which can obtain the benefits of both methods. To ensure the stability and effectiveness of the produced models, they are tested by examining the electrical load forecasting of various datasets. There are several beneficial aspects offered by CNN, and it can identify the necessary characteristics without human interaction. There is less dependency on pre-processing, which reduces the amount of human effort required to build its features. Furthermore, it is effective in both supervised and unsupervised learning situations, and simple to comprehend and execute. Most studies used a one-dimensional convolution layer (Conv1D) since they deal with sequence data. The hybrid CNN-LSTM for prediction shown in Fig 2.2 was proposed by (Rafi et al., 2021).

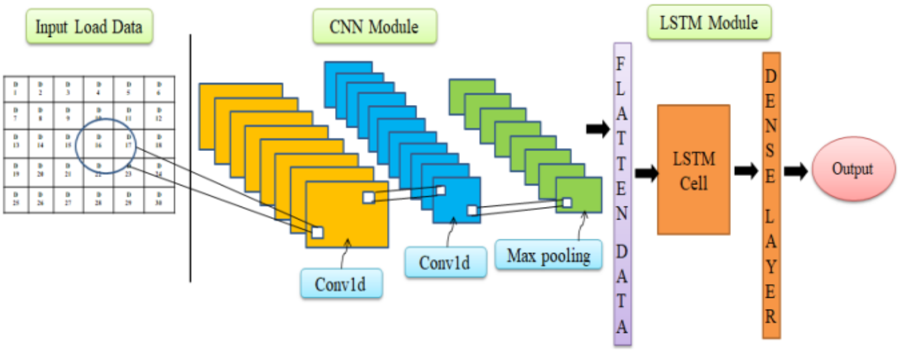


Figure 2.2 hybird CNN-LSTM model for prediction (Rafi et al., 2021).

In addition, CNN was proposed by (Tudose et al., 2021), which took into account pandemic impacts (COVID-19) and traditional exogenous parameters (weather, weekday, season, etc.). The pandemic limits are a crucial component of innovative load forecasting algorithms, as government choices and infection rates influence economic activity, modifying power demand at both aggregated and individual consumption levels. A simulated analysis of monthly electricity usage time series for 35 European countries was presented by (Dudek et al., 2021). Exponential Smoothing (ETS), and advanced LSTM are used in this model. ETS extracts the key components of each time series in real-time allowing the model to learn how to represent them. Multilayer LSTM has expanded recurrent skip connections and a spatial shortcut channel from lower layers to capture long-term seasonal patterns and enable more efficient training.

Also, a stacked LSTM network (SLSTMN) is used to suggest a model by (Farrag and Elattar, 2021). This model is typically built to anticipate the annual peak load or annual energy consumption as a single sequence. The proposed approach is intended to address this gap by predicting the daily load. When compared to other techniques on the same dataset as well as related work models on various datasets, SLSTMN achieves great accuracy and has the lowest percentage error (nearly 1%). Furthermore, via LSTM, the effect of weather factors on the prediction of a particular household's electricity usage was presented by (Wang et al., 2021). The analysis indicated that including weather data improved prediction accuracy, particularly in terms of temperature. The abstracts of the above studies are listed in table 2.1.

Table 2.1 Summaries of literature reviews.

Ref.	Year	Algorithm	Duration	No. of (Hidden _Layer)	Inputs	Error rate
(Jiao et al., 2018)	2018	LSTM+ K-means	Daily	LSTM(3)	Adjacent-time point correlation, day-related correlation, week-related correlation	%5.15
(Agrawal et al., 2018)	2018	LSTM	Five years	LSTM(3), Dense(1)	Dew point temperature, dry bulb temperature, load demand, day-ahead locational marginal pricing with its components	%6.54
(Tan et al., 2019)	2019	LSTM	From(1-5) minutes	LSTM(2), FC(1)	Demand load, hour, week-day, month	%4.17
(Bouktif et al., 2019)	2019	LSTM+ GRU	Daily, weekly, monthly, yearly	LSTM(1_4), GRU(1_4)	Load, temperature, humidity, wind-speed, week-day, weekend	%0.55
(Dong and Grumbach, 2019)	2019	LSTM/ GRU+ NN	One year	LSTM(1)/ GRU(1),NN(1)	Max demand-last year, max commercial load percentage, max residential load percentage, max temperature, max temperature change, net load change for a large number of customers	%6.67
(Tang et al., 2019)	2019	LSTM+ GRU	Hourly	LSTM(2), GRU(2)	Load, weather data	%1.90
(Wu et al., 2019)	2019	GRU	Hourly	GRU(3)	Load, hour of day, weekend, week-day, holiday, electricity price, dry bulb temperature.	%1.13
(Lai et	2020	DNN	Monthly	NN(4)	Different years' load	Different

al., 2020)					values on the same day, the adjacent loads, weather, calendar information	datasets Austria (2.79%) Czech (3.48%) Italy (2.97%)
(Sajjad et al., 2020)	2020	CNN+ GRU	Hourly	CNN(1),GRU(2)	Load, weather data	%4
(Kwon et al., 2020)	2020	LSTM	Day-ahead	LSTM(1),FC(1)	Load, weather data	%1.52
(Alhussein et al., 2020)	2020	LSTM+ CNN	Next 3-hour	LSTM(3),Conv1D(3),Pooling(3), FC(1)	Hour, holiday, week-day, electricity consumption data	4.01%, 4.76%, and 5.98% for one, two, and six next time steps.
(Shao et al., 2020)	2020	LSTM+ CNN	Next-hour	LSTM(2),Conv1D(3),Pooling(1), Flatten(1)	Power consumption data	%2.52
(Shang et al., 2021)	2021	LSTM+ CNN	Next-24 hour	LSTM(1),Conv1D(2),Pooling(1), Dense(2)	Temperature, humidity, wind-speed, weekday, weekend, holiday, electric-price	%1.34

(Farsi et al., 2021)	2021	LSTM+ CNN	Different time frames	LSTM(2),Conv1D(2),Pooling(1), Dense(3)
				Flatten(1)
					Load	German data (8.82)
						Malaysian data (1.77)

(Rafi et al., 2021)	2021	LSTM+ CNN	Different time frames	LSTM(1),Conv1D(2),pooling(1),Flatten(1), Dense(1)	Load	Weekly(4.84)
						Monthly(4.89)

(Tudose et al., 2021)	2021	CNN	Day-ahead	Conv1D(1_3)Dense(1)Flatten(1), Pooling(1_3)	Pandemic restriction, load, temperature, day of week, holiday, season	%3.59
(Dudek et al., 2021)	2021	LSTM	Monthly	LSTM(4)	Load	%1.61
(Farrag and Elattar, 2021)	2021	LSTM	Yearly	LSTM(3), Dense(3)	Daily-max load, daily- max temperature, daily-min temperature, holidays, weekday, and month	%1
(Wang et al., 2021)	2021	LSTM	Hour-ahead	2LSTM	Load, weather data	%7.43

2.7 Load Forecasting Studies for Kurdistan Region

Many techniques have been developed to forecast load in Kurdistan Region. In (Ali, 2020, Taherifard, 2019, Kareem and Majeed, 2006), the authors tried the statistical method which is the traditional method of time series. An attempt was made by (Ali, 2020) to forecast Erbil peak monthly demand. For a dataset of power consumers, a simple linear regression (LR) and ARIMA were utilized as forecasting models. They conclude that the ARIMA approach is significantly superior to employ for load forecasting than the regression analysis. For the following reasons, they recommend this strategy. First: despite a decrease in load needs in May and April, the ARIMA was able to forecast electrical load data. When the findings are compared to the actual electrical load needs, ARIMA not only anticipates electrical load but also forecasts future electrical load demands with a significantly lower inaccuracy. Second: ARIMA is thought to be more robust in forecasting electrical load demand due to its high accuracy and precision. Third: due to the obvious straightforward arithmetical calculations, the ARIMA approach gives findings significantly faster than regression analysis, whereas regression analysis requires certain math calculations before it can start forecasting electricity load.

ARIMA was also offered in (Taherifard, 2019), but this time for the province of Sulaimani. To forecast load and demand on a daily, weekly, and monthly basis, the model is tested on supply and demand datasets. The outcome demonstrated that the supply forecasts were more accurate than the demand forecasts. Another study (Kareem and Majeed, 2006), proposed a monthly peak-load demand based on the most extensively used traditional approach, seasonal autoregressive integrated moving average (SARIMA). The collected results reveal a reasonably accurate load estimation. Alternatively, other studies employed simple ML approaches such as ANNs, which have been successfully used in (Cankurt and Yasin, 2018, Muhammed, 2011, Rasool et al., 2009) studies, especially for STLF in Erbil. In (Cankurt and Yasin, 2018), the highest energy demand in Kurdistan is forecasted daily. To normalize the dataset, they employed a transformation technique that scales the input between the top and bottom boundaries of -1 and 1. It could help the network function better by decreasing the effects of noisy data and flattening the attribute distribution.

Two distinct ANN architectures for predicting the power for the upcoming hour are suggested by (Muhammed, 2011). Because the training and testing data are both fixed in length, the BP learning technique is chosen to train both network architectures. The findings show that the second network structure is more convenient and accurate than the first. This is because, instead of the two input characteristics (current hour, current hourly load) used in the first network construction, four input features (period, previous hour, current hour, and current hourly load) were chosen and fed to the network. In (Rasool et al., 2009), wavelet transform is used to improve the learning capability of ANN by merging them to forecast the coming seven days. The approach was evaluated with actual data based on 2006 dataset, and the error percent for the last week of August using average temperature was 2.99 percent, but it was 4.56 percent with maximum and minimum temperature. Given the inaccuracy of the available data, because they had a lack of weather information, results demonstrated an encouraging level of accuracy.

In (Melhum et al., 2013), three-layer feed-forward neural network (FFNN) with a BP for Duhok province was suggested. It considered the impact of the length of time spent disconnected on load. They applied and validated four models. The first and second models forecast values for one day ahead and incoming seven days, while the third and fourth models estimate values for the next and seven days, respectively, in terms of the amount of disconnected time. The findings indicated that neural networks are an effective and practical approach for STLF.

Another intelligence method that applied MTLF for Sulaimani was investigated by (Ali et al., 2020). An ANN was used to forecast monthly peak-load demand. Temperature and humidity are considered to be additional factors because the environment in Kurdistan is characterized by cold winters and warm summers. The demand is affected by the significant temperature change. Since electricity demand peaks every year in January, February, and December. As a result, they conclude that the city requires further investment in electricity energy load to meet consumer demand. The type of forecasting, forecasting period, and the input features used in each study for building level forecasting are discussed in table2.2.

Table 2.2 Summary of Kurdistan-related papers (grouped by area).

Ref.	Year	Algorithm	Duration	Forecast Type	Inputs	Province	
(Ali, 2020)	2020	ARIMA& LR	Monthly	MTLF	Load	Erbil	Statistical
(Taherifard , 2019)	2019	ARIMA	Daily, weekly, monthly	STLF+ MTLF	Load	Sulaimani	
(Kareem and Majeed, 2006)	2006	SARIMA	Monthly	MTLF	Load	Sulaimani	
(Cankurt and Yasin, 2018)	2018	ANN	From day (1-14)	STLF	Load	Erbil	Machine learning algorithm
(Muhammed, 2011)	2011	ANN	Next hour	STLF	Previous hour, Current hour, Current hour load, period	Erbil	
(Rasool et al., 2009)	2009	WT&ANN	Next week	STLF	Load, temperature	Erbil	
(Melhum et al., 2013)	2013	FFNN&BP	Day-ahead, week-ahead	STLF	Load	Duhok	
(Ali et al., 2020)	2020	ANN	Monthly	MTLF	Load, temperature, humidity, precipitation	Sulaimani	

CHAPTER THREE: METHODOLOGY

3.1 Introduction

This chapter discusses the applied deep learning methods. Deep learning forecasting techniques have been widely used in the load consumption system because of their capacity to accurately forecast and provide timely predictions. RNN and time-series models are subclasses of DL forecasting methods. For load forecasting, this study utilizes DL techniques, specifically LSTM and GRU models. To better understand of the procedure applied in the empirical study, the outcomes of the suggested model using those methods will be shown in the next chapter.

3.2 Deep Learning (DL)

Deep learning is a type of machine learning that encompasses a wide range of techniques. DL has been a developing study subject since 2006, examining performance in a variety of fields including, image segmentation, machine translation, speech recognition, and object detection. As the sample data for training grows, the algorithm's efficiency increases adaptively. It uses multiple nonlinear layer processing for supervised or unsupervised learning and attempts to benefit through hierarchical data descriptions (Solyali, 2020). In the science of ML, DL is a relatively recent concept.

The correlation between them is depicted in Fig. 3.1. Its goal is to create and simulate a human brain neural network for analysis and learning. It replicates the human brain's ability to process data such as text, sound, and images. In theory, a deep architecture is defined as a neural network with more than two layers (input and output). The approximation error can be decreased by introducing hidden layers between both the input and output layers. Deep architectures are useful for detecting and capturing higher-level representations and abstractions. However, it is not only about the number of layers; it is also about the concept of automating the construction of more complicated features at each stage. These complicated architectures aren't limited to hard tasks to learn, but they can also outperform humans' level performance in specific applications (Wani et al., 2020).

Many researchers argued that traditional ML approaches were limited in their ability to analyze raw data. These strategies are required for manually designing feature extractors that convert raw data via human engineering. On the other hand, DL models are representation or feature learning models that can automatically identify and classify many levels of features. Furthermore, it requires minimal manual engineering, allowing it to expand the amount of data and processing available. Deep neural networks are currently progressing in terms of new learning architectures and methods.

Deep learning makes use of the neuron, which is a basic computational unit that accepts many signals as input. It linearly integrates these signals with the weight before transferring the combined signals to the nonlinear tasks to generate outputs. Many different architectures and algorithms can be used to apply the concept of DL into practice such as (auto-encoder (AE), convolutional neural network (CNN), restricted boltzmann machine (RBM), deep stacking network (DSN), long short term memory (LSTM), gated recurrent unit (GRU) network, and recurrent neural network (RNN)) (Dargan et al., 2019).

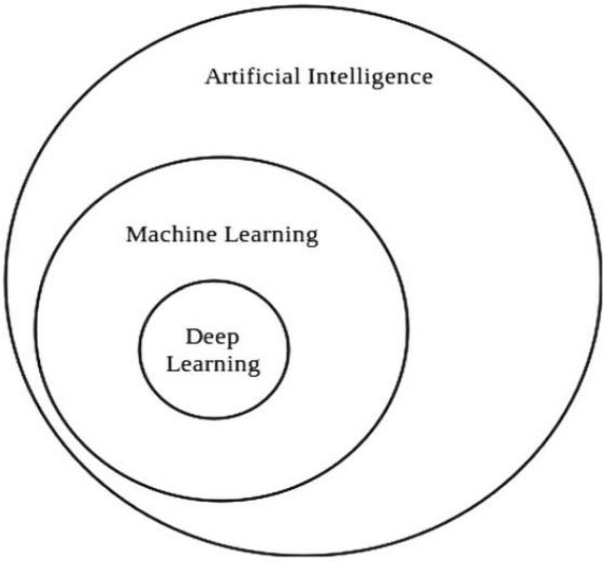


Figure 3.1 The link between ML and DL (Wani et al., 2020).

3.3 Recurrent Neural Network (RNN)

Traditional neural networks, such as the multilayer perceptron (MLP), can be used to solve sequence-based and time-series problems, but they have several drawbacks in practice. Some of these drawbacks include its stateless nature, fixed-sized inputs and outputs, messy scaling, and lack of knowledge of time-related structure (Brownlee, 2017). RNN is a superior alternative neural network for these kinds of situations and it is a feed-forward multi-neural network that includes an input layer, a hidden layer, and an output layer that uses extra feedback cycles from prior time steps that are used to hold temporal data as internal states. That is skilled at recognizing patterns in sequences of text, video, language, audio, and time-series data.

RNN is a powerful algorithm for classifying, clustering, and predicting data, especially time series and text. Because of its internal memory, it has been incredibly successful when applied to problems where the input data are in the form of a sequence for which predictions are to be made. It is designed to work with arbitrary inputs over long sequences, repeating the same task for each element inside the sequence and relying on past computations for output.

Instead of a single piece of data, the outcome is dependent on the sequence of data (Manaswi, 2018).

(Input1) → Output1

(Input2, Input1) → Output2

(Input3, Input2, Input1) → Output3

As shown, the output is determined by the current and prior inputs. Also, the nodes of the input layer are connected to the nodes of the hidden layer.

(Input1) → Hidden1

(Input2, Hidden1) → Hidden2

(Input3, Hidden1, Hidden2) → Hidden3

RNN has usually trained with the BP algorithm. Data flows forward when employing a feed-forward neural network that receives inputs (x) and generates output (y). The initial information is provided by the inputs (x), which are subsequently propagated up to the hidden units at every layer, eventually producing (y). This is referred to as "forward propagation" (Goodfellow et al., 2016). Transmitting information back is what the term "back propagation" signifies. This is precisely what the BP algorithm does: it returns the estimated loss to the system, where the optimizer adjusts the weights and biases.

Furthermore, it is a training process that is used to change weight in neural networks to reduce error. After obtaining network output, the predicted output is compared to the actual output, and a different error is calculated based on the difference. Weight is modified to reduce errors, and the resulting output will be closer to the desired output. This procedure is repeated indefinitely until overriding, with each iteration producing a more precise result than the last one. The training of the neural network refers to the complete process that occurs within the neural network layer (Géron, 2019). Unfortunately, due to its vanishing gradient problem, RNN is unable to train to connect knowledge with long-term dependence. Back propagation has difficulty training the sequence long-term (Chen et al., 2019). Figure 3.2 shows the RNN method additional feedback cycle structure.

