

PERFORMANCE EVALUATION OF NEW AND ADVANCED NEURAL
NETWORKS FOR SHORT TERM LOAD FORECASTING: CASE STUDIES FOR
MARITIMES AND ONTARIO

by

Syed Talha Mehmood

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
April 2014

© Copyright by Syed Talha Mehmood, 2014

Dedication

I dedicate this thesis to my dearest family and my beloved country Pakistan

Table of Contents

List of Tables	vi
List of Figures	vii
Abstract	xi
List of Abbreviations and Symbols Used	xii
Acknowledgement	xiii
Chapter 1: Introduction.....	1
1.1 Overview	1
1.2 Thesis Objective.....	3
1.3 Structure of the Thesis	5
Chapter 2: Short-Term Load forecasting (STLF).....	6
2.1 A brief description of Power Systems.....	6
2.2 The necessity & importance of STLF	7
2.3 A standardized approach to STLF	9
2.3.1 Data Pre-processing	9
2.3.2 Variable Selection.....	10
2.3.3 Training Frame Selection.....	10
2.3.4 Load Forecasting.....	11
2.3.5 Data Post-processing.....	11
2.4 Literature Review.....	12
2.4.1 Traditional Methods.....	12
2.4.1.1 Time-series models.....	12
2.4.1.2 Regression Models	12
2.4.1.3 State Space Models based Kalman Filter Estimation.....	13
2.4.2 Artificial Intelligence Methods.....	14
2.4.2.1 Knowledge Based Expert System	14
2.4.2.2 Fuzzy Inference System	16
2.4.2.3 Artificial Neural Network (ANN).....	17
2.4.2.4 Support Vector Machine (SVM).....	23

Chapter 3:	Artificial Neural Networks and proposed algorithm	25
3.1	Introduction	25
3.2	Artificial Neuron Model.....	26
3.3	Transfer Functions	27
3.4	Network Architectures	29
3.4.1	Single Layer Feed forward Neural Network.....	29
3.4.2	Multilayer Feed forward Neural Network	30
3.4.3	Cascaded Neural Network	32
3.4.4	Recurrent Neural Network.....	33
3.4.5	Hybrid Neural Network Architectures.....	34
3.5	Network Training.....	36
3.5.1	Error Back Propagation Algorithm.....	37
3.5.2	Newton Algorithm	43
3.5.3	Gauss – Newton Algorithm	45
3.5.4	Levenberg-Marquardt Back Propagation Algorithm.....	46
Chapter 4:	Implementation of ANN for STLF	49
4.1	Statistical Analysis and selection of Input parameters.....	49
4.2	Data Pre-Processing	55
4.3	Training of Neural Networks	56
4.4	Finalised networks used for Forecasting.....	56
4.5	Error Analysis and Performance Evaluation.....	60
4.6	Use of Software.....	60
Chapter 5:	Results and Discussion	61
5.1	Nova Scotia.....	61
5.1.1	24 hours ahead forecast for Weekday.....	61
5.1.2	Comparative Analysis of 24 hours ahead forecast for Weekday	70
5.1.3	Week ahead forecast for Weekdays.....	72
5.1.4	Comparative Analysis of Week ahead forecast for Weekdays.....	88
5.1.5	Load forecast for Weekend.....	91
5.1.6	Comparative Analysis of Load forecast for Weekend.....	97
5.1.7	Summary of Load forecast results for Nova Scotia	99
5.2	New Brunswick.....	100
5.2.1	24 hours ahead forecast for Weekdays	100

5.2.2	Comparative Analysis of 24 hours ahead forecast for Weekdays ..	109
5.2.3	Week ahead forecast for Weekdays	111
5.2.4	Comparative Analysis of Week ahead forecast for Weekdays	127
5.2.5	Load forecast for Weekend	130
5.2.6	Comparative Analysis of Load forecast for Weekend	135
5.2.7	Summary of Load forecast results for Nova Scotia	137
5.3	Ontario	138
5.3.1	24 hours ahead forecast for Weekdays	138
5.3.2	Comparative Analysis of 24 hours ahead forecast for Weekdays ..	147
5.3.3	Week ahead forecast for Weekdays	149
5.3.4	Week ahead forecast for Weekdays	165
5.3.5	Load forecast for Weekdays	168
5.3.6	Comparative Analysis of Load forecast for Weekdays	173
5.3.7	Summary of Load forecast results for Nova Scotia	175
5.4	Summary of Results	176
5.4.1	24 hours ahead forecast for Weekdays	176
5.4.2	Week ahead forecast for Weekdays	178
5.4.3	Weekend forecast	180
Chapter 6:	Conclusion	182
6.1	Summary and Contributions	182
6.2	Suggestions for Future Work	184
References	185
Appendix A:	Pseudo Code	191

List of Tables

Table 1: Transfer Functions	28
Table 2: Summary of selected Inputs for Weekdays and Weekend forecasts	55
Table 3: 24 hours ahead MAPE and RMSE for Nova Scotia	70
Table 4: Week ahead MAPE and RMSE for Nova Scotia	88
Table 5: Weekend forecast MAPE and RMSE for Nova Scotia	97
Table 6: Summary of Load forecast results for Nova Scotia	99
Table 7: 24 hours ahead MAPE and RMSE for New Brunswick	109
Table 8: Week ahead MAPE and RMSE for New Brunswick	127
Table 9: Weekend forecast MAPE and RMSE for New Brunswick	135
Table 10: Summary of Load forecast results for New Brunswick	137
Table 11: 24 hours ahead MAPE and RMSE for Ontario	147
Table 12: Week ahead MAPE and RMSE for Ontario	165
Table 13: Weekend forecast MAPE and RMSE for New Brunswick	173
Table 14: Summary of Load forecast results for Ontario	175
Table 15: Summary of 24 hours ahead forecast for Weekdays	176
Table 16: Summary of Week ahead forecast for Weekdays	178
Table 17: Summary of Weekend forecast	180

List of Figures

Figure 1: A generalized structure of Power System [3]	6
Figure 2: A single line diagram of electric Power System [1]	7
Figure 3: Standardized structure of STLF model [28]	9
Figure 4: Biological Neuron [13]	25
Figure 5: Biological Neuron Model [58]	26
Figure 6: Artificial Neuron Model [59]	27
Figure 7: Single Layer Perceptron Network [59]	29
Figure 8: Multi-layer Perceptron Network [59]	30
Figure 9: 3-layered Cascaded Neural Network	32
Figure 10: Recurrent Neural Network with Local feedback	33
Figure 11: Recurrent Neural Network with Global feedback	33
Figure 12: Cascaded Recurrent Neural Network with Local feedback	35
Figure 13: Cascaded Recurrent Neural Network with Global feedback	35
Figure 14: Local and Global Minima [63]	42
Figure 15: Effect of learning rate coefficient on learning curve [62]	42
Figure 16: Flowchart of LMBP algorithm [62]	48
Figure 17: Daily Load Curves for Nova Scotia	50
Figure 18: Daily Load Curves for New Brunswick	50
Figure 19: Daily Load Curves for New Brunswick	50
Figure 20: Weekly Load Curves for Nova Scotia	51
Figure 21: Weekly Load Curves for New Brunswick	52
Figure 22: Weekly Load Curves for Ontario	52
Figure 23: Weekly load and temperature (above), Daily temperature (below) for NS	53
Figure 24: Weekly load and temperature (above), Daily temperature (below) for NB	54
Figure 25: Weekly load and temperature (above), Daily temperature (below) for ON	54
Figure 26: Feed forward Neural Network (FFNN)	57
Figure 27: Cascaded Neural Network (CNN)	57
Figure 28: Recurrent Neural Network with Local feedback (RNNL)	58
Figure 29: Recurrent Neural Network with Global feedback (RNNG)	58