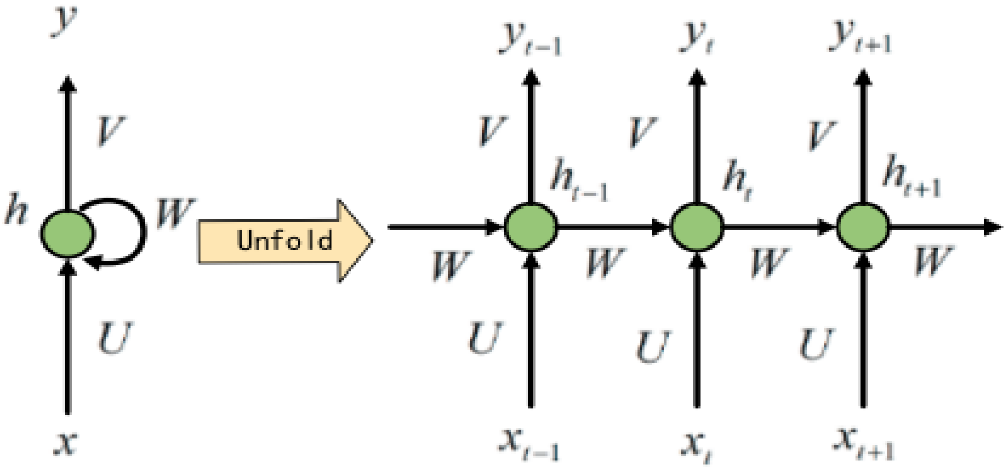


Figure 3.2 common structure of RNN (Chen et al., 2019).

As illustrated in fig 3.2, RNNs take a sequence of inputs of the neural network $\{x_0, x_1, \dots, x_t\}$, and previous hidden states h_{t-1}, to compute a sequence of outputs $\{y_0, y_{1,} \dots, y_t\}$, y_t is output at time step t. U, W, and V are the weights of the input to the hidden layer, the hidden layer to the hidden layer, and the hidden layer to the output respectively. Equation (1) is the mathematical representation of this unfolded process, where h_t =hidden state and σ =a non-linear activation function, such as a sigmoid function.

$$h_t = \sigma(W \times h_{t-1} + U \times x_t) \qquad 3.1$$

The output $y_t$is modified at every time step t, as follows:

$$y_t = \sigma(Vh_t) \qquad 3.2$$

3.4 Long Short-Term Memory (LSTM)

A more advanced version of the RNN architecture is the LSTM network. Hochreiter and Schmidhuber (1997) proposed the LSTM architecture (Hochreiter and Schmidhuber, 1997). The LSTM model is developed to overcome the drawback of the RNNs by adding a memory or cell state to the network. The cell state is responsible for adding or removing past information based on its relevance and importance to make predictions. It is currently extensively utilized because of its superior performance in precisely modeling both long- and short dependencies. LSTMs are commonly employed to solve problems in applications that deal with sequential data and have a more complicated structure than the RNN (Brownlee, 2017, Nielsen, 2015). It has S cell blocks connected in series, where S is the total time steps. Figure 3.3 illustrates the structure of an LSTM with C features and D hidden nodes. To regulate the state of each LSTM cell, there are three adjusting-gate blocks. . The gates are simple neural networks composed of weights, biases, and activation functions. The LSTM gates can be described as follows:

Forget gate: using information from the previous hidden state h_{t-1} and the present input x_t, this gate determines whether information from the cell state c_{t-1} (in Fig 3.4, the top horizontal line is colored orange) should be discarded. The weight matrix W_f is multiplied by current input x_t, but the recurrent weight matrix $R_f$is multiplied by the prior hidden-state h_{t-1}. These products outputs are combined to form a bias vector b_f. Lastly, a sigmoid function σ_g is used to generate the output vector f_t, which has values ranging from 0 to 1. The number "0" indicates that no information from the earlier time of the cell state is permitted to flow (this is not relevant information), whereas "1" indicates that all prior memory information is permitted to pass (extremely important). The function returns the result between "0" and "1" if the information is only partially relevant. This description can be expressed mathematically as follows:

$$f_t = \sigma_g(W_f x_t + R_f h_{t-1} + b_f) \qquad 3.3$$

The gate's activation function is denoted by σ_g, and all other elements and parameters are specified above. The weights W_f and R_f are matrices with dimensions $D \times C$ and $D \times D$, respectively, if the input x_t is a vector of sequences with C features and each cell contains D hidden units. The bias b_f, on the other hand, is a vector with D elements. As a result, the gate f_t's output is a D element vector.

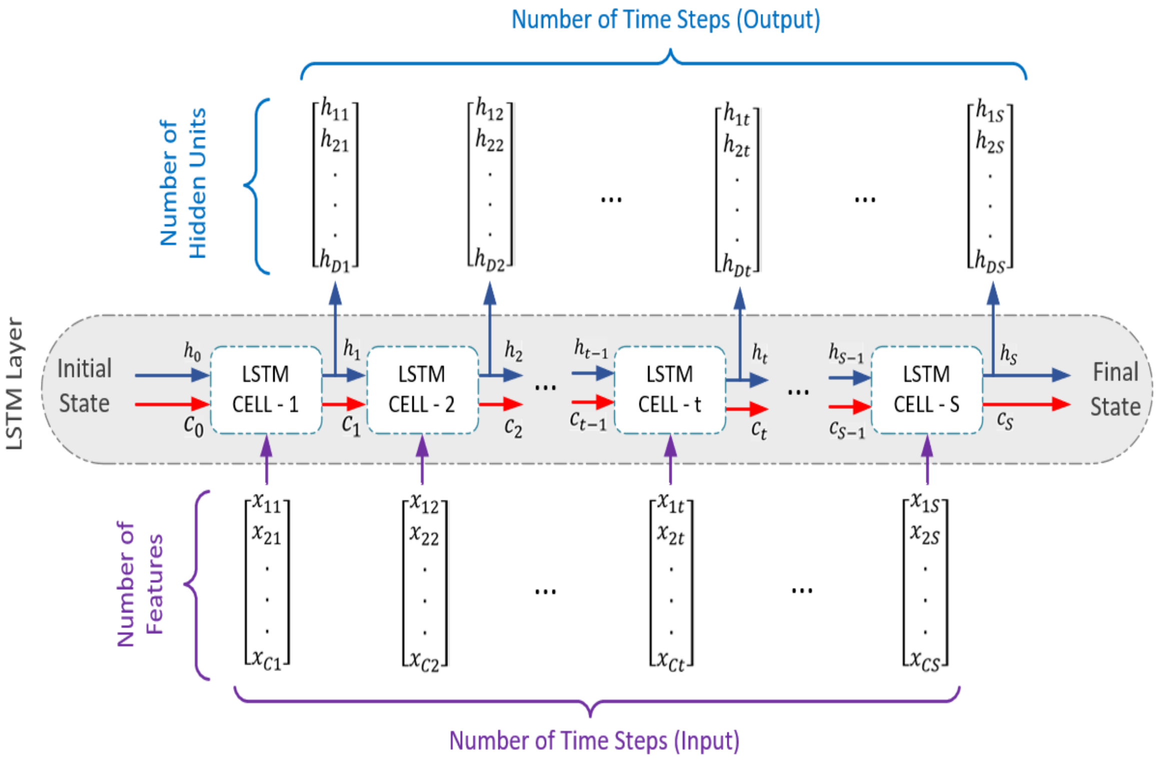

Number of Time Steps (Output)
Number of Hidden Units
$h_{11}, h_{21}, \ldots, h_{D1}$
$h_{12}, h_{22}, \ldots, h_{D2}$
$h_{1t}, h_{2t}, \ldots, h_{Dt}$
$h_{1S}, h_{2S}, \ldots, h_{DS}$
LSTM Layer
Initial State
h_0
c_0
LSTM CELL - 1
h_1
c_1
LSTM CELL - 2
h_2
c_2
h_{t-1}
c_{t-1}
LSTM CELL - t
h_t
c_t
h_{S-1}
c_{S-1}
LSTM CELL - S
h_S
c_S
Final State
Number of Features
$x_{11}, x_{21}, \ldots, x_{C1}$
$x_{12}, x_{22}, \ldots, x_{C2}$
$x_{1t}, x_{2t}, \ldots, x_{Ct}$
$x_{1S}, x_{2S}, \ldots, x_{CS}$
Number of Time Steps (Input)

Figure 3.3 An LSTM layer with multi-inputs and multi-outputs.

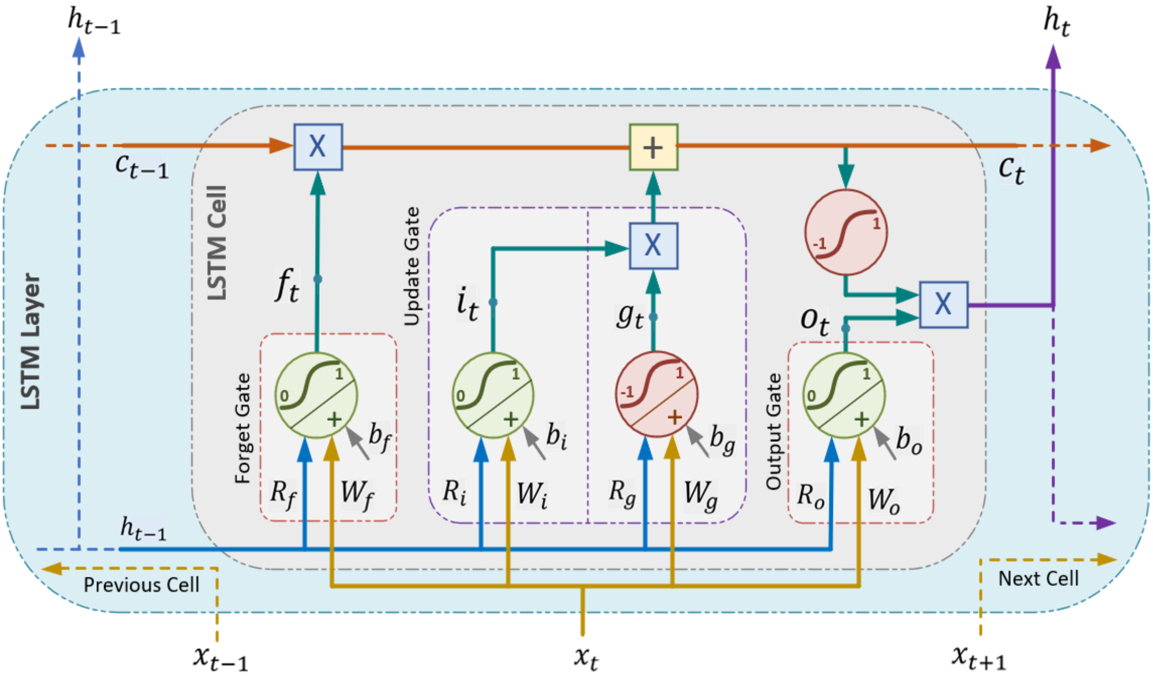


Figure 3.4 Internal structure of an LSTM cell.

Update gate: this gate updates the memory or cell state that was previously controlled by the forget gate. It is made up of two neural components: candidate cell g_t and input gate i_t, which are supplied with the same inputs as the forget-gate (x_t and h_{t-1}). The biases, weights, and activation functions of the i_t, and g_t branches, however, are different. The input x_t and h_{t-1} are weighted by matrices W_i and R_i, and biased with b_i, for the input-gate branch-i_t. Then, the sigmoid function σ_g was used to activate this branch.

The same is done with candidate-state branch-g_t, but instead, the designating letter g is used, and the sigmoid function $\sigma_g$is replaced with a tan hyperbolic (tanh or σ_s) which squishes the values between -1 and 1. The candidate state is controlled by the input branch, which controls the output of the squished data. Finally, the update gate output is produced by multiplying the outputs of these two neural networks, i_t and g_t.

The two networks can be represented mathematically as follows:

$$i_t = \sigma_g(W_i x_t + R_i h_{t-1} + b_i) \quad 3.4$$

$$g_t = \sigma_s(W_g x_t + R_g h_{t-1} + b_g) \quad 3.5$$

Where σ_s stands for state activation function.

The currently hidden state h_t is computed using this gate. After multiplying with the corresponding weights W_o and R_o, and adding to the bias b_o, transmit a copy of the combined input (x_t and h_{t-1}) to a sigmoid function σ_g. After squishing to the range [-1, 1] with the tanh function σ_s, the output o_t is multiplied by the current cell state c_t. The output gate can be mathematically expressed as follows:

$$o_t = \sigma_g(W_o x_t + R_o h_{t-1} + b_o) \quad 3.6$$

The new cell-state c_t and hidden-state h_t equations are as follows:

$$c_t = f_t.* c_{t-1} + i_t.* g_t \quad 3.7$$

$$h_t = o_t.* \sigma_s(c_t) \quad 3.8$$

The Hadamard multiplication is denoted by the operator $*$ (element-wise or pointwise operation). Because we are using MATLAB to train our network, the software's parameter and variable names are used in this study.

To summarize, the forget gate chooses which information from the old memory should be kept and which should be forgotten. The input gate is used to generate the current memory and update the relevant memory in the following block. The output gate is used to calculate the current block output as well as the next hidden state .It is important to note that all of the gates have identical inputs: three replicas of the prior hidden state paired with the current input (the bottom line in Fig 3.4). The LSTM memory or cell state shown at the top of Fig 3.4 is used by the network to learn about the order of input data sequences.

3.5 Gated Recurrent Unit (GRU)

The GRU model is a simplified and newer version of LSTM. It is composed of two gates and one candidate-state network, namely: reset gate r_t, update gate z_t, and candidate state $\tilde{h}_t$. The update gate used by the GRU is equivalent to the forget and input gates in the LSTM model combined as a single network as illustrated in Fig 3.5. It is used to figure out what information should be removed or added. The reset gate is used to determine how much information from the previous state to forget. In contrast to the LSTM, there is no cell state in the GRU network (Huang et al., 2019). In other words, the prior hidden state h_{t-1}can be considered as the cell state. The network parameters of the GRU are less than those in LSTM and hence the network requires less training time to learn about dependencies among the time-step observations or sequence data.

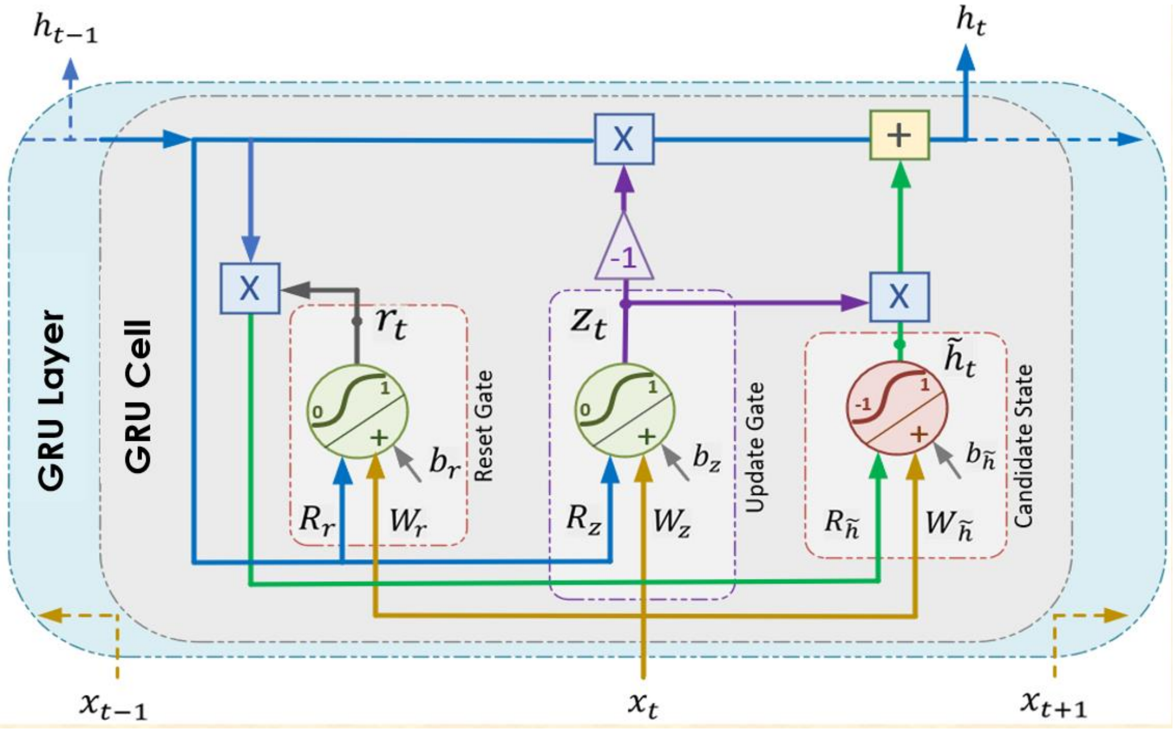


Figure 3.5 Internal structure of a GRU cell.

Mathematically, the following equations are used for the reset and update gates, candidate state, and the hidden state, respectively:

$$r_t = \sigma_g(W_r x_t + R_r h_{t-1} + b_r) \quad 3.9$$

$$z_t = \sigma_g(W_z x_t + R_z h_{t-1} + b_z) \quad 3.10$$

$$\tilde{h}_t = \sigma_s(W_{\tilde{h}} x_t + r_t R_{\tilde{h}} h_{t-1} + b_{\tilde{h}}) \quad 3.11$$

$$h_t = (1 - z_t) .* \tilde{h}_t + z_t .* h_{t-1} \quad 3.12$$

CHAPTER FOUR: RESULT AND DISCUSSION

4.1 Introduction

In this chapter, the datasets and system settings employed in our experiments are introduced to assess the performance of the proposed model. MATLAB Simulink has been used to create systems with multi-domain models in a block diagram context. Then it provides the results of the proposed model and compares different cases with others. Furthermore, it is discussed how converting a single input sequence to several input sequences can save time, reduce error, and improve performance.

4.2 Historical Data Augmentation (HDA)

A deep neural network and historical data augmentation (DNN–HDA) is proposed by (Lai et al., 2020) for data with high correlation which shows a great improvement in the accuracy. The method is based on dividing the input data into multiple sequences, each sequence represents a dataset for one year, as shown in Fig (4.1). However, when data is divided into multiple parts, information about the connection between the end of one part and the beginning of the next part is missing. For some data and load forecasting problems, this could be unproblematic. However, when the nature of data changes and includes high uncertainty and fluctuations in the time step information, this approach was found struggling to predict future load demand especially for long-term forecasting.

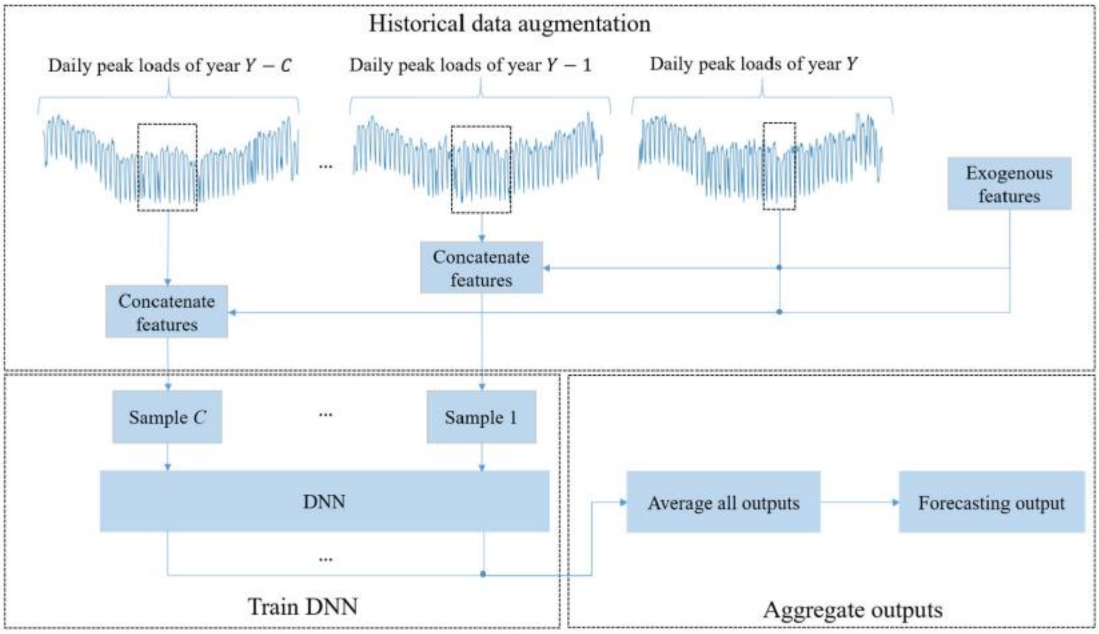


Figure 4.1 Historical data augmentation structure.

As shown, assuming that the loads are forecasted in year Y and there are C-year historical data. In the procedure of HDA, for each load to be predicted, several features could be used to construct C samples. After generating samples, the DNN is trained with the training data. For one predicted load, there are C outputs that will be averaged to generate the final prediction.

To connect the two ends of the yearly sub-sets, this study proposes the use of a two-year sub-dataset. Figures 4.2a-d illustrates and display the differences, once applied to Erbil load data, it causes issues and unconnected data. When comparing it with two years of input per sequence the percentage of an error is higher. Whereas the average error in HDA is 634.0072/2696 or %23, it is reduced to 199.4247 or %7.40 in the proposed model. Where 2696 MW is the dataset's maximum load demand for the years 2015 to 2020. Also, the expected shape for the forecast data is completely different from the training and tested data shapes.

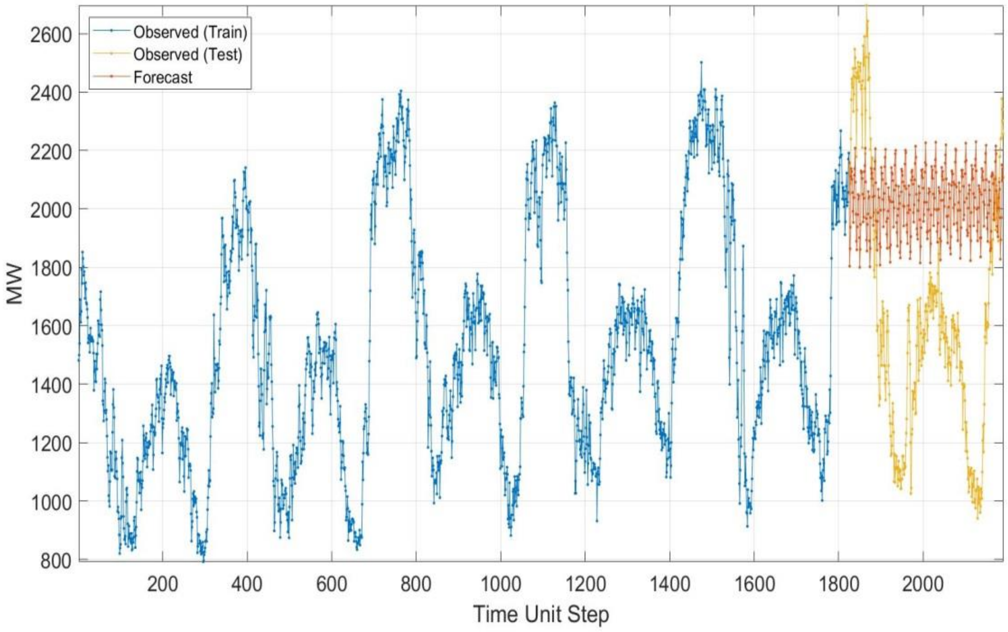


Figure 4.2a Forecasting one year, by one year input per sequences / (HDA) approach.

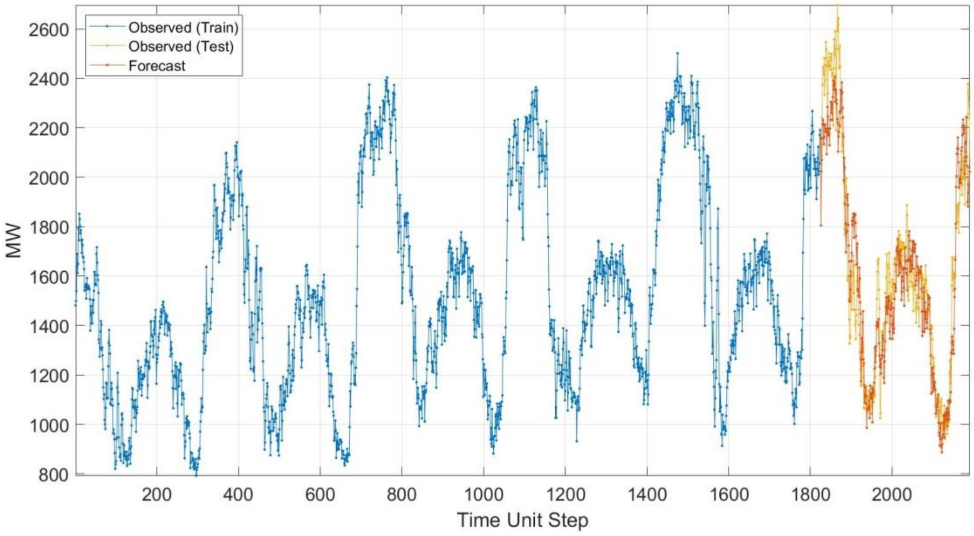


Figure 4.2b Forecasting one-year, by two years of input per sequence.

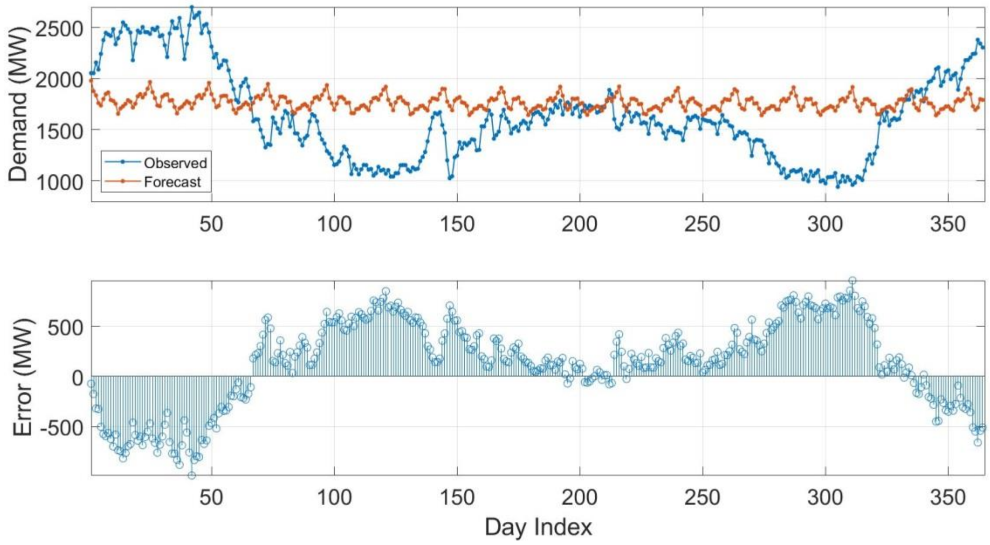


Figure 4.2c Error rate , by HDA approach.

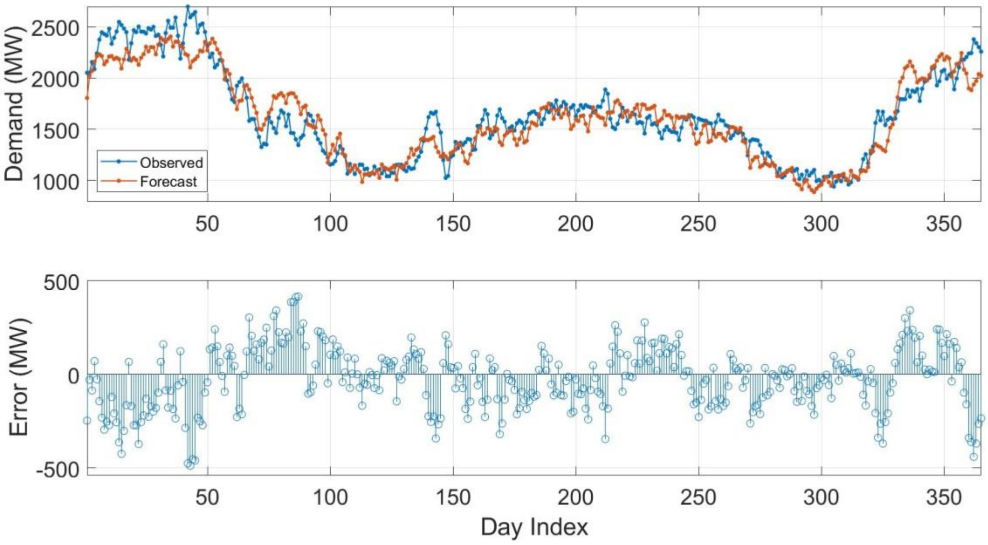


Figure 4.2d Error rate , by two years of input per sequence.

4.3 Dataset Description

In this study, a historical dataset is collected for the Kurdistan regional power system containing load profiles for Erbil governorate for the range of years 2015 to 2020.The daily consumption of the load dataset was used to train our model. The Erbil load dataset consists of the following data: date, average demand, average load, maximum and minimum demand, and maximum and minimum load (MW/h) attributes.

Training and test subsets are separated from the rest of the data. The data from the first five years are utilized for training the network, while the data from the last year is used to test the trained network. These two subsets are referred to as XTrain and XTest, and they represent the predictors or independent variables for the respective training and test datasets, respectively. The response or dependent variables for the relevant training and test datasets YTrain and YTest are generated by moving moving the predictors in a one-time step.

A correlation analysis is performed on the sample data in order to see how the dataset for one year is connected with another yearly dataset from the same time-series sequence. Correlation coefficients between all pairs of variables in the matrix of time series data of the (Erbil_load) dataset are plotted in Fig 4.1 by using the *[R,PValue] = corrplot(X)* equation in MATLAB. The plot is a numVars-by-numVars grid, where numVars is the number of time series variables (columns/sequences) in X. The equation returns the correlation matrix in the plots R as well as a matrix of p-values, R denotes a relationship or correlation between variables and PValue denotes a test of the null hypothesis that each pair of coefficients is uncorrelated against the alternative hypothesis of a nonzero correlation (MATLAB, 18/6/2022 Retrieved).

Figure 4.3 depicts the six-year association among pairs of time-series data, where the x-axis and y-axis represent daily Erbil load in megawatts from 2015 to 2020.

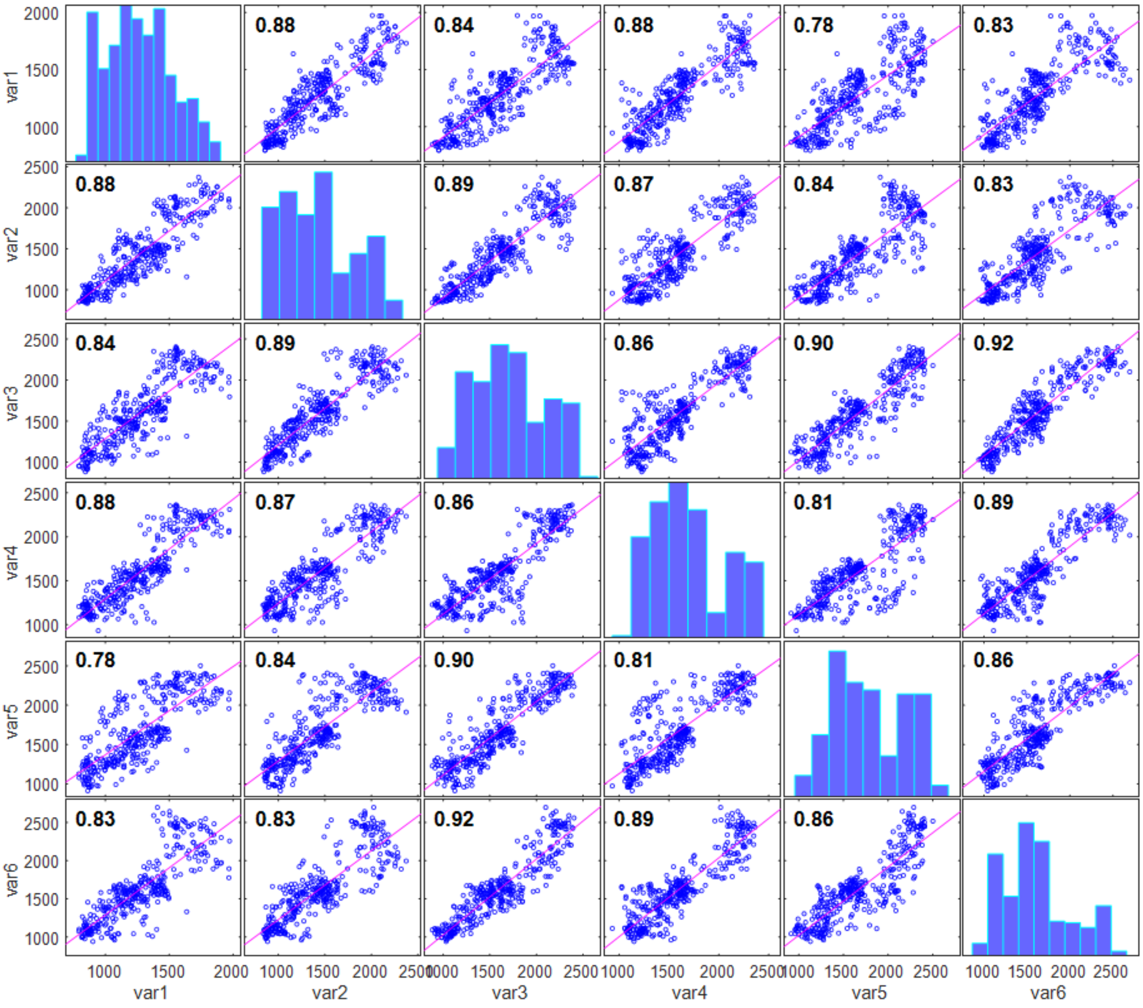


Figure 4.3 Correlation analysis of the input data.

The histograms of data are displayed diagonally, and this position has a value of R=1 due to the correlation between the same years. The scatter plots of pair variables are shown in the off-diagonal figures. A scatter plot is a graph that illustrates the relationship between two numeric values. Every element of the dataset is represented by a point with x-y coordinates corresponding to the two variable values (Jacoby, 2000). R is a number that ranges from 0 to 1. There is no correlation if the value is zero, and there is a perfect correlation if the value is one. When the R is negative, the variables are inversely related (Akoglu, 2018). The correlation coefficients for each pair of variables are highlighted on the graph and listed in table 4.1.

Table 4.1 Numerical values for the correlation analysis.

Erbil_LoadData **Corelation**	**Year-2015**	**Year-2016**	**Year-2017**	**Year-2018**	**Year-2019**	**Year-2020**
Year-2015	**1**	0.8828	0.8374	0.8793	0.7777	0.8257
Year-2016	0.8828	**1**	0.8935	0.8708	0.8352	0.8341
Year-2017	0.8374	0.8935	**1**	0.86	0.8953	0.9167
Year-2018	0.8793	0.8708	0.86	**1**	0.8118	0.8899
Year-2019	0.7777	0.8352	0.8953	0. .8118	**1**	0.8613
Year-2020	0.8257	0.8341	0.9167	0.8899	0.8613	**1**

It can be observed from these plots and the table that the input loads used in this study are highly correlated. The minimum and maximum correlation coefficients are 0.777 and 0.9167, respectively, and the average of these off-diagonal values is 0.8991. The implicit relationships motivate us to investigate the use of this nature in the historical data to improve load forecasting. As mentioned earlier, one recent study (Lai et al., 2020) observed this correlation using another dataset and introduces the concept of historical data augmentation (HDA). However, for high uncertainty data with fast changes in the time step information, the use of one-year data for training a long-term dataset is a challenging problem; the data corresponding to the end of one year has no connection with the beginning of the next year dataset.

In fact, if the starting day of historical data marks the first day of a year, then the data starts in the middle of a winter season which has a similar load-profile shape concerning the loads obtained for the end of the previous year. Therefore, in the results section, a simple modification of this method is proposed to remove this shortcoming and accelerate the process.

4.4 Results

To predict the future values of load demands, one time-step ahead forecasting (OTSAF) or multiple time-steps ahead forecasting (MTSAF) can be used. For their future prediction, both approaches use an initial value computed from the last time-step of the historical load demands. However, the difference between OTSAF and MTSAF is in the way the network is updated for the next predictions. OTSAF updates the network using the current value of the test data whereas MTSAF updates the network from the current predicted value. In other words, in MTSAF, the test dataset is not used anymore for future time-step prediction except for the first one. For the rest of remaining predictions, it loop over the predicted values once at a time until the end of the time-step sequence. It should be noted that for the OTSAF, the network state needs to be reset to prevent the influence of past predictions on new data forecasting.

In this study, single and multi sequences input-output forecasting scenarios are addressed. The historical load demands are used as the input for single sequence prediction. Weather data, load demands, and week-day and holiday information are among the inputs for the multi-sequence. In addition to the traditional way for data inputs with a whole series of time steps, a change to the data input is offered by separating the data into many subsets and considering a two-year period per subset rather than a single-year dataset.