Figure 30: Cascaded – Recurrent Neural Network with Local feedback (CRNNL)	59
Figure 31: Cascaded – Recurrent Neural Network with Global feedback (CRNNG)	59
Figure 32: 3-layered feed forward Neural Network (FFNN (3L))	59
Figure 33: 3-layered cascaded Neural Network (CNN (3L))	60
Figure 34: Day ahead forecast results for FFNN	62
Figure 35: Day ahead forecast results for CNN	63
Figure 36: Day ahead forecast results for RNNL	64
Figure 37: Day ahead forecast results for RNNG	65
Figure 38: Day ahead forecast results for CRNNL	66
Figure 39: Day ahead forecast results for CRNNG	67
Figure 40: Day ahead forecast results for 3-layered FFNN	68
Figure 41: Day ahead forecast results for 3-layered CNN	69
Figure 42: 24 hours ahead forecasting error performance comparison between FFNN and CNN (3L) for Nova Scotia	71
Figure 43: Week ahead forecast results for FFNN	73
Figure 44: Week ahead forecast results for CNN	74
Figure 45: Week ahead forecast results for RNNL	76
Figure 46: Week ahead forecast results for RNNG	78
Figure 47: Week ahead forecast results for CRNNL	80
Figure 48: Week ahead forecast results for CRNNG	82
Figure 49: Week ahead forecast results for 3-layered FFNN	84
Figure 50: Week ahead forecast results for 3-layered CNN	86
Figure 51: Week ahead forecasting error performance of comparison between FFNN and CNN (3L) for Nova Scotia	90
Figure 52: Weekend forecast results (FFNN, CN, RNNL, RNNG)	93
Figure 53: Weekend forecast results (CRNNL, CRNG, FFNN (3L), CNN (3L))	95
Figure 54: Weekend forecasting error performance comparison between RNNG and FFNN (3L) for Nova Scotia	98
Figure 55: Graphical presentation of load forecast results for Nova Scotia	99
Figure 56: Day ahead forecast results for FFNN	101
Figure 57: Day ahead forecast results for CNN	102

Figure 58: Day ahead forecast results for RNNL	103
Figure 59: Day ahead forecast results for RNNG	104
Figure 60: Day ahead forecast results for CRNNL	105
Figure 61: Day ahead forecast results for CRNNG	106
Figure 62: Day ahead forecast results for 3-layered FFNN	107
Figure 63: Day ahead forecast results for 3-layered CNN	108
Figure 64: 24 hour ahead forecasting error performance comparison between FFNN and CNN (3L) for New Brunswick	110
Figure 65: Week ahead forecast results for FFNN	112
Figure 66: Week ahead forecast results for CNN	113
Figure 67: Week ahead forecast results for RNNL	115
Figure 68: Week ahead forecast results for RNNG	117
Figure 69: Week ahead forecast results for CRNNL	119
Figure 70: Week ahead forecast results for CRNNG	121
Figure 71: Week ahead forecast results for 3-layered FFNN	123
Figure 72: Week ahead forecast results for 3-layered CNN	125
Figure 73: Week ahead forecasting error performance comparison between FFNN and CNN for New Brunswick	129
Figure 74: Weekend forecast results (FFNN, CNN, RNNL and RNNG)	131
Figure 75: Weekend forecast results: (CRNNL, CRNNG, FFNN (3L) and CNN (3L))	134
Figure 76: Weekend forecasting error performance comparison between RNNG and CRNNL for New Brunswick	136
Figure 77: Graphical presentation of load forecast results for New Brunswick	137
Figure 78: Day ahead forecast results for FFNN	139
Figure 79: Day ahead forecast results for CNN	140
Figure 80: Day ahead forecast results for RNNL	141
Figure 81: Day ahead forecast results for RNNG	142
Figure 82: Day ahead forecast results for CRNNL	143
Figure 83: Day ahead forecast results for CRNNG	144
Figure 84: Day ahead forecast results for 3-layered FFNN	145
Figure 85: Day ahead forecast results for 3-layered CNN	146

Figure 86: 24 hours ahead forecasting error performance comparison between CRNNG and CNN (3L) for Ontario	148
Figure 87: Week ahead forecast results for FFNN	150
Figure 88: Week ahead forecast results for CNN	151
Figure 89: Week ahead forecast results for RNNL	153
Figure 90: Week ahead forecast results for RNNG	155
Figure 91: Week ahead forecast results for CRNNL	157
Figure 92: Week ahead forecast results for CRNNG	159
Figure 93: Week ahead forecast results for 3-layered FFNN	161
Figure 94: Week ahead forecast results for 3-layered CNN	163
Figure 95: Week ahead forecasting error performance comparison between FFNN and FFNN (3L) for Ontario	167
Figure 96: Weekend forecast results (FFNN, CNN, RNNL and RNNG)	169
Figure 97: Weekend forecast results (CRNNL, CRNNG, FFNN (3L) and CNN (3L))	171
Figure 98: Weekend forecasting error performance comparison between RNNL and FFNN (3L) for Ontario	174
Figure 99: Graphical presentation of load forecast results for Ontario	175
Figure 100: Graphical summary of Day ahead forecast results	177
Figure 101: Graphical summary of Week ahead forecast results	179
Figure 102: Graphical summary of Weekend forecast results	181

Abstract

Electric power systems are huge real time energy distribution networks where accurate short term load forecasting (STLF) plays an essential role. This thesis is an effort to comprehensively investigate new and advanced neural network (NN) architectures to perform STLF. Two hybrid and two 3-layered NN architectures are introduced. Each network is individually tested to generate weekday and weekend forecasts using data from three jurisdictions of Canada.

Overall findings suggest that 3-layered cascaded NN have outperformed almost all others for weekday forecasts. For weekend forecasts 3-layered feed forward NN produced most accurate results. Recurrent and hybrid networks performed well during peak hours but due to occurrence of constant high error spikes were not able to achieve high accuracy.

List of Abbreviations and Symbols Used

STLF	Short Term Load Forecasting
EMS	Energy Management System
ANN	Artificial Neural Networks
FFNN	Feed forward Neural Network
CNN	Cascaded Neural Network
RNNL	Recurrent Neural Network with Local feedback
RNNG	Recurrent Neural Network with Global feedback
CRNNL	Cascaded – Recurrent Neural Network with Local feedback
CRNNG	Cascaded – Recurrent Neural Network with Global feedback
EBP	Error Back Propagation
LMBP	Levenberg-Marquardt Back Propagation
Δw	Weight correction
η	Learning coefficient
δ	Gradient
μ	Combination coefficient

Acknowledgement

First and foremost I thank Almighty Allah, the most powerful, the most gracious for blessing me with resources and opportunity to fulfill my ambition. I would also like to acknowledge my supervisor Dr. M. El-Hawary who guided me wisely and proved to be an inspiration throughout the course of this work. I am also very thankful to the supervisory committee members Dr. T. Little and Dr. W. Phillips for reviewing my work.

A special thanks to my most beloved parents who encouraged me and has been a constant source of strength for me to finish this work. They kept me motivated while taking care of everything back home. I would also like to praise my wife for being patient and caring. Due to her support I was able to focus all my energies to advance in my work. I would also like to admire my siblings who have always cheered me up in tough situations.

I thank FGS and the whole Department of Electrical and Computer Engineering, Dalhousie University for providing me financial support and environment in which I could express my thoughts to the best of my imagination. And I finally thank the people of Halifax, Nova Scotia for making me feel like home for the whole time period.

Chapter 1: Introduction

1.1 Overview

The electric power system is one of the biggest real time energy distribution systems where the generation, transmission and distribution of electrical energy take place simultaneously. With negligible storage of energy in the system, operation of an electric power system is a challenging task. The system response is dictated by on-line load requirements, and to strike a consistent balance between supply and demand of electric power is the main task of power systems operation management also known as Energy Management System. In order to achieve this task, an adequate amount of generation resources should be available at all times. The optimal allocation of these resources requires an accurate forecast of future load demands in order to ensure smooth operation and decrease generation cost [1 – 3]. All involved entities in a power system perform load forecasting on a continuous basis in order to operate efficiently in today's deregulated environment. Whether be Generation companies, Transmission companies, Independent Power Producers, Independent System Operators or Regional Transmission Organizations, all base their planning, negotiations, and operations on the basis of load forecasting results achieved by adopting one of load forecasting methods available.

Reduction in forecasting accuracy results in uneconomic operation. Hodge in [4] studied the implications of both positive and negative load forecasting error and concluded that errors on either side may result in high operating costs. For example one of the fallouts of underestimating load, is limited electric supply which may lead to reduction in energy quality which might prove costly. On the other hand an overestimation causes excessive use of precious resources and results in expensive

generation [5]. This relationship between forecasting error and operating costs quantified by Hobbs in [6] showed that a 1% reduction in forecasting error of 10 GW utility results in an annual savings of US \$ 1.6 million. Hipert in [7] says that “The costs of the error are so high that research that could help reducing it in a few percent points would be amply justified”.

With respect to time horizons, load forecasting can be classified into three types: Long term, Medium term and Short term load forecasting. The Long Term Load Forecasting (LTLF) consists of forecasts from one to several years and has applications in long term planning of power systems. The Medium term load forecasting (MTLF) predicts load weekly, monthly and yearly for the mid-term planning tasks such as maintenance, fuel supply planning and generation scheduling of a power plant. The Short term load forecasting (STLF) on which this thesis is focused, predicts load demand for one hour to one week ahead. It assists in day to day operation of a power system and helps in performing functions like unit commitment, economic load dispatch, automatic generation Control, power system security analysis and electric price forecasting in deregulated power markets [8 – 9].

For accurate STLF the most important aspect to deal with is the system load model. The load response is non-linear and is influenced by many factors such as time of day, day of week; weather specially temperature, season, market economic conditions and several random factors depending on social activities of the population, geography and characteristics of terrain [10]. Hipert in [7] indicates the effect of load of previous hour, same hour on previous days, same hour on same day of previous week and exogenous variables such as weather-related variables, weekdays, weekends and holidays

on having a substantial impact on load and to be considered while modeling load for STLF.

1.2 Thesis Objective

STLF has been one of the most researched areas of power systems studies and until now numerous forecasting methods have been introduced and employed [1, 5, 7 – 12]. Many literature reviews have classified these methods into two main types which are: conventional or classical and computational or artificial intelligence techniques. Classical methods mostly include Time series models, Regression models and State space models in conjunction with estimation using Kalman filter. On the other hand models based on Expert systems, Fuzzy Inference, Machine Learning such as Support Vector machines (SVMs) and Neural networks (NNs) along with some hybrid models are classified as artificial intelligence methods. Classical methods treat load response as a linear device. The load series however, is a non-linear function especially considering distributed generation and deregulated market conditions. It requires computational intelligence based techniques [7] in order to accommodate non-linearity in load model. Standalone classical methods are no more sufficient enough to fulfill the needs of efficient economic operation.

Among artificial intelligence methods Neural Networks (NNs) has gained much of attention during last three decades [12]. NNs are inspired by function of neurons in human brains [13]. They perform parallel processing in order to model non-linear systems such as load series by using artificial neuron model. The key reason for popularity of NNs is their ability to model load series with the help of effective training algorithm which uses historical and current data in order to teach the behavior of load to

NN and make it prone to any non-linearity that occur due to certain events during the operation.

STLF through NN has been a popular topic among researchers and an immense amount of studies have been carried out to investigate the performance of different techniques and architectures of NN to improve the accuracy of STLF. In particular, ANNSTLF method introduced by Khotanzad et. al [14] in 1998 uses a Multilayer Perceptron (MLP) feed forward NN (FFNN) architecture trained by the error Back Propagation (BP) method. Two ANN forecasters are used in parallel. Base load is predicted by one method and change in load by the other and a Recursive Least Squares (RLS) algorithm is used to combine both and give final forecast. Finally the effect of weather elements has been taken care of by transforming them into effective temperatures. 35 utility in various states of USA and Canada adopted this method. Similar sort of studies have been done in [15 – 21].

Moreover, Cascaded NN (CNN) and Recurrent NN (RNN) have not been given much attention recently. Though some studies do test these networks in [22 – 27] but a thorough model is not being developed to take advantage of all benefits that might be achieved as a result of using these advanced architectures.

This thesis embodies an effort to comprehensively investigate CNNs and RNNs to perform STLF and explore their advantages and disadvantages Hybrid network architectures combining CNN and RNN are introduced. In this architecture inputs are being cascaded in each layer as well as a local and global feedback loops are also used. Furthermore, two 3-layered versions of FFNN and CNN are also demonstrated to produce good forecasts. A total of eight architectures are individually tested and their

results are compared with each other. Case study data from three Canadian jurisdictions (Nova Scotia, New Brunswick and Ontario) are used to test the performance of understudy networks on real data. These jurisdictions are selected because their load demands differ from each other both in size and response. In this way the adaptability of each network is tested.

This thesis also includes a study of load response of each jurisdiction in order to find out dominating factors affecting the load and a common set of variables for each dataset has been selected. Data preprocessing, separation of data into weekdays and weekends and training and testing sets of data have been carried out. Day ahead and weak ahead forecasts for weekdays and separate forecast for weekend are performed using all architectures for the three aforementioned datasets.

1.3 Structure of the Thesis

This introductory chapter is followed by an overview of STLF and a literature review in Chapter 2. Chapter 3 gives a theoretical background of Artificial Neural network architectures and learning algorithms used in this thesis. Chapter 4 treats the implementation of ANNs to perform STLF. Chapter 5 presents a comparative analysis and discussion of the results achieved for all three jurisdictions. Lastly, chapter 6 contains conclusion of the research and suggestions for future work.