This study includes different models with separate network training settings. The models are decided empirically starting from a single LSTM model with default values. The number of recurrent neural networks is increased gradually until a satisfying result is obtained. Most of the models need at least three blocks of LSTM, GRU, or a combination of them with a fully connected layer to get an acceptable accuracy. The gradient threshold is set to 1.0 to avoid any exploding in the network update. The initial value of learning rate is chosen to be in the range 0.001 to 0.01 to balance between training time and model accuracy. Reducing this value increases the training time but might reduce the error. The maximum number of epochs is not fixed here and it varies from a network to network according to the complexity of the model and pattern of the input data. To evaluate our models, the forecasting error is calculated using the root mean square error (RMSE)**,** which is the difference between both the forecasted and real load data.

4.4.1 One Time-step Ahead Forecasting, OTSAF Approach

We begin with the results of the traditional OTSAF model, in which the input data is presented as a single set of time-series load demands without being divided into subgroups and without taking into account other variables including time or weather data. Figure 4.4a illustrates the data for training colored blue for the years 2015 through 2019 then the test data for 2020 and lastly the forecasting values put over the test data for comparison. The x axis represents the day index, which begins on January 1, 2015 and ends on December 31, 2020, whereas the y axis represents the load demands in megawatts. A deep neural network is a network that is empirically chosen, use three LSTM layers with 128 hidden units each which setting by default.

Figure 4.4b displays the observed and forecasted values for the dataset's last year, 2020, highlighting the various MW errors. For this setting, the RMSE is 84.3636/2696 MW, and the relative percentage error is 3.12. It is important to note that MATLAB and Simulink programs produce exactly the same results. A maximum of 100 epochs (Fig. 4.4c) as shown to be sufficient for achieving the above accuracy in a network using OTSAF. On a typical PC, training the network using the method took about six minutes.

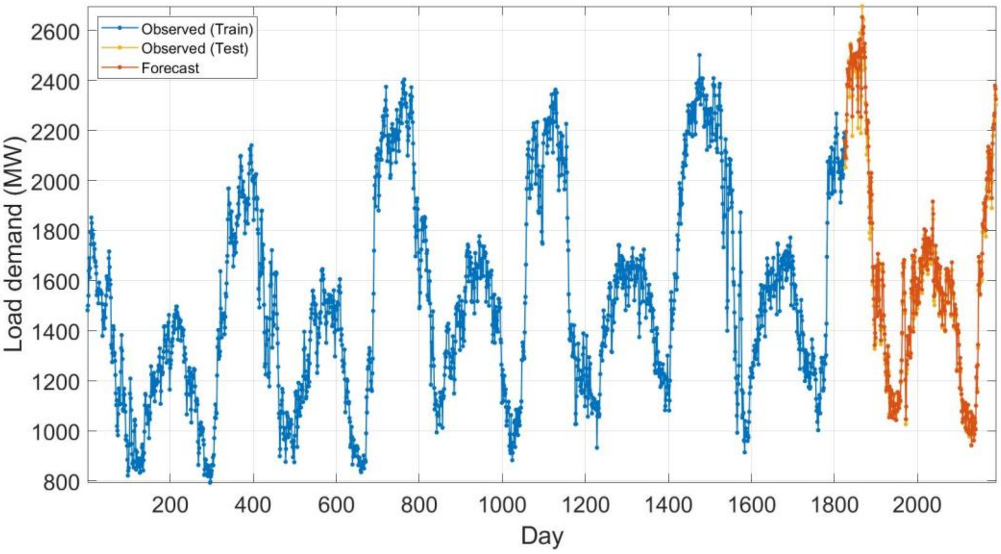


Figure 4.4a OTSAF Model for Load Forecasting

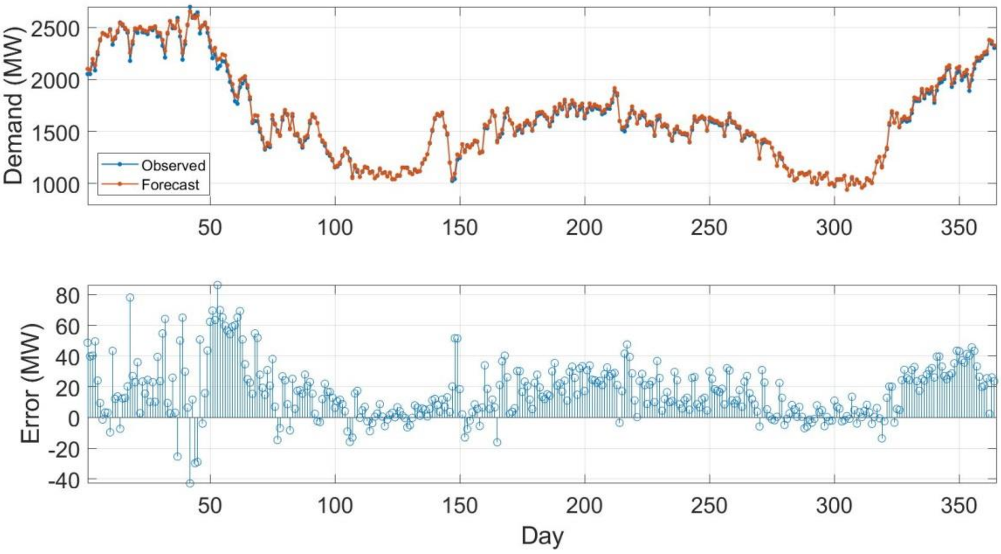
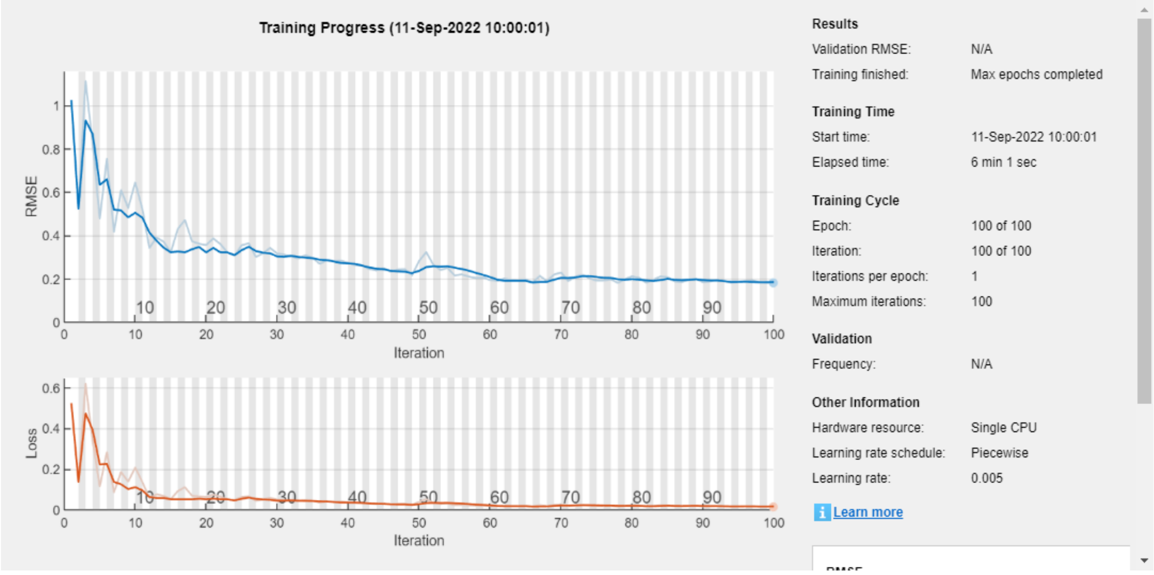


Figure 4.4b Errors in model output–OTSAF



Results

Validation RMSE: N/A

Training finished: Max epochs completed

Training Time

Start time: 11-Sep-2022 10:00:01

Elapsed time: 6 min 1 sec

Training Cycle

Epoch: 100 of 100

Iteration: 100 of 100

Iterations per epoch: 1

Maximum iterations: 100

Validation

Frequency: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Piecewise

Learning rate: 0.005

Learn more

Figure 4.4c OTSAF model for training and loss function errors.

4.4.2 Multi Time-step Ahead Forecasting, (MTSAF) Approach

On the same data set, we apply the MTSAF method and create a network that can learn from the five years of training data and forecast the load demand for the next year. The network architecture is selected experimentally and it consists of two layers of LSTM on the top connected to two layers of GRU on the bottom. The number of hidden units for these layers is chosen empirically to be 128, 64, 32, and 16, respectively. A maximum number of 1500 epochs are chosen to train the network with a learning rate of 0.001. The results shown in Fig. 4.5a–c show that the network learned from training the data and predicting the next-year forecasting given only a single-day initial value and loop over until the end of the year.

Compared to the case of OTSAF, the relative RMSE error is 216.08/2696 or 8.01% which is higher than the previous case. This is expected as we know that OTSAF is a one-day ahead forecasting whereas MTSAF here is a 365-day ahead forecasting. It is worth pointing out that this method of updating network parameters requires a relatively long training time. Compared to OTSAF, the network training process with MTSAF takes about eight times longer.

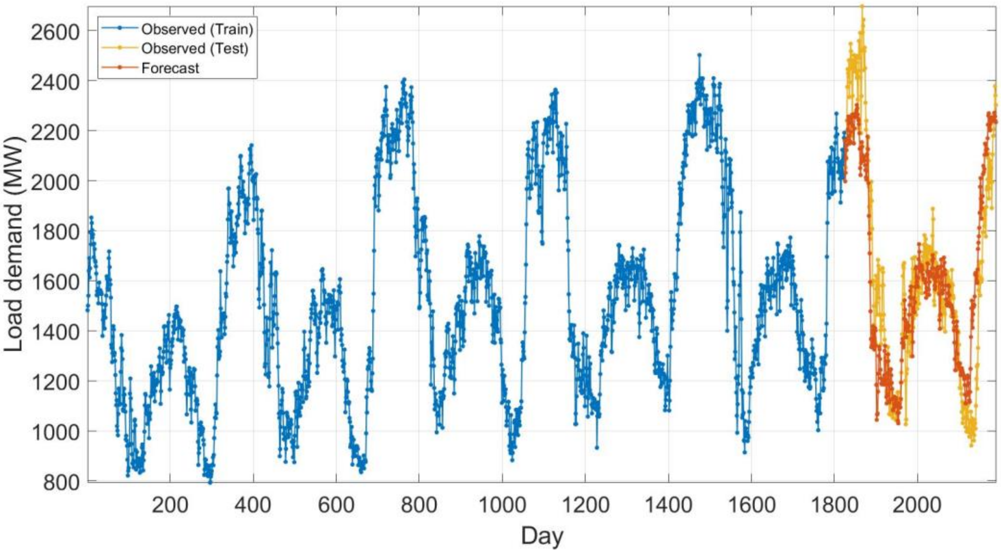


Figure 4.5a MTSAF model for load forecasting

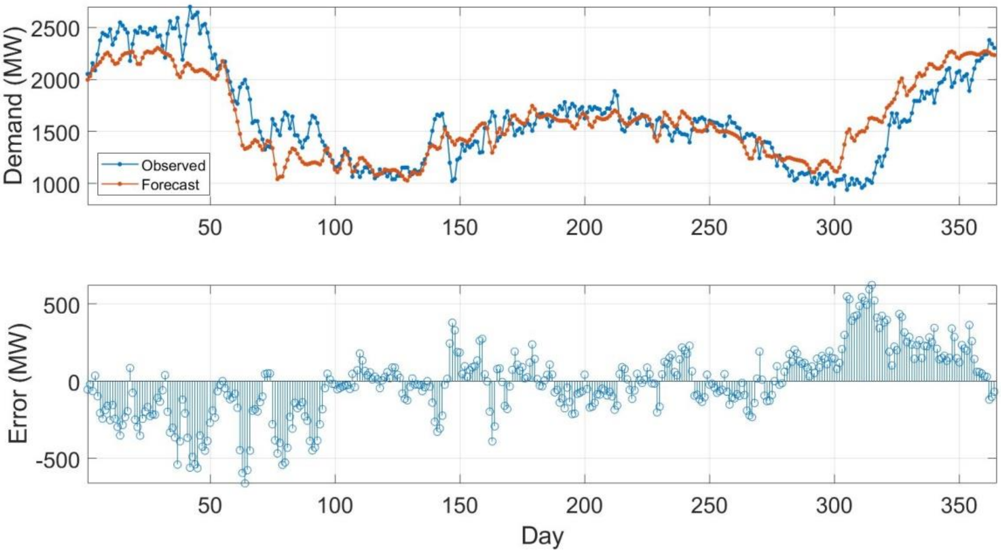
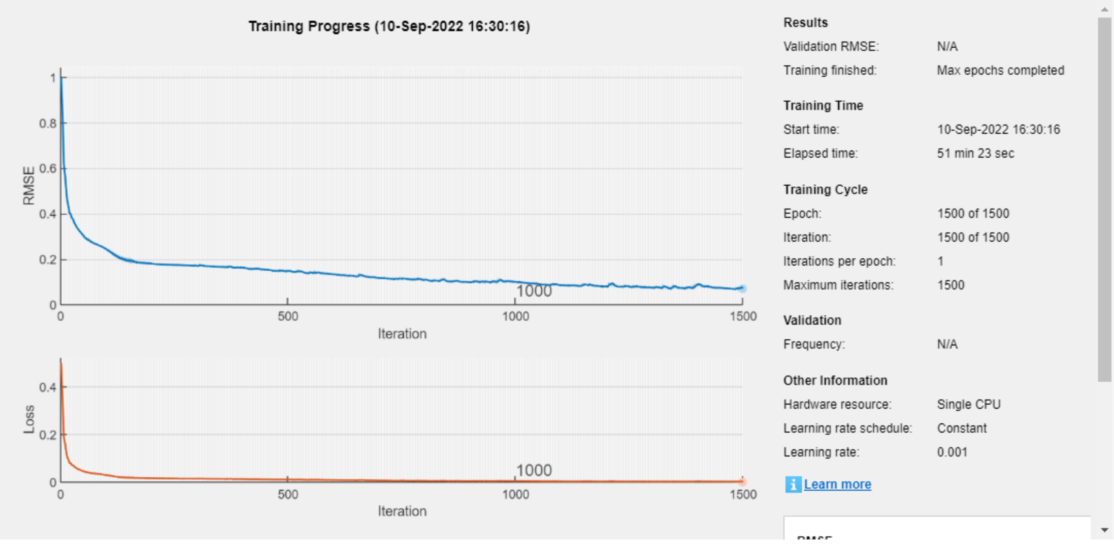


Figure 4.5b Errors in model output–MTSAF



Results
Validation RMSE: N/A
Training finished: Max epochs completed

Training Time
Start time: 10-Sep-2022 16:30:16
Elapsed time: 51 min 23 sec

Training Cycle
Epoch: 1500 of 1500
Iteration: 1500 of 1500
Iterations per epoch: 1
Maximum iterations: 1500

Validation
Frequency: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

Figure 4.5c MTSAF model for training and loss function errors.

4.4.3 Single Sequence per Variable

The load demands are employed as an input for forecasting. The network can be trained using multivariable inputs including weather data, weekday, and weekend information. The necessary data for the average daily temperature for the region is collected and preprocessed. The one-hot encoding technique is used for the calendrical variables so that it does not give more weights to week-day variables. However, the weekend days have different one-hot values owing to the reduction in power consumption during these days. Categorical data are transformed into numerical data via one-hot encoding. Binary features are created from categorical features, therefore if a feature is represented by a certain column, it obtains a 1. Otherwise, it receives a 0. The network is trained with the above four input variables. The corresponding errors for the output variables are 206.5580, 3.6647, 0.5347, and 0.5353, respectively. The relative percentage error for the load demand is calculated to be 7.66%, and the results are plotted and displayed in Fig. 4.6a–c. Because each variable has a lengthy sequence, it takes a long time. The outcome of the load during the computation of the factors generally changed, decreasing from 8.01 to 7.66.