Chapter 2: Short-Term Load forecasting (STLF)

2.1 A brief description of Power Systems

Power systems are complex energy transformation systems where energy is converted from various forms to electrical energy and transmitted and distributed to customers. It consists of three stages namely generation, transmission and distribution. A generalized structure of a power system is shown in figure 1:

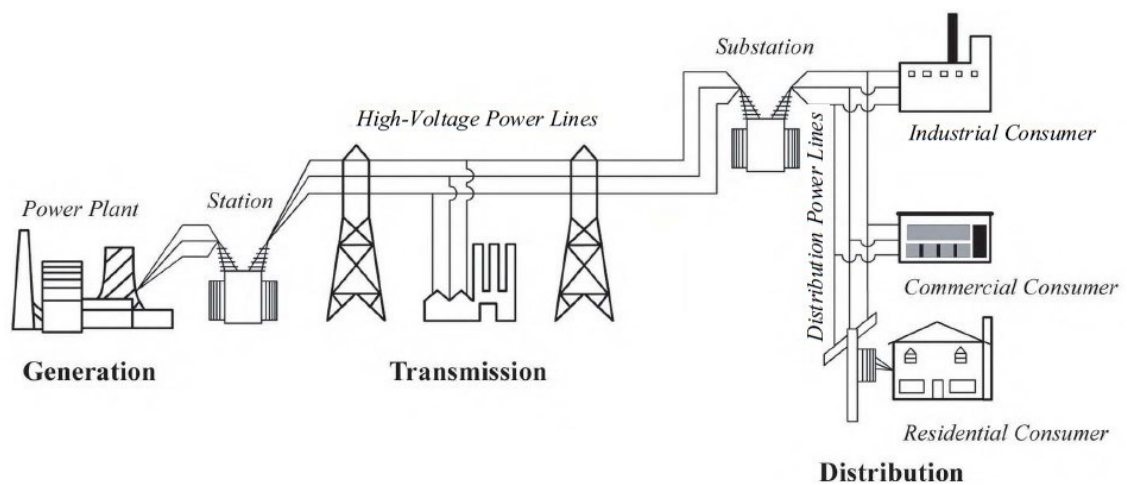


Figure 1: A generalized structure of Power System [3]

The bulk of generation is performed at distant locations from where power is transferred through high voltage transmission lines to distribution centers from where it's stepped down and delivered to the customers. Increasingly, distributed generators (DG) are being included in the network at the transmission and distribution stages in order to reduce reliance on non-environmentally friendly fossil-fuel based generation, reduce cost of power, and reducing power losses in transmission and distribution. A single line diagram of an electric power system is shown in Fig. 2 showing various elements of power system.

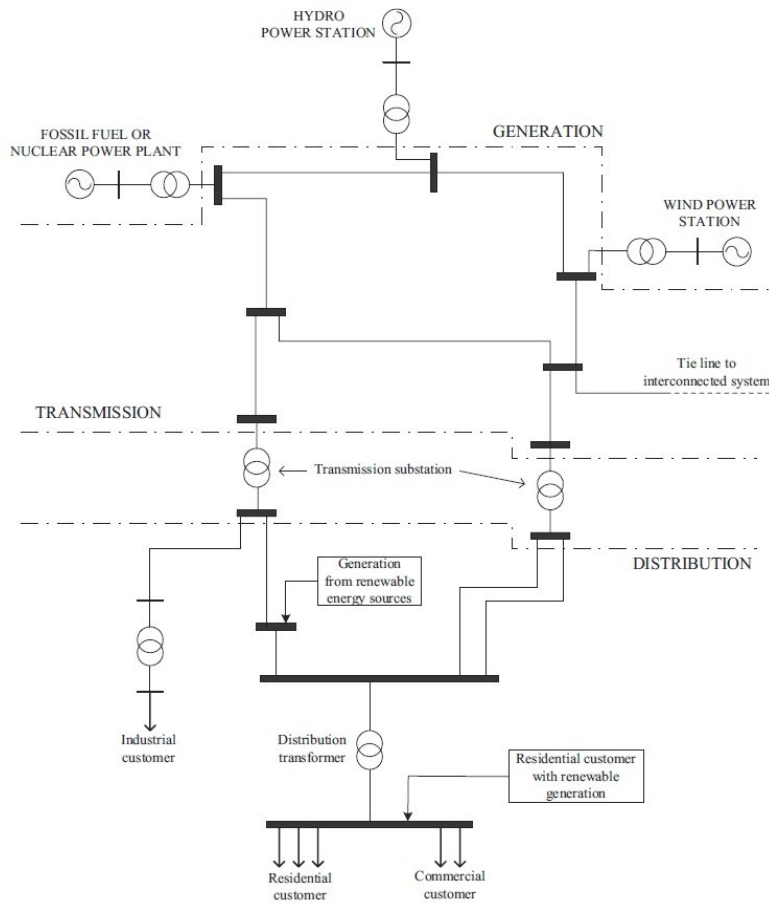


Figure 2: A single line diagram of electric Power System [1]

The architecture of power system is different for different facilities depending on several factors related to the generating sources and load requirements. All present-day power systems are ac three-phased systems with predefined voltage and frequency.

2.2 The necessity & importance of STLF

One of the most important tasks of power system management also referred as Energy Management Systems (EMS) is to fulfill load requirements at all time. Since traditionally there has been no significant storage, a power system consists of multiple generating sources in order to provide enough redundancy to system. The extra

generation resources are also referred to as spinning reserves. To have a certain amount of spinning reserves floating all the time is extremely crucial and EMS is responsible for taking this operating decision. Since various jurisdictions set various capacity margins an accurate short term load forecasting is required to maintain the security of the system and help EMS to have sufficient spinning reserves.

Kyriakides in [1] explained the significance of STLF in modern day operation of power systems. For any power utility to function economically without compromising security and reliability of the system has become a challenging task due to deregulated market and load demand, regulations on power quality and several system requirements. Power system studies such as load flow, contingency analysis, protection studies, unit commitment or economic load dispatch all rely on the results of STLF. For example the output of STLF is input for load flow studies and contingency analysis. With the help of these studies line flows, required active and reactive power generation as well as bus voltages are determined. When a contingency occurs, prevailing conditions can be determined.

Furthermore, STLF is valuable for operations specialists to prepare for corrective measures like increasing spinning reserves, load shedding, switching interconnections, starting up any available generators or even exporting power from other utilities in case of any type of faults estimated. Also with increase in deregulation the accurate forecasting has become more essential both for plant and system operators on or market players like generation or transmission companies. The preceding actions highlight the importance of STLF in maintaining security and reliability of a power system.

2.3 A standardized approach to STLF

STLF is a complex problem in which the relationship between load and the variables influencing it is generalized in order to predict future load values. Generally the correlation between load and related variables is identified and then quantified using an appropriate technique [28]. In recent years many different ways particularly those involving artificial intelligence based methods have been developed and employed in industry to perform STLF. López and Valero in [29] tried to come up with a standard model based on recently published methods. The standard model introduced in [29] contains processes which are common in most methods under study. The outline of the model introduced contains five stages which are found to be common between forecasting models. The effectiveness and inadequacy of a forecasting model depend on all stages of standard model shown in figure 3 below followed by a brief description of each stage:

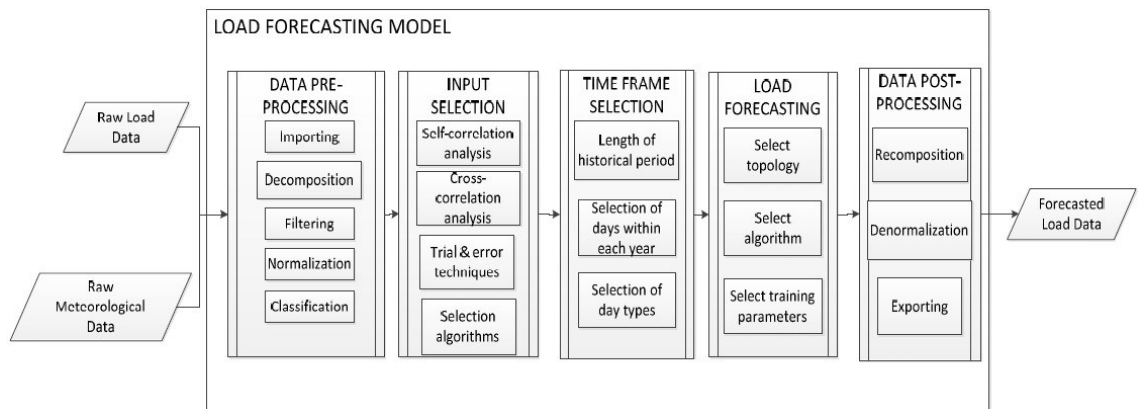


Figure 3: Standardized structure of STLF model [28]

2.3.1 Data Pre-processing

The raw input data of load and other effective variables are presented in this stage. The raw data is first filtered to deal with abnormalities or missing values and then it is

normalized within a range to simplify the process. The normalized data is then classified in accordance with the load profile such as weekday, weekend, holiday or any special human social activity. Decomposing the data into components might also be required in some models. It involves transforming data using the Fourier or Wavelets transforms or fuzzy membership functions so that it is ready to apply respective forecasting technique.

2.3.2 Variable Selection

In this stage statistical tools such as self-correlation, cross-correlation and trial and error analysis are applied to variables affecting future forecast load values. Past load values affecting future load are selected through self –correlation analysis. Furthermore, in cross-correlation factors related to other series like the influence of weather related variables on the load series are determined. Also trial and error methods employed and checked to see if there are any other variables affecting the load other than those determined by statistical analysis. So, this stage transforms raw data into a database including prominent information related to the forecast.

2.3.3 Training Frame Selection

This stage is deemed to be a most crucial stage of the model. It involves selection of training frame from data which teaches forecasting engine the desired behavior of load. Therefore, selection of these data points should be done very carefully as accuracy of forecasting engine depends on them. The training period that likely replicates the forecasting period should be identified and chosen. This is similar to the classification done during the first stage. This step may consist of selecting days from similar times of

the year, days based on analogous events or days with similar weather and social activities like weekdays and weekends.

2.3.4 Load Forecasting

First, the forecasting engine is selected from techniques such as Neural Networks, Fuzzy systems, Support Vector Machines and various models based in regression analysis. Next the topology of a forecasting engine is decided and employed according to given conditions. This step may include selection of neurons and layers or fuzzy rules or certain parameters for the selected forecasting engine. Next a training algorithm is selected to use the training data from previous stage to train the forecasting engine and estimate the weights for each input. This training allows the forecasting engine to learn load behavior. Finally a training duration is selected. This is the allowable time following which training needs to be stopped to avoid any aliases. The outcome at this stage is a forecast which needs further processing before being presented as the final forecasting results.

2.3.5 Data Post-processing

The steps taken during the first stage are reversed and the data from load forecast is transformed back to an understandable format. In addition performance evaluation of the forecast is also performed in this stage. Depending on what steps were taken in the first stage on raw data, the forecasted data is either denormalized or recomposed. Then the performance of the forecast is determined through calculating accuracy – usually represented by Mean Absolute Percentage Error (MAPE), robustness – the percentage and frequency of any erratic results and universality – the performance of the same model

using different datasets and conditions. To establish the efficiency of a certain model to claim its applicability in real world applications, a certain amount of results are required.

2.4 Literature Review

As discussed in the introduction, the STLF is categorized into two types: traditional and computational intelligence based methods.

2.4.1 Traditional Methods

Early methods for STLF were based on statistical approaches, which may be further classified as Time series and Regression based models and State Space models- Kalman filtering for estimation.

2.4.1.1 Time-series models

Also called univariate models the time series based models perform statistical analysis of auto-correlation of the load series [29], which means that the load is presented as a function of its past observations. Time series methods include Autoregressive (AR), Moving Average (MA), and Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) models. These methods are analyzed and explained in [1, 8, 10, 11, 30, 31].

2.4.1.2 Regression Models

Since load is also influenced by several other factors like weather conditions and social events, regression based models take these variables into account to model load series. Moghram and Rahman in [11] presented a load expression based on multiple linear regressions in which certain weather and non-weather variables represented the

electrical load. Papalexopoulos *et. al* [32] presented a regression based STLF method utilizing the data for past 15 days in conjunction with ARIMA and Multiple regression based peak load models. Haida and Muto in [33] combined multivariate regression analysis with a transformation technique to build a model based on a temperature-load relationship which utilizes both the annual and the latest trend in temperature and its influence on load in order to forecast. Mbamalu and El-Hawary in [34] introduced a seasonal multiplicative autoregressive model using an iteratively reweighted least squares algorithm.

2.4.1.3 State Space Models based Kalman Filter Estimation

In a state-space model the load value is treated as a state variable in a state space formulation, with the system being represented by set of equations called state equations and measurement equations. Kalman filter is an algorithm which uses the state space model to estimate the value of the next state which in the case of STLF is the forecast. Kalman filter use present and past data to predict the values of future load. Soliman and Al-Hamadi in [35] used a time-varying load model based weather and load variables with moving window, then the Kalman filter is used to estimate the coefficients of each variable. This load model is applied to real data to forecast hour ahead and day ahead loads. Separate profiling for weekend and weekdays has also being taken into account.

Zheng and Girgis *et. al* [9] introduced a hybrid method which combine the Kalman filter and Wavelet transformation techniques. The original signal is represented as a unique wavelet transform in the form of coefficient. These coefficients are then used as coefficients of Kalman filter. In addition time splitting is performed to split data into 3 groups. A recursive Kalman filter is applied on each set to predict next state of

coefficients and the signal is reconstructed to represent the load forecast. This method is implemented on weather insensitive and a weather sensitive model with the performance of the latter found to be better. A simple Kalman filter model is also presented in [11] which is similar to the one presented by Soliman and Al-Hamadi in [35] but without using a moving window and weather variables. Identification of state space model parameters is the most important and difficult task in Kalman filter based methods.