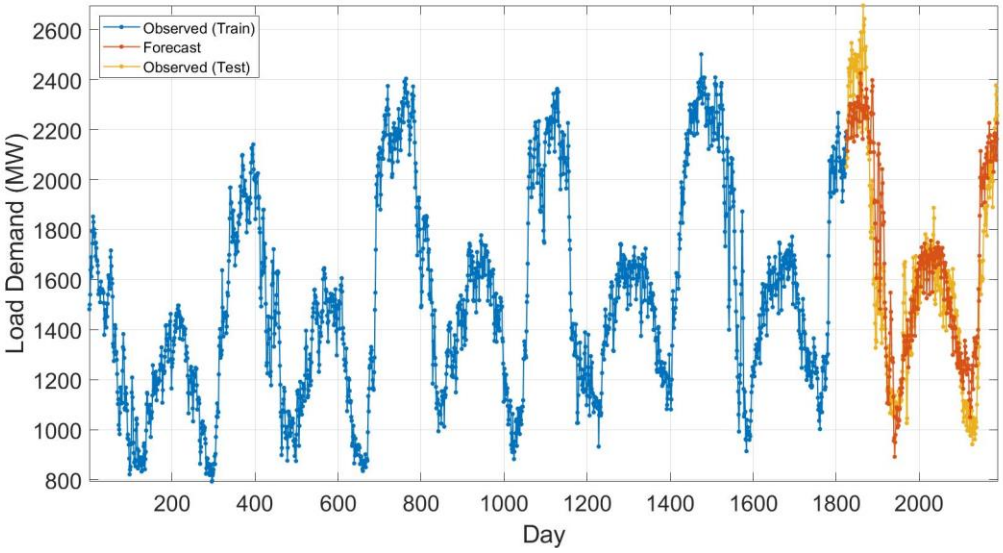


Figure 4.6a Single sequence per variable model for load forecasting

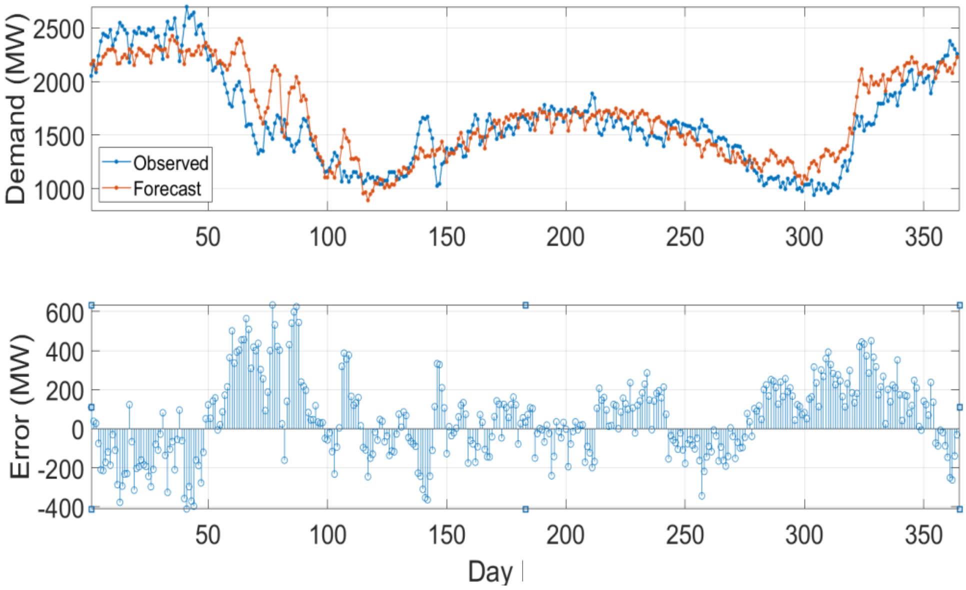
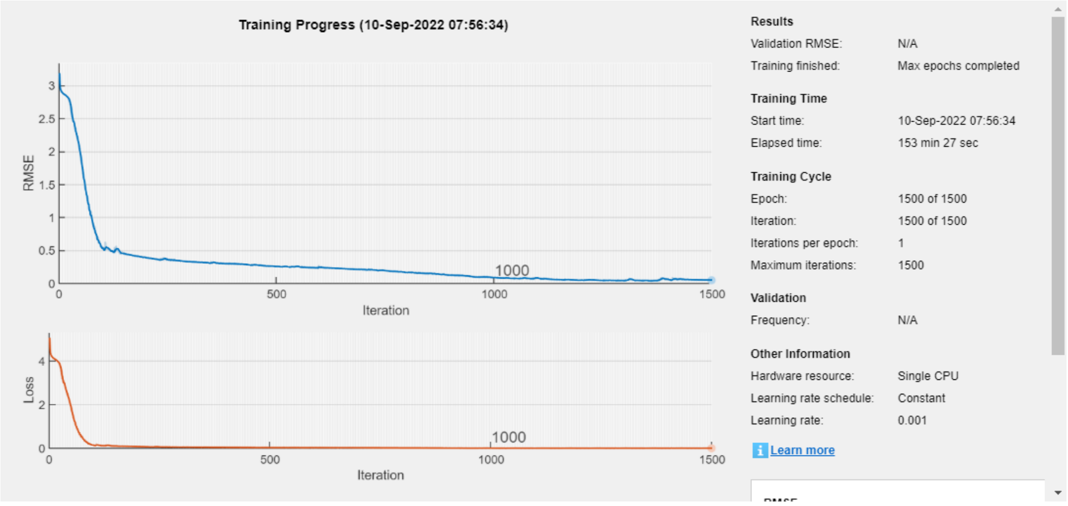


Figure 4.6b Errors in single sequence per variable model



Results

Validation RMSE: N/A

Training finished: Max epochs completed

Training Time

Start time: 10-Sep-2022 07:56:34

Elapsed time: 153 min 27 sec

Training Cycle

Epoch: 1500 of 1500

Iteration: 1500 of 1500

Iterations per epoch: 1

Maximum iterations: 1500

Validation

Frequency: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.001

Learn more

Figure 4.6c Single sequence per variable model for training and loss function errors

4.4.4 Multi Sequences with Single-Variable

The current scenario is when the input is separated into several subsets, each of which is for two consecutive years (or the same year twice) in order to connect the two ends of the year. As a result, our model has five-sequence inputs. For this case, an LSTM-GRU hybrid network is used, with the same number of hidden units and layers just like the MTSAF case. The average error rate is determined to be 199.4247/2696 MW, with the corresponding value of 7.40, which is lower than the error in the MTSAF model single-sequence case (8.01).

Not only is the error smaller, but the training period is also significantly less than in the MTSAF scenario. The results are plotted and shown in Fig. 4.7a–c. It is worth mentioning that the previous forecasting method used for the data augmentation failed here to learn from the data and predict the next 365-day demands using the exact training settings and network structure above. The gap of information between the starting and ending points of the yearly dataset had a significant impact on model accuracy.

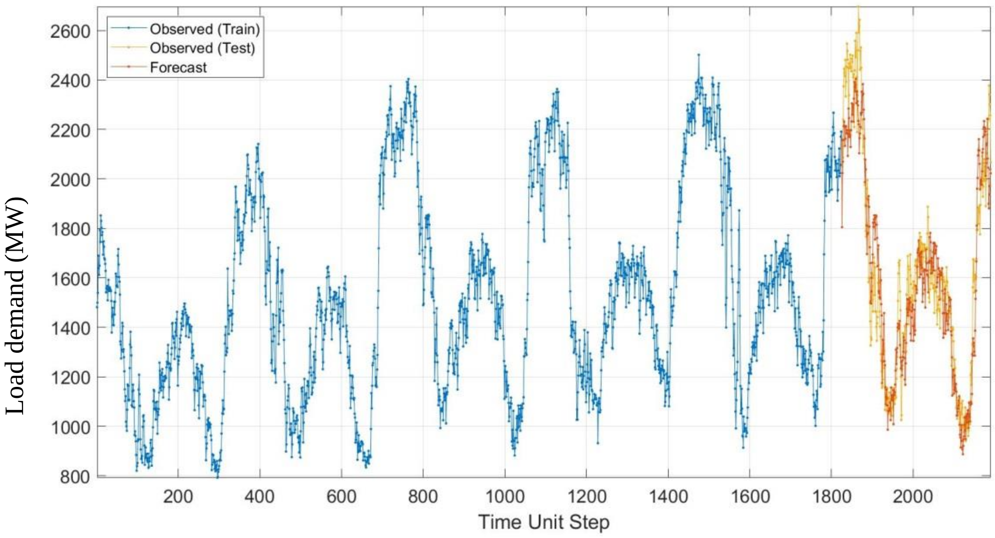


Figure 4.7a Single variable multi sequences model for load forecasting

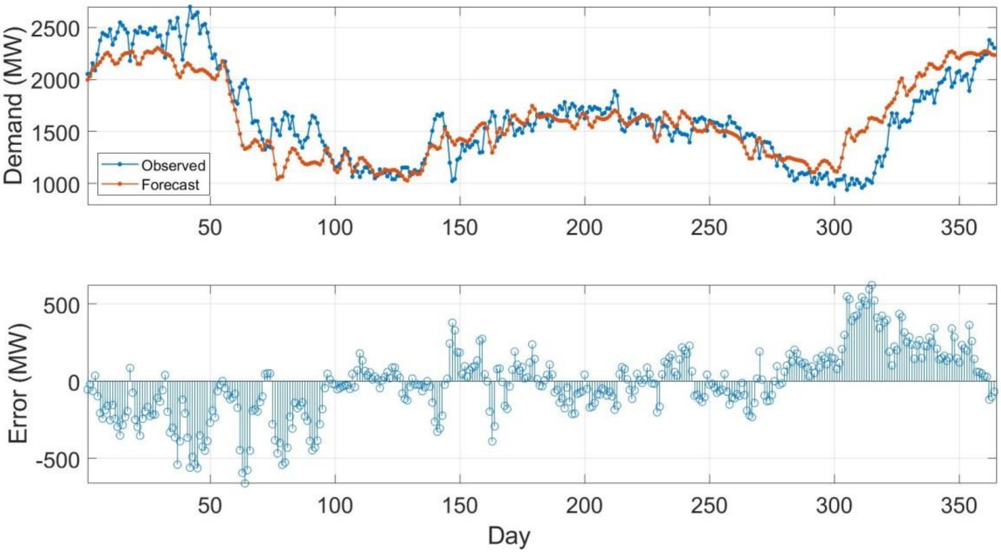
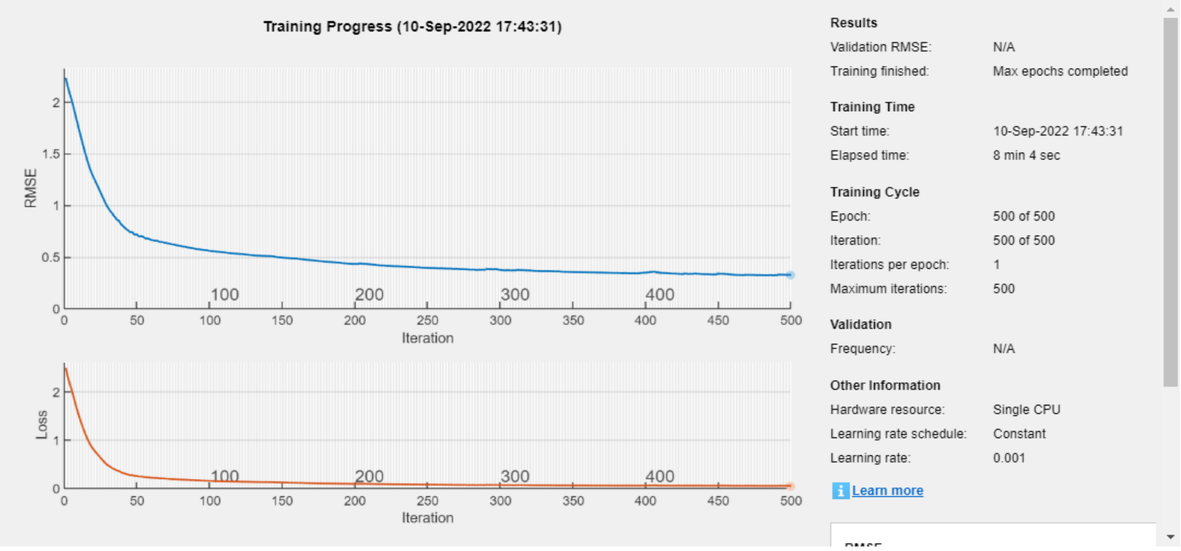


Figure 4.7b Errors in single variable multi-sequence model



Results

Validation RMSE:	N/A
Training finished:	Max epochs completed

Training Time

Start time:	10-Sep-2022 17:43:31
Elapsed time:	8 min 4 sec

Training Cycle

Epoch:	500 of 500
Iteration:	500 of 500
Iterations per epoch:	1
Maximum iterations:	500

Validation

Frequency:	N/A

Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.001

Figure 4.7c Single variable multi sequences model for training and loss function errors.

4.4.5 Multi-Sequences per Variable

By dividing the demand sequence into several training yearly subsets, it is possible to forecast future load demands using multi-input data augmentation. For this study case, the same input variables as in the (single sequence per variable) case are used. Load demand, averaged-daily temperature, weekday information, and weekend day data are the input factors. The total input sequences are 20 sequences, five per each input variable representing the five-year training sets. The network is trained and simulated with the test data. The average errors for each variable are computed to be 198.7859, 2.10207, 0.4669, and 2.9073 respectively. According to calculations, the load demand relative percentage error is 7.37%, which is lower than the single sequence per variable model equivalent to classical inputs. Figure 4.8a–c shows the outcomes of the network training and model analysis for this situation.

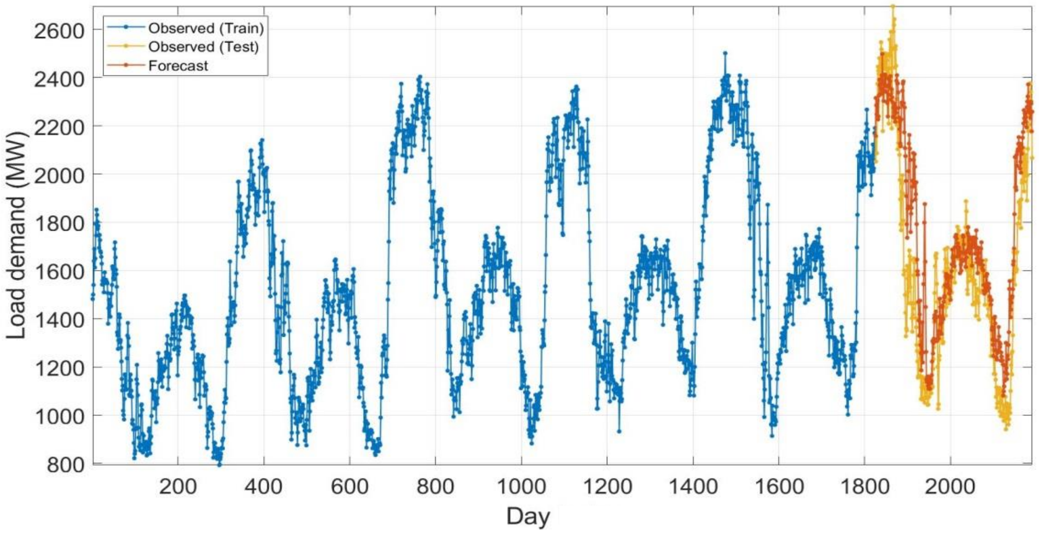


Figure. 4.8a Multi varible multi sequences model for load forecasting

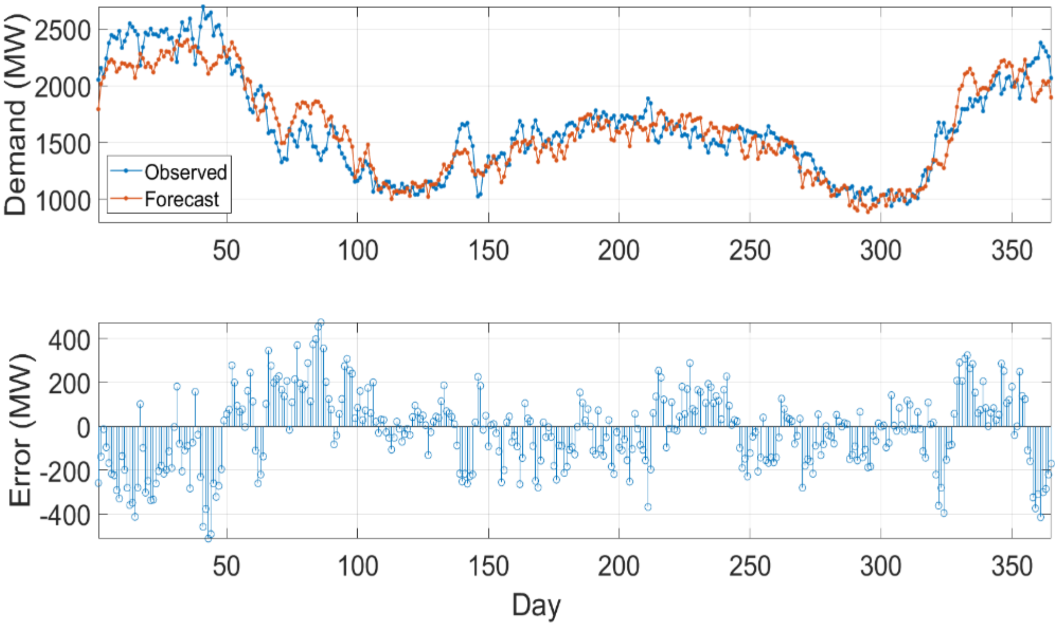
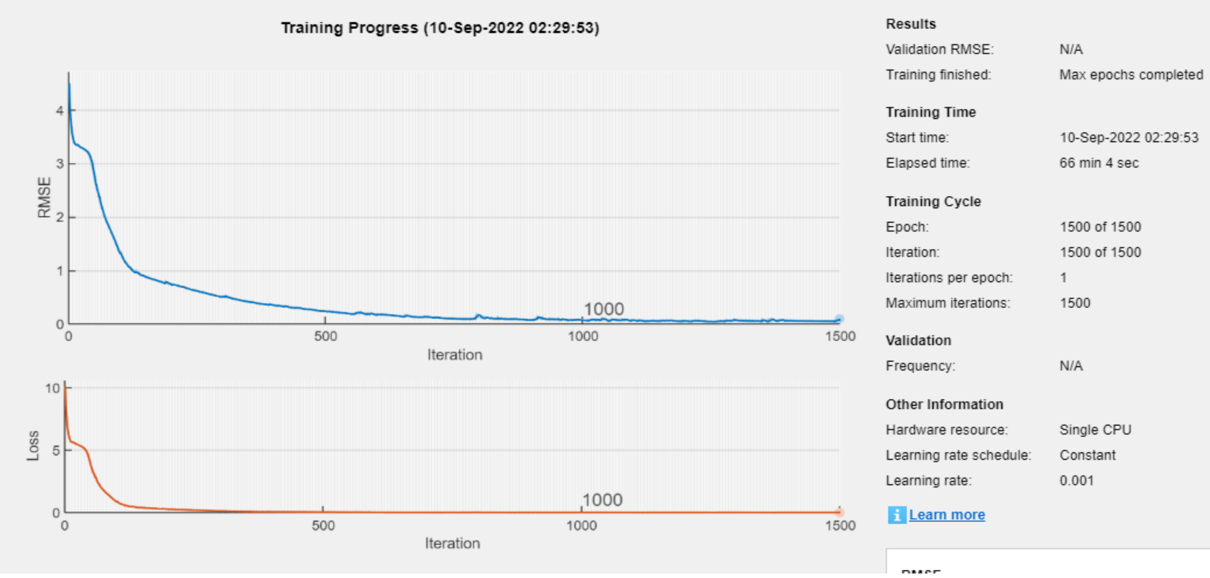


Figure. 4.8b Erorrs in multi varible multi sequences model for load forecasting



Results	
Validation RMSE:	N/A
Training finished:	Max epochs completed
Training Time	
Start time:	10-Sep-2022 02:29:53
Elapsed time:	66 min 4 sec
Training Cycle	
Epoch:	1500 of 1500
Iteration:	1500 of 1500
Iterations per epoch:	1
Maximum iterations:	1500
Validation	
Frequency:	N/A
Other Information	
Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.001

Figure. 4.8c Multi variable multi sequences model for training and loss function errors.

4.4.6 Forecasting Loads with Different Sampling Rates

All the networks created above are for input data with a daily sampling rate of one prediction. Investigating the problem of the same data input but different sampling rates, like one prediction each week, will be beneficial. The input is smoothed with MATLAB Gaussian function for comparison purposes. The difference between the original and smoothed data is shown in Fig 4.9a. The previously discussed OTSAF example is reproduced here with a new sample rate and smoothed input. The findings are displayed in Fig. 4.9a–d, and the network is DL with the same structure as the OTSAF. The relative error is calculated to be 10.1117/2696 corresponding to a value of 0.37, which is small enough to allow for reliable load forecasting. For the remaining models in this study, the same technique can be used.

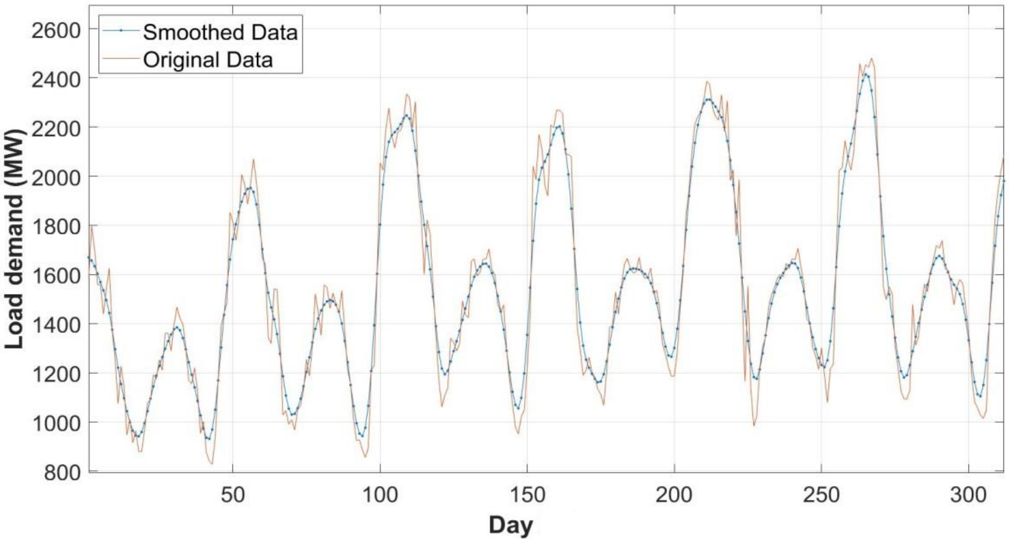


Figure. 4.9a Load forecasting, original and smoothed weekly data.

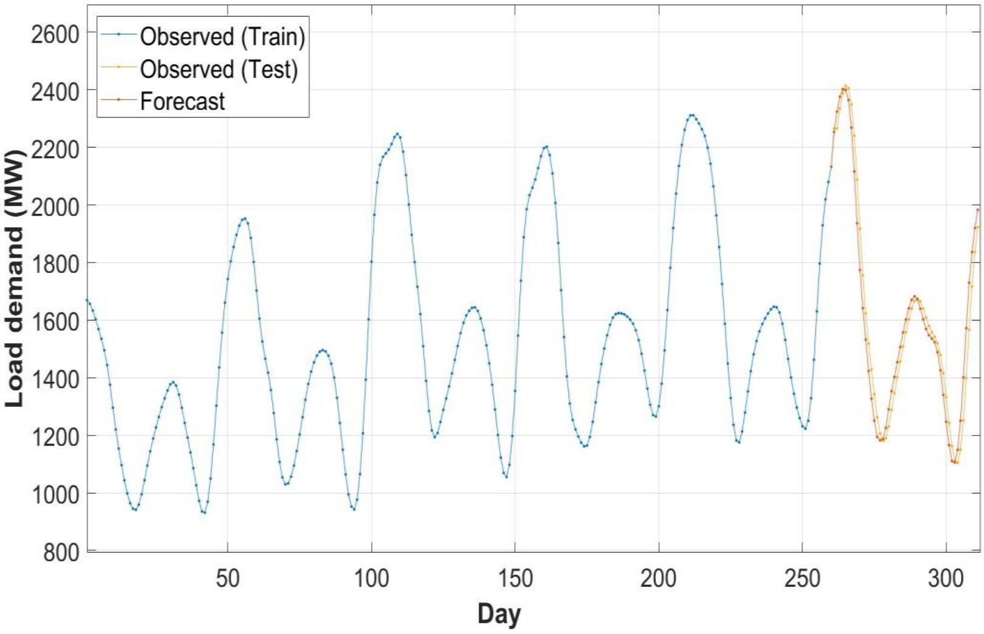


Figure. 4.9b Load forecasting, smoothed weekly data.

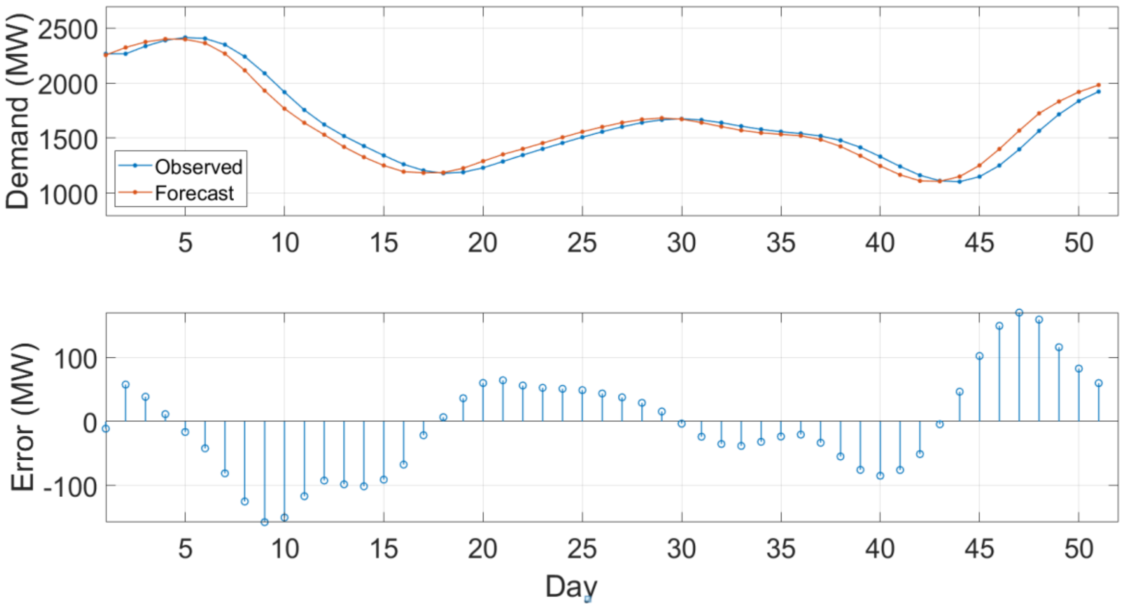
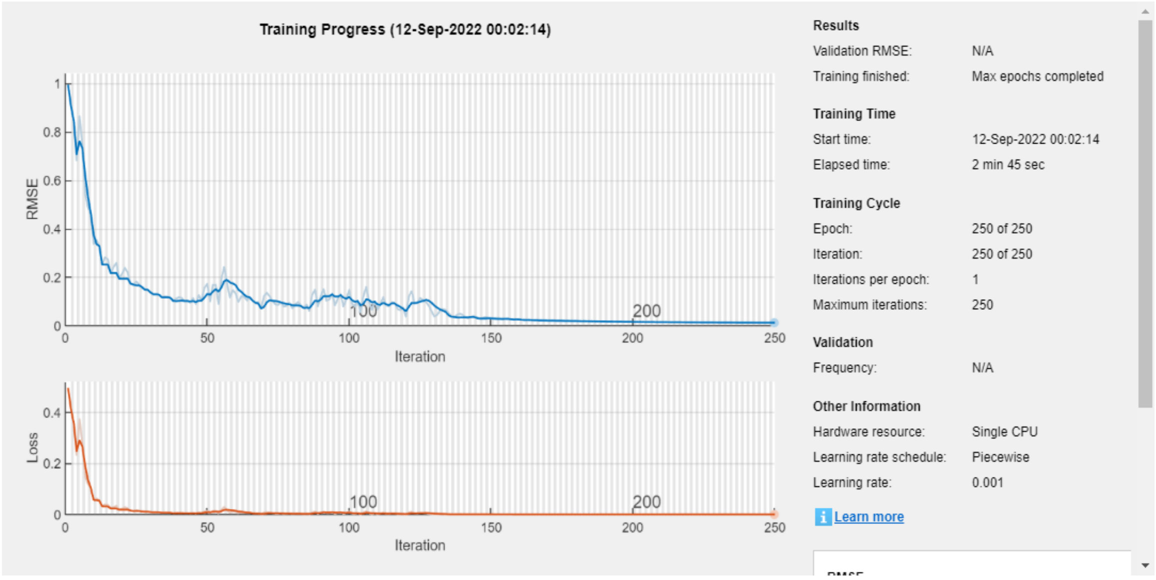


Figure. 4.9c Forecasted errors– smoothed weekly data.



Results

Validation RMSE: N/A

Training finished: Max epochs completed

Training Time

Start time: 12-Sep-2022 00:02:14

Elapsed time: 2 min 45 sec

Training Cycle

Epoch: 250 of 250

Iteration: 250 of 250

Iterations per epoch: 1

Maximum iterations: 250

Validation

Frequency: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Piecewise

Learning rate: 0.001

Figure. 4.9d Training and loss function errors– smoothed weekly data.

4.5 Simulink Model

This section presents the Simulink model developed for load forecasting and applied to the OTSAF and MTSAF methods. Simulink is a visualized version of MATLAB and is bidirectionally connected to MATLAB. It has several advantages over coding in MATLAB. For instance, you can see how the algorithm works by looking into the block diagram shown as a chart for the problem. It is easier to understand how the algorithm works visually by separating the essential processes into blocks that are connected to one another. An additional built-in or customizable component can be easily set up and added to the model. The blocks and the signals can hold values in the form of scalars or vectors. The block diagrams for OTSAF and MTSAF approaches can be seen in Fig. 4.10 (a,b).

In the OTSAF, a variable from the test dataset is required for each time step to predict the next load demand. The network will have a vector of test inputs (or matrix in the case of multiple sequences). The three main processes in the OTSAF approach are standardization, prediction, and unstandardization. The first block is standardization. In order to standardize or rescale the range of features in the input data set, standardization is an important process that is frequently used as a pre-processing step in many ML models. Additionally, feature scaling speeds up the training and convergence of ML algorithms. The process of the standardization block is done, once the standard deviation and mean for the test data are obtained. To reset the network after each prediction, the standardization block requires a counter block. The counter is always reset to zero when it first starts.

The process of forecasting happens in the block stateful prediction. It can utilize a trained recurrent neural network to forecast. A MAT file can be used to import the trained network. MAT are categorized as data files that include variables, functions, arrays, and other information that belongs to the MATLAB preferences. Each prediction in this block modifies the network's state. The output of this block, which is the expected data or forecasted (YPredict), serves as the input for the last block. In the last block, the predicted output is compared with the test data to evaluate the process accuracy.

The MTSAF model processes are different from the previous one since training data is used instead of test data. In addition, it has an updating loop signal. To avoid an algebraic loop in the model, a memory block is added between the two ends of the prediction block. For multiple sequence problems, it can keep all output sequences unchanged and plot the results or evaluate a statistical value for these outputs such as their average, minimum, or maximum. As can be observed in Fig. 4.10b, the current output is fed to the input of the prediction block to be used for forecasting the next time step of load demand. By making this loop, we are replacing the for-loop command required by MATLAB to achieve this task pragmatically. It is more beneficial to see visually how the algorithm works by showing the main steps in blocks connected to each other. To move the input from the first-time-step value acquired from the test data to the next values derived from the prediction, a switch and clock blocks are added. When the initial input meets the selected criterion, switch block is used to pass through the predicted output that serves as input for the upcoming step. Data transfer between different rates and tasks is managed by rate transition blocks. Comparing the outcomes (Ypredict) to the test data (YTest) from the scope block is the final step for the MTSAF model.

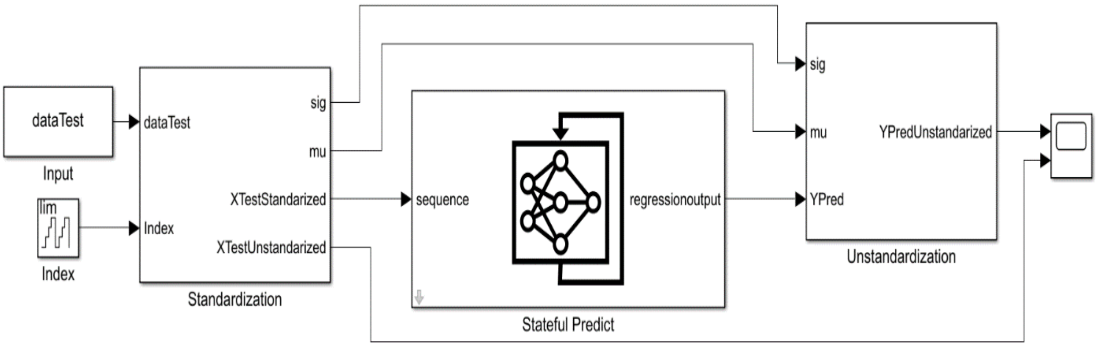


Figure. 4.10a Simulink program for the OTSAF model developed for this study.

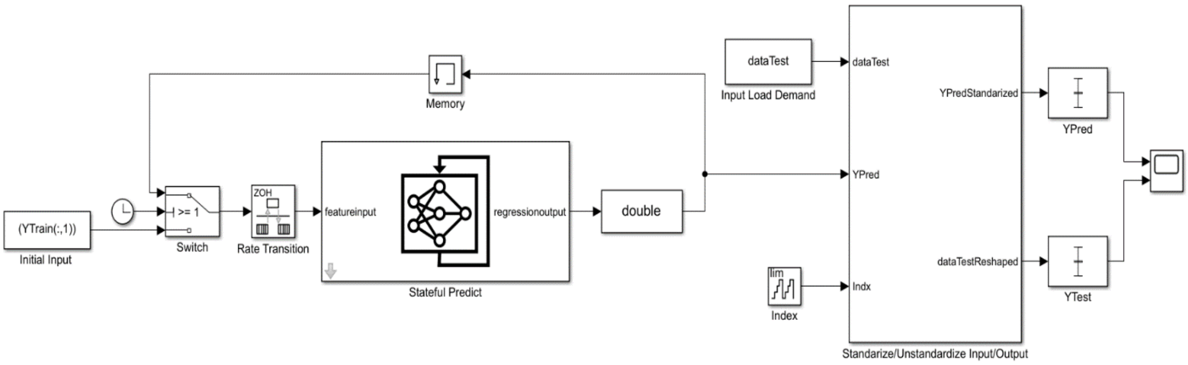


Figure. 4.10b Simulink program for the MTSAF model developed for this study.

4.6 Discussion

In this section, the different models discussed so far are compared to their errors and training times. The OTSAF model requires less time for training the network compared to the same network using MTSAF approach owing to its forecasting time window. The time ratio is 361/3083 (seconds) which is around 12%. Error ratio is also different for these methods with a percentage of 3.12% for OTSAF and 8.01% for MTSAF. However, when the data is smoothed, and sampling ratio is changed from one-day to one-week per prediction, additional improvements in training time and model error of OTSAF are obtained which are found to be 165s (it was 361s) and 0.37% (it was 3.12%). Next, we compare the classical method of data inputting as one sequence and the proposed data augmentation technique. The main difference is in the training time where the proposed model requires only 15% (484/3083) of the training time of the classical model. The error is also improved, which is around 8 % (8.01/7.40). Table 4.2 provides more information and comparisons between the case results.

Table 4.2 Comparison among models

Cases	OTSAF	MTSAF	Multi-Var Single-Seq	Single-Var Multi-Seq	Multi-Var Multi-Seq	Weekly
Inputs	Load	Load	Load, Weather, Weekday, Weekend	Load	Load Weather, Weekday, Weekend	Load

Structure	LSTM(128)
	LSTM(128)
	LSTM(128)
		LSTM(128)
		LSTM(64)
		GRU(32)
		GRU(16)
			LSTM(256)
			LSTM(128)
			LSTM(64)
				LSTM(128)
				LSTM(64)
				GRU(32)
				GRU(16)
					LSTM(256)
					LSTM(128)
					LSTM(64)
						LSTM(128)
						LSTM(128)
						LSTM(128)

Forecast Duration	daily	365_day	365_day	365_day	365_day	Weekly
RMSE/MW **Percentage**	84.36 3.12%	216.08 8.01%	206.55 7.66%	199.42 7.40%	198.78 7.37%	10.11 0.37
Time/minutes	6	51	153	8	66	3

Another significant improvement in the model is that the previous model in the literature with the one-year data division fails to accurately predict the 365-day ahead demand for this dataset. For the multi-variable models, the proposed data augmentation improves the accuracy with 4% less error (7.66/7.37) and accelerates the learning process 232% (3964/9207) times faster than the classical inputting with one sequence per a variable.

CHAPTER FIVE: CONCLUSION AND FUTURE WORKS

5.1 Conclusion

This study presented model of machine learning and specifically deep learning in forecasting load demand for the Kurdistan Region. While the previous regional-related studies employed some statistical methods, this study is the first of its kind to apply a better approach to forecasting. Deep learning has been proven to be more powerful than statistical in the aspects of accuracy and training time speed. For time series and sequence-based challenges, deep learning networks are applied with state-of-the-art LSTM and GRU algorithms.

Forecasting load is a hard to process, especially when dealing with nonlinearity and external influences. Our empirical analysis focused on applying deep learning techniques to multi-sequence input variables. Multiple input sequences are employed to increase the generality of the model including load demands, temperature data, and essential calendrical data such as weekday and weekend information. While the literature uses mainly MATLAB coding for forecasting load demands, this study introduces MATLAB and Simulink programs to present the algorithm in a visualized way. Simulink has a layout similar to a block diagram, making it simple to read. In both OTSAF and MTSAF scenarios Simulink was employed, and the output from the MATLAB and Simulink programs are similar. The test data employed in this study is the load profile for Kurdistan regional power system. The load data from 2015 to 2020 was obtained from the Erbil control center.