2.4.2 Artificial Intelligence Methods

Hipert in [10] stated that due to their ability to relate load with physical components traditional models have assisted on field engineers and system operators to understand the behavior of load. However, there has been a significant evolution of the complexity in the system over time which has made load response non-linear [28]. Also the scarcity of resources has made accurate STLF necessary for optimal operation. Ongoing research to fulfill requirements of the electric utility industry has turned attention to the application of artificial intelligence based techniques for STLF. Artificial intelligence methods include many different approaches to STLF, and researchers have widely classified them as Knowledge Based Expert System (KBES), Fuzzy Inference System (FIS), Artificial Neural Network (ANN) and Support Vector Machine (SVM) [1], [8] and [10]. There are some hybrid models as well but they also make use of either one of these techniques or in combination with classical methods.

2.4.2.1 Knowledge Based Expert System

Knowledge Based Expert System (KBES) perform electric load forecasting by utilizing the experience of human experts with years of on-field experience in electricity

generation and synchronizing it with computer program called an expert system [9, 36]. Rahman and Bhatnagar et al. [37] introduced an expert system which uses load forecasting knowledge of the on-field experts to build its knowledge base KB. On the basis of this KB a set of IF-THEN rules were built which defined the relationships between variations in system load with respect to changes in natural and network conditions affecting electricity usage. These rules act as a forecasting engine to produce results. The variables in the knowledge base of this expert system consist of year, month, hour, day type, temperature, and load data which require continuous update to achieve good forecasting results.

Ho et al. [38] proposed a KBES to perform STLF for the Taiwan power system. Historical load data and weather parameters of 5 years along with on field operator's knowledge were used and eleven day types were determined. Results indicate an improved performance of the developed algorithm over conventional the Box-Jenkins time series method. Rahman and Hazim in [39] proposed an expert system which was site-independent. A parameterized rule base was created by extracting knowledge about the load and influencing factors. A parameter database play the role of a knowledge base for the system which changes from site to site and it is coupled with a parameterized rule base to generate forecasts. The technique was tested on real data from several sites located in 7 states of the USA and the results demonstrated low forecasting errors.

Some hybrid methods have also been developed which combine KBES with other forecasting methods. Dash et al. [40] combined fuzzy neural networks with expert systems. The neural networks inputs were the fuzzy membership values of the load and weather variables which were used to predict the output membership values. An adaptive

fuzzy correction scheme was adopted for getting final load. Extensive studies were performed for all seasons and average, peak and hourly load forecasts were generated which resulted in minimized forecasting error. Mohamad in [41] combined expert systems and neural networks to produce hourly load forecasting in Egypt. The operator was allowed to intervene during the forecasting process to make the process more accurate. Kim et al. [42] used a hybrid approach for STLF for Korea Electric Power Corporation. The method was a two-step approach in which initial load prediction was obtained by using a trained neural network and then temperature changes and holidays are accommodated by a fuzzy expert system.

2.4.2.2 *Fuzzy Inference System*

The Fuzzy Inference System (FIS) generalizes the Boolean logic which has the values of 0 and 1 to fuzzy membership functions. It means that an input can have certain meaningful values instead of the two extremes of Boolean logic. After the inputs are represented by these membership functions, a set of fuzzy rules are used to generate outputs. These outputs are then defuzzified to obtain original forecasts. An implementation of a fuzzy inference model to achieve STLF has recently been done by Ahmadi and Bevrani in [43]. Six inputs were used namely last day's consumption, last week's consumption, forecasted temperature, and last day's temperature, weather and day type values. The output consumption along with other two inputs representing electricity consumption were divided into seven Gaussian membership functions, temperature inputs had three and weather had four membership functions and the day type was represented by holiday and weekday membership functions. The inference model was implemented to a real load of an electric utility in Iran with satisfactory results. Two hybrid methods

combining fuzzy inference, neural networks and expert systems presented by Dash et al. [40] and Kim et al. [42] have already been discussed in the last section.

Furthermore, Soozanchi and Yaghabi et al. [44] studied the design of (STLF) systems with the help of Adaptive Neural-Fuzzy Inference System (ANFIS). First the probability of chaos and predictability rate of time series load curve by a Lyapunov exponent was reviewed to determine the rate and horizon of forecasting. Then a multi Artificial Neuro-Fuzzy Inference System (ANFIS) was used to obtain the desired forecast. A separate adaptive neural-fuzzy system for each season of the year was used. Data used as input for multi ANFIS include the date of the day, maximum and minimum temperature, climate condition and the load consumed in previous days. The results points out the impact of temperature on load forecast.

Petrounias and Kodogiannis et al. [45] developed a novel fuzzy logic system applied in the structure of a neural network (AFINN) model to perform STLF for the Power System of the Greek Island of Crete. A clustering algorithm consisting of two stages was used to determine the rules, number of fuzzy sets, and initial values of the parameters of the fuzzy membership functions. These parameters of membership functions were tuned by the error back propagation algorithm. The forecast results were achieved through a minimum and maximum load time series. The proposed intelligent forecasting model gave better forecasting results as compared to conventional neural networks models.

2.4.2.3 *Artificial Neural Network (ANN)*

The artificial neural network (ANN) is one of the most recent popular methods for STLF. Based on the principle of biological neuron, the central part of an ANN is an

artificial neuron [13]. An artificial neuron model performs the sum of the product of the inputs and the respective weights connected to them and pass the sum through a transfer function to produce output. In order to use ANN to perform STLF, the first thing to do is to select a suitable architecture, for example feed forward (FNN), Cascaded (CNN), Recurrent (RNN), Spiking (SNN), Wavelet (WNN), Neuro-Fuzzy Networks and several others. Next, a learning algorithm needs to be selected to train the network. Furthermore, the number and connectivity of the layers and neuron is to be taken care of. In general, neural networks are used to model the load series at first using a number of inputs which affect the load demand and a suitable architecture. Then historical data is used to train the network by using an appropriate learning algorithm so that weights for every input may be determined. Lastly present online information is used as the input of the trained network in order to obtain outputs. The outputs acquired are linear or nonlinear mathematical functions of its inputs. These outputs may be the inputs of other networks as well as the actual forecasted loads [8].