In the Kurdistan Region, when the weather is typically cold, power consumption is found to be higher. People need to use more electricity throughout the winter, which peaks around December and January. Also, the relationship between observations in the input data is conducted using correlation analysis which showed a high correlation value among the time series observations. While the previous data augmentation approach was unsuccessful in training the network for several cases, the proposed method demonstrates its ability to forecast the subsequent 365-day load demands in a comparatively short training time and with better accuracy.

Additionally, using smoothed data to evaluate the suggested model at different timescales, such as weekly, it was found to have lower error. It also indicated that an OTSAF forecasting structure outperforms an MTSAF forecasting structure in terms of accuracy. This is to be anticipated because OTSAF forecasts one day ahead of time, whereas MTSAF forecasts 365 days ahead of time, so it has a longer sequence of data. We obtain the result more rapidly when multiple sequences are utilized instead of a single sequence for a variable.

5.2 Suggestion for Future Study

The work that was done in this study can be further enhanced in the future in the following ways:

- Getting involved with smart meter data, as it is more accurate. This data was not available for us to include in our model.
- Forecast the load for Kurdistan as a whole rather than one area. This process requires load and weather data from all the provinces in the region which was not possible for us to add it to the study, and it was out of the scope of this thesis.
- For quick and accurate load forecasting, parallel computations are particularly beneficial for creating speedy simulations with a larger number of processors.

REFERENCES

AGRAWAL, R. K., MUCHAHARY, F. & TRIPATHI, M. M.2018. Long term load forecasting with hourly predictions based on long-short-term-memory networks. IEEE Texas Power and Energy Conference (TPEC), IEEE, 1-6.

AKOGLU, H. 2018. User's guide to correlation coefficients. *Turkish journal of emergency medicine,* 18**,** 91-93.

AL MAMUN, A., SOHEL, M., MOHAMMAD, N., SUNNY, M. S. H., DIPTA, D. R. & HOSSAIN, E. 2020. A comprehensive review of the load forecasting techniques using single and hybrid predictive models. *IEEE Access,* 8**,** 134911-134939.

ALHUSSEIN, M., AURANGZEB, K. & HAIDER, S. I. 2020. Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access,* 8**,** 180544-180557.

ALI, W. H. 2020. Midterm Load Forecasting Analysis For Erbil Governorate Based On Predictive Models. *Zanco Journal of Pure and Applied Sciences,* 32**,** 20-29.

ALI, W. H., HUSSEIN, M. A. & MINA, L. M. 2020. Monthly Peak-load Demand Forecasting for Sulaimani Governorate Using Different Weather Conditions Based on ANN Model. *Sulaimania Journal for Engineering Sciences,* 7.

ASKARI, M. & KEYNIA, F. 2020. Mid-term electricity load forecasting by a new composite method based on optimal learning MLP algorithm. *IET Generation, Transmission & Distribution,* 14**,** 845-852.

BENDAOUD, N. M. M., FARAH, N. & AHMED, S. B. 2021. Comparing Generative Adversarial Networks architectures for electricity demand forecasting. *Energy and Buildings,* 247**,** 111152.

BOUKTIF, S., FIAZ, A., OUNI, A. & SERHANI, M. A. 2018. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies,* 11**,** 1636.

BOUKTIF, S., FIAZ, A., OUNI, A. & SERHANI, M. A. 2019. Single and multi-sequence deep learning models for short and medium term electric load forecasting. *Energies,* 12**,** 149.

BROWNLEE, J. 2017. *Long short-term memory networks with python: develop sequence prediction models with deep learning*, Machine Learning Mastery.

CANKURT, S. & YASIN, M. 2018. Modelling Energy Demand Forecasting Using Neural Networks with Univariate Time Series. *Eurasian Journal of Science and Engineering,* 4**,** 134-140.

CHEN, J., JING, H., CHANG, Y. & LIU, Q. 2019. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process. *Reliability Engineering & System Safety,* 185**,** 372-382.

DARGAN, S., KUMAR, M., AYYAGARI, M. R. & KUMAR, G. 2019. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering***,** 1-22.

DASH, S. K., ROCCOTELLI, M., KHANSAMA, R. R., FANTI, M. P. & MANGINI, A. M. 2021. Long Term Household Electricity Demand Forecasting Based on RNN-GBRT Model and a Novel Energy Theft Detection Method. *Applied Sciences,* 11**,** 8612.

DONG, M. & GRUMBACH, L. 2019. A hybrid distribution feeder long-term load forecasting method based on sequence prediction. *IEEE Transactions on Smart Grid,* 11**,** 470-482.

DONG, Y., MA, X. & FU, T. 2021. Electrical load forecasting: A deep learning approach based on K-nearest neighbors. *Applied Soft Computing,* 99**,** 106900.

DUDEK, G. 2016. Neural networks for pattern-based short-term load forecasting: A comparative study. *Neurocomputing,* 205**,** 64-74.

DUDEK, G., PEŁKA, P. & SMYL, S. 2021. A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting. *IEEE Transactions on Neural Networks and Learning Systems*.

ESKANDARI, H., IMANI, M. & MOGHADDAM, M. P. 2021. Convolutional and recurrent neural network based model for short-term load forecasting. *Electric Power Systems Research,* 195**,** 107173.

FARRAG, T. A. & ELATTAR, E. E. 2021. Optimized Deep stacked long short-term memory network for long-term load forecasting. *IEEE Access,* 9**,** 68511-68522.

FARSI, B., AMAYRI, M., BOUGUILA, N. & EICKER, U. 2021. On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach. *IEEE Access,* 9**,** 31191-31212.

GÉRON, A. 2019. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, O'Reilly Media.

GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. 2016. *Deep learning*, MIT press.

HAMAD, Z. & ABDULRAHMAN, I. 2022. Deep learning-based load forecasting considering data reshaping using MATLAB\Simulink. *International Journal of Energy and Environmental Engineering***,** 1-17.

HE, W. 2017. Load forecasting via deep neural networks. *Procedia Computer Science,* 122**,** 308-314.

HOCHREITER, S. & SCHMIDHUBER, J. 1997. Long short-term memory. *Neural computation,* 9**,** 1735-1780.

HUANG, K., HUSSAIN, A., WANG, Q.-F. & ZHANG, R. 2019. *Deep learning: fundamentals, theory and applications*, Springer.

JACOB, M., NEVES, C. & VUKADINOVIĆ GREETHAM, D. 2020. *Forecasting and assessing risk of individual electricity peaks*, Springer Nature.

JACOBY, W. G. 2000. Loess:: a nonparametric, graphical tool for depicting relationships between variables. *Electoral studies,* 19**,** 577-613.

JIANG, Y., HUANG, Q., ZHANG, K., LIN, Z., ZHANG, T., HU, X., LIU, S., JIANG, C., YANG, L. & LIN, Z. 2021. Medium-long term load forecasting method considering industry correlation for power management. *Energy Reports,* 7**,** 1231-1238.

JIAO, R., ZHANG, T., JIANG, Y. & HE, H. 2018. Short-term non-residential load forecasting based on multiple sequences LSTM recurrent neural network. *IEEE Access,* 6**,** 59438-59448.

JIN, X.-B., ZHENG, W.-Z., KONG, J.-L., WANG, X.-Y., BAI, Y.-T., SU, T.-L. & LIN, S. 2021. Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. *Energies,* 14**,** 1596.

KAREEM, Y. & MAJEED, A. R. Monthly Peak-load Demand Forecasting for Sulaimany Governorate Using SARIMA. 2006 IEEE/PES Transmission & Distribution Conference and Exposition: Latin America, 2006. IEEE, 1-5.

KHAN, H., TAN, A., XIAO, Y., SREERAM, V. & IU, H. H. 2013. An implementation of novel CMAC algorithm for very short term load forecasting. *Journal of Ambient Intelligence and Humanized Computing,* 4**,** 673-683.

KHUNTIA, S. R., RUEDA, J. L. & VAN DER MEIJDEN, M. A. 2016. Forecasting the load of electrical power systems in mid-and long-term horizons: A review. *IET Generation, Transmission & Distribution,* 10**,** 3971-3977.

KIM, Y., SON, H.-G. & KIM, S. 2019. Short term electricity load forecasting for institutional buildings. *Energy Reports,* 5**,** 1270-1280.

KUSTER, C., REZGUI, Y. & MOURSHED, M. 2017. Electrical load forecasting models: A critical systematic review. *Sustainable cities and society,* 35**,** 257-270.

KWON, B.-S., PARK, R.-J. & SONG, K.-B. 2020. Short-term load forecasting based on deep neural networks using LSTM layer. *Journal of Electrical Engineering & Technology,* 15**,** 1501-1509.

LAI, C. S., MO, Z., WANG, T., YUAN, H., NG, W. W. & LAI, L. L. 2020. Load forecasting based on deep neural network and historical data augmentation. *IET Generation, Transmission & Distribution,* 14**,** 5927-5934.

LIANG, Y., ZHI, L. & HAIWEI, Y. 2022. Medium-term Load Forecasting Method with Improved Deep Belief Network for Renewable Energy. *Distributed Generation & Alternative Energy Journal***,** 485–500-485–500.

LIU, D., ZENG, L., LI, C., MA, K., CHEN, Y. & CAO, Y. 2016. A distributed short-term load forecasting method based on local weather information. *IEEE Systems journal,* 12**,** 208-215.

LÓPEZ, M., SANS, C., VALERO, S. & SENABRE, C. 2018. Empirical comparison of neural network and auto-regressive models in short-term load forecasting. *Energies,* 11**,** 2080.

LUSIS, P., KHALILPOUR, K. R., ANDREW, L. & LIEBMAN, A. 2017. Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Applied energy,* 205**,** 654-669.

MA, W., FANG, S., LIU, G. & ZHOU, R. 2017. Modeling of district load forecasting for distributed energy system. *Applied Energy,* 204**,** 181-205.

MANASWI, N. K. 2018. *Deep learning with applications using python: chatbots and face, object, and speech recognition with tensorflow and keras*, Apress.

MARTÍNEZ-ÁLVAREZ, F., TRONCOSO, A., ASENCIO-CORTÉS, G. & RIQUELME, J. C. 2015. A survey on data mining techniques applied to electricity-related time series forecasting. *Energies,* 8**,** 13162-13193.

MATLAB 18/6/2022 Retrieved. corrplot function. 2021b ed.

MELHUM, A. I., OMAR, L. & MAHMOOD, S. A. 2013. Short term load forecasting using artificial neural network. *International Journal of Soft Computing and Engineering (IJSCE),* 3**,** 56-58.

MOHAMMED, N. A. 2018. Modelling of unsuppressed electrical demand forecasting in Iraq for long term. *Energy,* 162**,** 354-363.

MONTGOMERY, D. C., JENNINGS, C. L. & KULAHCI, M. 2015. *Introduction to time series analysis and forecasting*, John Wiley & Sons.

MUHAMMED, A. A. 2011. Short-Term Electricity Load Forecasting of Kurdistan-Erbil Power System Using Artificial Neural Network. *The 4th International Scientific Conference of Salahaddin University-Erbil, October 18-20, 2011*.

NIELSEN, M. A. 2015. *Neural networks and deep learning*, Determination press San Francisco, CA, USA.

RAFI, S. H., DEEBA, S. R. & HOSSAIN, E. 2021. A short-term load forecasting method using integrated CNN and LSTM network. *IEEE Access,* 9**,** 32436-32448.

RAJAGUKGUK, R. A., RAMADHAN, R. A. & LEE, H.-J. 2020. A review on deep learning models for forecasting time series data of solar irradiance and photovoltaic power. *Energies,* 13**,** 6623.

RANA, M. & RAHMAN, A. 2020. Multiple steps ahead solar photovoltaic power forecasting based on univariate machine learning models and data re-sampling. *Sustainable Energy, Grids and Networks,* 21**,** 100286.

RASOOL, A., FTTAH, A. & SADIK, I. 2009. Short Term Load Forecasting for ErbilDistribution System Using ANN and WaveletTransform. *International Journal of Computer and Electrical Engineering,* 1**,** 334.

REDDY, S. S. 2018. Bat algorithm-based back propagation approach for short-term load forecasting considering weather factors. *Electrical Engineering,* 100**,** 1297-1303.

SAJJAD, M., KHAN, Z. A., ULLAH, A., HUSSAIN, T., ULLAH, W., LEE, M. Y. & BAIK, S. W. 2020. A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. *Ieee Access,* 8**,** 143759-143768.

SEHOVAC, L. & GROLINGER, K. 2020. Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention. *IEEE Access,* 8**,** 36411-36426.

SHANG, C., GAO, J., LIU, H. & LIU, F. 2021. Short-Term Load Forecasting Based on PSO-KFCM Daily Load Curve Clustering and CNN-LSTM Model. *IEEE Access,* 9**,** 50344-50357.

SHAO, X., PU, C., ZHANG, Y. & KIM, C. S. 2020. Domain fusion CNN-LSTM for short-term power consumption forecasting. *IEEE Access,* 8**,** 188352-188362.

SOLIMAN, S. A.-H. & AL-KANDARI, A. M. 2010. *Electrical load forecasting: modeling and model construction,* Elsevier.

SOLYALI, D. 2020. A comparative analysis of machine learning approaches for short-/long-term electricity load forecasting in Cyprus. *Sustainability,* 12**,** 3612.

TAHERIFARD, E. 2019. Load and Demand Forecasting in Iraqi Kurdistan using Time series modelling.

TALAAT, M., FARAHAT, M., MANSOUR, N. & HATATA, A. 2020. Load forecasting based on grasshopper optimization and a multilayer feed-forward neural network using regressive approach. *Energy,* 196**,** 117087.

TAN, M., YUAN, S., LI, S., SU, Y., LI, H. & HE, F. 2019. Ultra-short-term industrial power demand forecasting using LSTM based hybrid ensemble learning. *IEEE transactions on power systems,* 35**,** 2937-2948.

TANG, X., DAI, Y., WANG, T. & CHEN, Y. 2019. Short-term power load forecasting based on multi-layer bidirectional recurrent neural network. *IET Generation, Transmission & Distribution*, 13**,** 3847-3854.

TUDOSE, A. M., PICIOROAGA, I. I., SIDEA, D. O., BULAC, C. & BOICEA, V. A. 2021. Short-term load forecasting using convolutional neural networks in COVID-19 context: The romanian case study. *Energies,* 14**,** 4046.

WANG, Y., ZHANG, N. & CHEN, X. 2021. A Short-Term Residential Load Forecasting Model Based on LSTM Recurrent Neural Network Considering Weather Features. *Energies,* 14**,** 2737.

WANI, M. A., BHAT, F. A., AFZAL, S. & KHAN, A. I. 2020. *Advances in deep learning*, Springer.

WEI, N., LI, C., PENG, X., ZENG, F. & LU, X. 2019. Conventional models and artificial intelligence-based models for energy consumption forecasting: A review. *Journal of Petroleum Science and Engineering,* 181**,** 106187.

WU, W., LIAO, W., MIAO, J. & DU, G. 2019. Using gated recurrent unit network to forecast short-term load considering impact of electricity price. *Energy Procedia,* 158**,** 3369-3374.

XIE, J., HONG, T. & STROUD, J. 2015. Long-term retail energy forecasting with consideration of residential customer attrition. *IEEE transactions on smart grid,* 6**,** 2245-2252.

ZHAO, X., GAO, W., QIAN, F. & GE, J. 2021. Electricity cost comparison of dynamic pricing model based on load forecasting in home energy management system. *Energy,* 229**,** 120538.

International Journal of Energy and Environmental Engineering (2022) 13:853–869
https://doi.org/10.1007/s40095-022-00480-x

ORIGINAL RESEARCH

Deep learning-based load forecasting considering data reshaping using MATLAB\Simulink

Zhalla Hamad[1] · Ismael Abdulrahman[1]

Received: 9 December 2021 / Accepted: 21 January 2022 / Published online: 16 February 2022

Abstract

Load forecasting is a nonlinear problem and complex task that plays a key role in power system planning, operation, and control. A recent study proposed a deep learning approach called historical data augmentation (HDA) to improve the accuracy of the load forecasting model by dividing the input data into several yearly sub-datasets. When the original data is associated with high time step changes from 1 year to another, the approach was not found as effective as it should be for long-term forecasting because the time-series information is disconnected by the approach between the end of 1-year sub-data and the beginning of the next-year sub-data. Alternatively, this paper proposes the use of 2-year sub-dataset in order to connect the two ends of the yearly subsets. A correlation analysis is conducted to show how the yearly datasets are correlated to each other. In addition, a Simulink-based program is introduced to simulate the problem which has an advantage of visualizing the algorithm. To increase the model generalization, several inputs are considered in the model including load demand profile, weather information, and some important categorical data such as week-day and weekend data that are embedded using one-hot encoding technique. The deep learning methods used in this study are the long short-term memory (LSTM) and gated rest unit (GRU) neural networks which have been increasingly employed in the recent years for time series and sequence problems. To provide a theoretical background on these models, a new picturized detail is presented. The proposed method is applied to the Kurdistan regional load demands and compared with classical methods of data inputting demonstrating improvements in both the model accuracy and training time.

Keywords Load forecasting · Deep learning · LSTM · GRU · MATLAB · Simulink · Kurdistan region

Introduction

Load forecasting is a method to predict future load demands by analyzing historical data and finding dependency patterns of its time-step observations. It has many applications in power system operation and planning including demand response, scheduling, unit commitment, energy trading, system planning, and energy policy [1]. Accurate load forecasting helps power companies and decision-makers to make a balance between supply and demand, prevent power interruptions due to load shedding, and avoid excess reserve of power generation. Load forecasting problem is a challenging task due to its complexity, uncertainty, and variety of factors affecting the prediction. It is considered as a type of time-series problems that needs a special solution. Depending on its application, load forecasting can be classified into: very–short load forecasting (VSTLF), short–term load forecasting (STLF), medium–term load forecasting (MTLF), and long–term load forecasting (LTLF). VSTLF is used in the problems of demand response and real-time operation that require a time horizon of a few minutes to several hours ahead. Forecasting the load demand from one day to several days ahead is called STLF, whereas forecasting from 1 week to several weeks ahead is known as MTLF. These two types of forecasting cover the majority of load-forecasting studies in the literature and are mainly used in scheduling, unit commitment, and energy marketing. Lastly, LTLF refers to the forecasting with a time frame of up to serval years ahead and it is useful for planning and energy-trading purposes. [1, 2].