Bakirtzis and Petridis [15] used a Feed forward Neural Network (FFNN) to forecast the load for Greek Public Power Cooperation. The Back Propagation (BP) algorithm was used for training the network with 63 inputs, 24 hidden and 24 output neurons. The historical load data of past two days along with historical and present maximum and minimum temperature data for each day was used as inputs. The network was trained by using the data for one year. It was found that it took more training time for an increased or decreased number of hidden neurons and an optimum amount of 24 neurons were used in the hidden layer. A single ANN with an increased input of day type along with seven ANNs for each day was used in the experiments. In addition the data

was divided into weekdays, weekends and holidays for the single ANN. Results show that the single ANN outperforms the other forecasts. Moreover, Khotanzad et. al [14] introduced a method called ANNSTLF. FFNN architecture was used trained by the BP algorithm. Two ANN forecasters were used in parallel. One predicted the base load while the other forecasts the change in load. A Recursive Least Square (RLS) algorithm was used to combine both to produce the final forecast. Finally the effect of weather elements was taken care of by transforming them into effective temperature. Data from 10 utility companies was tested and results show increased accuracy as compared to using a single ANN for the forecast. 35 utility in various states of USA and Canada adopted this method. Ortiz-Arroyo, Skov and Huynh in [16] used a single ANN with BP. They took the month of the year, the day of the week, holiday and week number as inputs and experimented with different combinations of number of hidden neurons and epochs to training the network. The results achieved showed a 29% increase in accuracy over the SVM methods that won the EUNITE (European Network of Excellence on Intelligent Technologies for Smart Adaptive Systems) competition in 2001. Dai and Wang [17] also adopted FFNN for forecasting. They used pattern recognition to determine the inputs and forecasted 24 hour ahead load of the Jiangxi Province power system of China. Likewise Gu and Wang et. al [18] presented FFNN with additive weather features and normalized data. Along with the loads and minimum and maximum temperatures they used six different weather features as one of the inputs. This weather feature input had six levels each denoted by numerical values between 0 and 1. These levels were thundershower, shower, and moderate rain, light rain, cloudy and sunny. The data from Shanghai district

of China was tested. After training and simulation the method was found to be applicable for that location.

Different variations in the same method of FFNN were also implemented and tested in recent publications during past 3 years. Shi and Niu in [19] used different transfer functions for input and hidden layers and forecasted the load by considering it a time series prediction problem. Moreover, Ghate and Tasre et. al [20] adopted FFNN and presented a seasonal forecast. Similar data variables were used as defined above but months were grouped according to 4 seasons in order to study the variations in load. The BP algorithm was used for training FFNN in all the aforementioned methods. Wang, Niu and Ji in [21] also selected the BP to train FFNN as forecasting engine but with an Improved Variable Learning (IVL) rate. They used the Genetic Algorithm in conjunction with BP to improve learning speed. The results showed improvement in the learning rate as well as accuracy over conventional FFNN.

AlFuhaid and El-Sayed in [22] applied Cascaded ANN (CNN) architecture for STLF. The load for every half hour for the next 24 hours was forecasted using the historical load and weather data. First a simple FFNN was used to predict peak, minimum and daily energy. These predictions were used as additional inputs to the CNN. The data from the electric power system of Kuwait was used for the training and testing of the network. Results indicated a reduction in Mean Absolute Percentage Error (MAPE) from 3.367% to 2.707 % when CNN was compared with FFNN.

Sajjad and Farshif in [49] presented a new hybrid forecast strategy based on CNN to improve accuracy of STLF. This method used the wavelet transform to filter the inputs to remove any corrupted data. A CNN consisting of three neural networks forecasted the

load and an intelligent two-stage feature selection were used to tune the CNN through a cross-validation technique to generate forecast. Loads of PJM and New York City were forecasted. Results from the proposed algorithm were found to be most accurate compared to conventional FFNN, CNN and SVR methods.

A Recurrent Neural Network (RNN) was employed for STLF by Lee and Choi in [24] with an adaptive learning rate. The network presented was diagonally recurrent (DRNN) meaning only the neurons in the hidden layer were recurrent as compared to the fully connected recurrent neural network (FRNN) where all neurons are coupled to one another. It was illustrated that DRNN require fewer weights than FRNN and was faster in convergence. In addition BP Algorithm with adaptive learning rate also ensured rapid convergence. The forecasting accuracy throughout the year was evaluated to consider the effect of seasonal load variations. Simulation results proved DRNN forecast to be better than FRNN.

In addition, two Bilinear RNNs (BLRNNs) were proposed by Park in [23] and [25]. The network architecture of a BLRNN consisted of Input, Hidden and Output layers. The neuron in input layer receives two feedbacks from hidden layer, one with a delay and the other is direct. Both feedbacks were further coupled with original inputs before connecting to the input neuron. In [25] the BLRNN-based predictor was implemented and its performance was compared with the conventional Multi-Layer Perceptron NN (MLPNN). Load data from the North-American Electric Utility (NAEU) was used for the task. The proposed BLRNN model showed an improvement of about 28% decrease in MAPE of performance in comparison with the traditional MLPNN. To further improve the accuracy Park, Woo and Han in [23] presented an Adaptive Multi-

resolution based bilinear Recurrent Neural Network (AMBLRNN). In this architecture the learning process of BLRNN of [25] was improved further by applying the wavelet transform for multi-resolution analysis of the data. The same data as used in [25] was employed for testing. The performance of AMBLRNN was found to be even better than BLRNN and MLPNN by 10 % - 25% in terms of MAPE.

Some hybrid models are also used which involve the combination of ANNs and other methods by different means to achieve improved accuracy. Zhao and Su et. al [27] employed a Weiner model to represent the load series. Then Kalman filter and Elman Neural Network (ENN) were used to fit the linear and non-linear parts of the system. An ENN is a type of ANN which contains 4 layers – input, feedback, hidden and output layers. No input is coupled with feedback unlike the BLRNN. Kalman filter was used to minimize disturbances from system's linear part and ENN was applied for non-linear load prediction.. A simulation example was demonstrated which used normalized load and weather data from a southern city in China for training and testing the algorithm. The results showed good learning efficiency and precision. A similar method using ENN for STLF was also established by Xia and Yand in [26]. The historical load and comprehensive consideration of various meteorological factors were used as input for the network. The result proved the effectiveness of this method for STLF for the power grid of Yichang city.

Zhang and Li in [47] forecasted the load using FFNN with BP and LSSVM separately. After getting the results from both, a minimum variance method was used to combine the results and produce final forecast. The results showed the improved MAPE of the combined results as compared to separate forecasts. Ko and Lee in [48] introduced

a hybrid model to harvest better forecasting results by utilizing the advantages of Radial Basis Function NN (RBFNN), Support Vector Regression (SVR) and Dual Extended Kalman Filter (DEKF). The method used SVR for initialization of RBFNN and DEKF was implemented for training it. The model was applied to actual load data for one, three and seven day ahead forecasts. Weekdays, weekends and holidays were separately treated for forecasting. The results proved fast convergence and better accuracy of the proposed SVR-RBFNN-DEKF based method over DEKF-RBFNN or Gradient Decent GRD-RBFNN.

2.4.2.4 *Support Vector Machine (SVM)*

Based on the principle of structural risk minimization (SRM) support vector machines (SVMs) are able to achieve an optimum networks structure. In order to perform STLF SVMs use Vapnik's ϵ -insensitive loss function [50] to treat STLF as nonlinear regression estimation problems which is also referred in literature as support vector regression (SVR). At first a nonlinear mapping (by using kernel functions) of the input data is performed which maps it into a high dimensional (feature) space. Then simple linear functions are used to create linear decision boundaries in the feature space. When we compare with ANNs the process of selecting architecture is replaced here by the problem of choosing suitable kernel and parameters for the support vector machine. Thus to decrease the forecasting error, values of the cost of the error C , ϵ -insensitive tube width, the type of kernel and the parameters inside the kernel functions must be chosen carefully during the SVM training [6,36,50,52]. Guo and Liang in [51] used a Gaussian kernel function along with user specified parameters for forecasting load in a city of China. The training sets were created by selecting similar days having a weather profile

same as the forecasting day in order to avoid interference of the non-correlative samples. The training of SVMs was done using these preprocessed samples. The SVM based method for forecasting was found to be efficient.