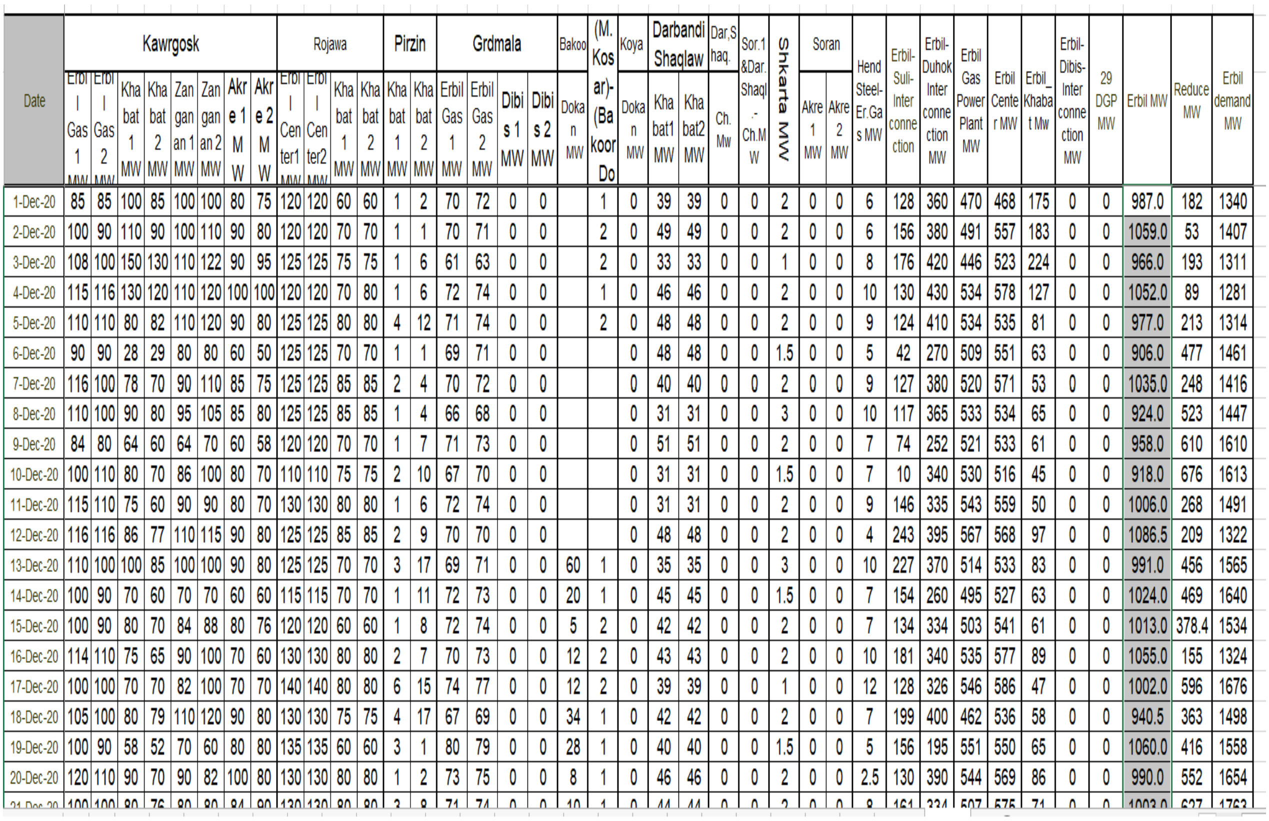
✉ Ismael Abdulrahman
ismael.abdulrahman@epu.edu.iq

Zhalla Hamad
zhalla.mei20@epu.edu.iq

[1] Department of Information System Engineering Techniques, Erbil Technical Engineering College, Erbil Polytechnic University, Erbil, Kurdistan region 44001, Iraq

Springer

Erbil_Load Dataset [year 2018,month December]



Date	Kawrgosk								Rojawa				Pirzin		Grdmala				Bakoo	(M.Kosar)-(Bakoor Do	Koya	Darbandi Shaqlaw		Dar,Shaq.	Sor.1 &Dar.Shaql.-Ch.MW	Shkarta MW	Soran		Hend Steel-Er.Gas MW	Erbil-Suli-Interconnection	Erbil-Duhok Interconnection MW	Erbil Gas Power Plant MW	Erbil Center MW	Erbil_Khabat Mw	Erbil-Dibis-Interconnection MW	29 DGP MW	Erbil MW	Reduce MW	Erbil demand MW
	Erbil Gas 1 MW	Erbil Gas 2 MW	Khabat 1 MW	Khabat 2 MW	Zangan 1 MW	Zangan 2 MW	Akre 1 MW	Akre 2 MW	Erbil Center1 MW	Erbil Center2 MW	Khabat 1 MW	Khabat 2 MW	Khabat 1 MW	Khabat 2 MW	Erbil Gas 1 MW	Erbil Gas 2 MW	Dibis 1 MW	Dibis 2 MW	Dokan MW		Dokan MW	Khabat1 MW	Khabat2 MW	Ch. Mw			Akre 1 MW	Akre 2 MW											
1-Dec-20	85	85	100	85	100	100	80	75	120	120	60	60	1	2	70	72	0	0		1	0	39	39	0	0	2	0	0	6	128	360	470	468	175	0	0	987.0	182	1340
2-Dec-20	100	90	110	90	100	110	90	80	120	120	70	70	1	1	70	71	0	0		2	0	49	49	0	0	2	0	0	6	156	380	491	557	183	0	0	1059.0	53	1407
3-Dec-20	108	100	150	130	110	122	90	95	125	125	75	75	1	6	61	63	0	0		2	0	33	33	0	0	1	0	0	8	176	420	446	523	224	0	0	966.0	193	1311
4-Dec-20	115	116	130	120	110	120	100	100	120	120	70	80	1	6	72	74	0	0		1	0	46	46	0	0	2	0	0	10	130	430	534	578	127	0	0	1052.0	89	1281
5-Dec-20	110	110	80	82	110	120	90	80	125	125	80	80	4	12	71	74	0	0		2	0	48	48	0	0	2	0	0	9	124	410	534	535	81	0	0	977.0	213	1314
6-Dec-20	90	90	28	29	80	80	60	50	125	125	70	70	1	1	69	71	0	0			0	48	48	0	0	1.5	0	0	5	42	270	509	551	63	0	0	906.0	477	1461
7-Dec-20	116	100	78	70	90	110	85	75	125	125	85	85	2	4	70	72	0	0			0	40	40	0	0	2	0	0	9	127	380	520	571	53	0	0	1035.0	248	1416
8-Dec-20	110	100	90	80	95	105	85	80	125	125	85	85	1	4	66	68	0	0			0	31	31	0	0	3	0	0	10	117	365	533	534	65	0	0	924.0	523	1447
9-Dec-20	84	80	64	60	64	70	60	58	120	120	70	70	1	7	71	73	0	0			0	51	51	0	0	2	0	0	7	74	252	521	533	61	0	0	958.0	610	1610
10-Dec-20	100	110	80	70	86	100	80	70	110	110	75	75	2	10	67	70	0	0			0	31	31	0	0	1.5	0	0	7	10	340	530	516	45	0	0	918.0	676	1613
11-Dec-20	115	110	75	60	90	90	80	70	130	130	80	80	1	6	72	74	0	0			0	31	31	0	0	2	0	0	9	146	335	543	559	50	0	0	1006.0	268	1491
12-Dec-20	116	116	86	77	110	115	90	80	125	125	85	85	2	9	70	70	0	0			0	48	48	0	0	2	0	0	4	243	395	567	568	97	0	0	1086.5	209	1322
13-Dec-20	110	100	100	85	100	100	90	80	125	125	70	70	3	17	69	71	0	0	60	1	0	35	35	0	0	3	0	0	10	227	370	514	533	83	0	0	991.0	456	1565
14-Dec-20	100	90	70	60	70	70	60	60	115	115	70	70	1	11	72	73	0	0	20	1	0	45	45	0	0	1.5	0	0	7	154	260	495	527	63	0	0	1024.0	469	1640
15-Dec-20	100	90	80	70	84	88	80	76	120	120	60	60	1	8	72	74	0	0	5	2	0	42	42	0	0	2	0	0	7	134	334	503	541	61	0	0	1013.0	378.4	1534
16-Dec-20	114	110	75	65	90	100	70	60	130	130	80	80	2	7	70	73	0	0	12	2	0	43	43	0	0	2	0	0	10	181	340	535	577	89	0	0	1055.0	155	1324
17-Dec-20	100	100	70	70	82	100	70	70	140	140	80	80	6	15	74	77	0	0	12	2	0	39	39	0	0	1	0	0	12	128	326	546	586	47	0	0	1002.0	596	1676
18-Dec-20	105	100	80	79	110	120	90	80	130	130	75	75	4	17	67	69	0	0	34	1	0	42	42	0	0	2	0	0	7	199	400	462	536	58	0	0	940.5	363	1498
19-Dec-20	100	90	58	52	70	60	80	80	135	135	60	60	3	1	80	79	0	0	28	1	0	40	40	0	0	1.5	0	0	5	156	195	551	550	65	0	0	1060.0	416	1558
20-Dec-20	120	110	90	70	90	82	100	80	130	130	80	80	1	2	73	75	0	0	8	1	0	46	46	0	0	2	0	0	2.5	130	390	544	569	86	0	0	990.0	552	1654
21-Dec-20	100	100	80	76	80	80	84	90	130	130	80	80	3	8	71	74	0	0	10	1	0	44	44	0	0	2	0	0	8	161	334	507	575	71	0	0	1003.0	627	1763

Erbil_Load Dataset [year 2018,month December] _ preproccesing

date	Averg_load	max_load	min_load	Averg_demand	max_demand	min_demand
1-Dec-18	993.05	1132.20	810.21	1902.69	2177.26	1288.37
2-Dec-18	996.87	1048.68	944.58	1963.74	2152.49	1548.73
3-Dec-18	973.45	1095.26	899.66	1907.46	2098.90	1434.98
4-Dec-18	998.97	1159.24	855.75	2030.77	2253.35	1552.33
5-Dec-18	955.54	1044.73	855.32	2054.57	2324.39	1501.53
6-Dec-18	947.82	1082.58	793.96	2071.35	2333.26	1541.99
7-Dec-18	961.38	1067.91	662.31	2011.70	2205.39	1615.34
8-Dec-18	1059.79	1133.88	947.35	2108.27	2344.77	1511.72
9-Dec-18	999.73	1124.68	843.40	2182.32	2380.75	1745.16
10-Dec-18	1015.15	1077.25	877.38	2163.21	2364.15	1752.62
11-Dec-18	1073.47	1176.57	969.82	2118.94	2357.52	1702.33
12-Dec-18	1107.25	1192.72	1024.24	2060.51	2274.37	1627.16
13-Dec-18	1019.12	1184.65	745.01	2139.51	2358.26	1705.43
14-Dec-18	1067.65	1148.87	972.20	2102.18	2258.60	1764.08
15-Dec-18	1031.47	1154.20	938.04	2196.51	2372.24	1857.31
16-Dec-18	1023.97	1107.68	946.85	2189.96	2379.47	1926.61
17-Dec-18	927.98	1110.76	732.73	2276.60	2481.58	1987.29
18-Dec-18	996.32	1063.51	891.11	2300.50	2456.98	1981.30
19-Dec-18	972.66	1084.78	869.46	2278.76	2433.45	2022.88
20-Dec-18	962.54	1083.72	846.33	2232.78	2445.26	1928.60
21-Dec-18	998.81	1085.00	909.99	2187.08	2327.80	1913.30
22-Dec-18	995.73	1096.37	873.60	2243.19	2398.79	2040.15
23-Dec-18	1001.91	1090.10	909.14	2264.48	2447.60	2023.90
24-Dec-18	1031.63	1103.49	949.85	2246.10	2475.56	1929.83
25-Dec-18	997.90	1119.92	911.30	2231.08	2408.18	1952.92
26-Dec-18	981.78	1071.54	864.99	2301.63	2455.29	2044.55
27-Dec-18	1008.59	1148.92	845.31	2293.84	2479.48	1944.31
28-Dec-18	981.19	1084.68	763.47	2176.68	2363.98	1822.53
29-Dec-18	1006.31	1120.80	849.90	2257.78	2430.31	1996.25
30-Dec-18	987.19	1115.86	889.01	2311.09	2468.70	2052.78
31-Dec-18	922.11	1086.22	496.92	2282.55	2452.94	2040.95

Erbil_Weather Dataset [year 2018, month December]

ڕۆژ	بەرزترین پلەی گەرمی	نزمترین پلەی گەرمی	تێکرایی پلەی گەرمی	ڕێژەی شێ	تێبینی
1	16.7	12.0	14.4	83.1	
2	18.5	12.0	15.3	92.8	
3	19.6	9.8	14.7	78.8	
4	17.6	9.9	13.8	73.6	
5	16.7	13.5	15.1	79.8	
6	13.7	12.1	12.9	91.0	
7	13.2	10.5	11.9	94.0	
8	11.9	10.2	11.1	89.0	
9	13.8	10.0	11.9	89.6	
10	15.3	7.0	11.2	84.1	
11	17.6	8.0	12.8	74.3	
12	15.7	11.5	13.6	70.4	
13	12.1	11.4	11.8	90.3	
14	14.0	6.6	10.3	77.8	
15	14.1	4.0	9.1	78.3	
16	17.3	5.0	11.2	74.8	
17	10.9	9.5	10.2	92.1	
18	11.1	10.0	10.6	96.1	
19	17.6	8.7	13.2	77.3	
20	13.6	10.4	12.0	85.5	
21	12.1	9.4	10.8	91.8	
22	15.1	5.2	10.2	83.0	
23	15.5	4.5	10.0	85.4	
24	11.7	7.0	9.4	90.8	
25	13.9	5.6	9.8	87.4	
26	15.7	4.6	10.2	60.9	
27	13.4	11.0	12.2	71.9	
28	11.0	9.0	10.0	87.8	
29	11.6	5.0	8.3	78.3	
30	11.4	4.2	7.8	65.1	
31	12.5	8.7	10.6	72.1	
کۆ	14.4	8.6	11.5	82.1	

پوختە

پێشبینیکردنی بار کێشەیەکی ناجێگر و ئەرکێکی ئاڵۆزە، کە ڕۆڵێکی سەرەکی دەگێڕێت لە پلاندانان و کارپێکردن و کۆنترۆڵکردنی سیستەمی کارەبادا. ئەم توێژینەوەیە پێشنیاری باشترکردنی مۆدێلی (Deep Learning) دەکات، کە پێی دەوترێت زیادکردنی داتا مێژووییەکان (HDA) بۆ بەرزکردنەوەی وردی مۆدێلی پێشبینیکردنی بار بە دابەشکردنی ئەو داتایانەی کە هەیە بۆ چەند کۆمەڵە داتایەکی لاوەکی ساڵانە. کاتێک داتا مێژوویە کۆکراوەکان کە پەیوەندییان بە گۆڕانکارییە بەرزەکانی هەنگاوی کاتەوە هەیە لە ساڵێکەوە بۆ ساڵێکی تر، مۆدێلەکە بەم شێوەیە کاریگەر نیە، بۆ پێشبینی درێژخایەن. چونکە زانیارییەکانی زنجیرە کاتییەکان کە دەکەویتە نێوان کۆتایی ساڵەکانەوە پچڕاون. ئەم تێزە پێشنیاری بەکارهێنانی بەستنەوەی دوو ساڵە دەکات بە مەبەستی فێربوونی مۆدێلەکە و بەستنەوە کۆتا داتاکانی نێوان ساڵەکان. بۆ ئەم مەبەستە شیکارییەکی پەیوەندی ئەنجام دەدات، بۆ ئەوەی نیشان بدات کە چۆن کۆمەڵە داتا ساڵانەکان پەیوەندییان بە یەکەوە هەیە.

جگە لەمەش (Simulink) دەخرێتەڕوو بۆ نیشاندانی کێشەکە بەشێوەی بلۆك، کە تێگەیشتن لە مۆدێلەکە ئاسانتر دەکات. هەروەها بۆ زیاتر گشتاندنی مۆدێلەکە، چەندین زانیاری لە مۆدێلەکەدا لەبەرچاو دەگیرێن. لەوانە: (پرۆفایلی داواکاری بار، زانیاری کەشوهەوا، هەندێک داتای گرنگ وەک داتای ڕۆژەکانی هەفتە و کۆتایی هەفتە، کە وەك هۆکاری کاریگەر لەسەر بار هەژمار دەکرێن. شێوازەکانی (Deep Learning) کە لەم لێکۆڵینەوەیەدا بەکارهێنراون بریتین لە (LSTM)و (GRU)، کە لە ساڵانی ڕابردوودا زیاتر بۆ کێشەکانی زنجیرە کاتیی و زنجیرەکان بەکارهێنراون. مۆدێلی پێشنیارکراو بۆ داواکارییەکانی باری لە کوردستان بەکاردەهێنرێت و بەراورد دەکرێت لەگەڵ شێوازە کلاسیکییەکانی داخڵکردنی داتا. چونکە لەم مۆدێلە لە جیاتی یەك زنجیرە بەکاربهێندریت کە شێوازە کلاسیکیەکەیە بۆ داخل کردنی داتا، چەند زنجیرەك بەکاردێنین. دەرئەنجامەکەش بریتی بوو لە: (کەم کردنەوەی کاتی ڕاهێنان، ووردی و دروستی زیاتر لە ئەنجامدا).

حکومەتی هەرێمی کوردستان
سەرۆکایەتی ئەنجومەنی وەزیران
وەزارەتی خوێندنی باڵا و توێژینەوەی زانستی
زانکۆی پۆلیتکنیکی هەولێر
کۆلێژی تەکنیکی ئەندازیاری
بەشی سیستمی زانیاری

پێشبینیکردنی داواکاری بار لە پارێزگای هەولێر لەسەر بنەمایی سیستەمیکی فێرکراو

نامەیەکە
پێشکەشی ئەنجومەنی کۆلیژی تەکنیکی ئەندازیاری کراوە
لە زانکۆی پۆلیتەکنیکی هەولێر وەک بەشێک لە
پێداویستیەکانی بەدەستهێنانی پلەی ماستەر لە تەکنیکی
سیستەمی زانیاری

لەلایەن
ژاڵە جمیل حمد
بەکالۆریۆسی لە ئەندازیاری پرۆگرامسازی - زانکۆی کۆیە - 2011

بە سەرپەرشتی
د. اسماعیل خورشید عبدالرحمن

هەولێر- کوردستان
گەڵارێزان 2722