Similarly, Turkay and Demren in [52] applied SVM to model the load in order to represent the nonlinear relations with the factors that affect the load. The load and weather data of Istanbul (European Side) were used. The inputs included the past load data, daily average temperature, calendar days, holidays and electricity price. The program LibSVM was used to implement SVM to perform STLF. Moreover, Ye and Liu et. al [53] compared SVM with BPNN and Time series methods. The electric load of Hubei (China) was used for the comparison. The forecasting results indicated a MAPE of 1.91% for SVM which proved to be the best in contrast to BPNN (4.06%) and time series (4.47%).

Hong et. al [36] introduced a SVR model with immune algorithm (IA) to forecast the electric loads. IA was applied for the parameter determination of SVR model. The proposed SVR model with IA (SVRIA) produced better forecasting results than the other methods like SVM with Gaussian kernel, regression model and ANN model. Furthermore, Nie and Liu in [54] presented a hybrid model combining the autoregressive integrated moving average (ARIMA) and SVMs. They used ARIMA to forecast the basic linear part of load and SVMs to forecast the sensitive non-linear part of load. The hybrid model resulted in high forecasting accuracy.

Chapter 3: Artificial Neural Networks and proposed algorithm

3.1 Introduction

Artificial neural networks are inspired by functionality of human brain. The capability of a human brain to remember, recall, correlate, interpret, recognize and reason have made its functionality attractive to make machines to imitate it in order to model and simulate and perform a variety of tasks. The human brain consists of billions of parallel computing elements called neurons. They interconnect via a network of axons, synapses and dendrites. Thus, the human brain is analogous to a vast interconnected Electrical switching network accustomed to biochemical processes. Figure 4 illustrates components of a biological neural network:

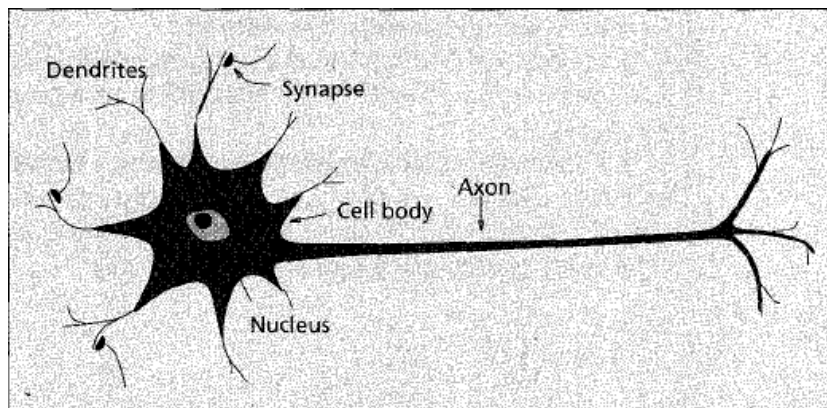


Figure 4: Biological Neuron [13]

A typical brain cell has four major regions namely, cell body, dendrites, axon and synapse. Dendrites receive information from neurons through axons via an axon-dendrite connect called the synapse. An axon is like a transmission line which carries an impulse from neuron that splits into thousands of branches. A synapse is present at the end of each branch which converts the impulse into electrical effects that pass through dendrites and result in inhibitory or excitatory response in connected neurons. The neuron collects

signals from all dendrites connected to it and reacts to the total of its aggregated input. When the excitatory input exceeds the inhibitory input by an amount called the threshold, the neuron fires a spike of electrical activity down its axon. Furthermore, learning takes place by changing the effectiveness of synapses which in turns changes the influence of one neuron on another [55, 56].

It has been observed that a biological network performs temporal integration and summation of incoming signals. The result is a complex biochemical process which is much different from digital computation but the characteristic feature of the neuron is that the signal in the nerve fiber is either absent or has a maximum value. It means that information exchange between cells is transmitted by means of a binary signal. Figure 5 shows a model of biological neurons:

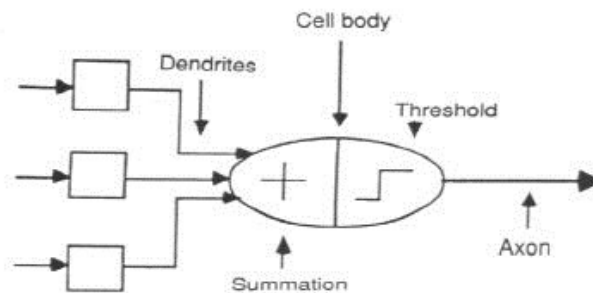


Figure 5: Biological Neuron Model [58]

3.2 Artificial Neuron Model

McCulloch and Pitts in [57] first described the model of an artificial neuron which can imitate functioning of biological neuron. In real life an artificial neuron model performs a sum of products n of inputs denoted here by p and weights w connected to them and bias b . This sum is fed to a nonlinear transfer function $f(\cdot)$ to produce the

output. The weights w and bias b are adjustable parameters. Figure 6 demonstrates an artificial neuron model [59]:

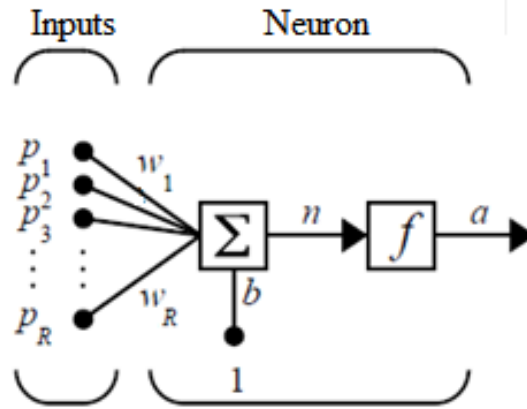


Figure 6: Artificial Neuron Model [59]

Mathematically, n is given by the following equations,

$$n = w_1 p_1 + w_2 p_2 + \dots + w_R p_R + b \quad (3.1)$$

$$n = \sum_{j=1}^R w_j p_j + b \quad (3.2)$$









where R is number of inputs. The neuron's output is given by,

$$a = f(n) = f\left(\sum_{j=1}^R w_j p_j + b\right) \quad (3.3)$$

3.3 Transfer Functions

The output of a neuron depends upon the transfer function $f(\cdot)$ selected. A transfer function (also called an activation function) might be linear or non-linear and must be selected so as to fulfill the requirements of the problem at hand. Table 1 lists commonly used activation functions along with their response and equations.

Table 1: Transfer Functions

Name	Input/Output Relation	Response
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$	
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$	
Linear	$a = n$	
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$	
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$	
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$	
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$	

3.4 Network Architectures

3.4.1 Single Layer Feed forward Neural Network

The single-layer feed-forward neural network also known as single layer perceptron (SLP) contains one level of connection of neurons. It simply models the relationship of the input vector p to the output vector a through the adjustment of weights w . SLP might consists of multiple numbers of neurons. A simple SLP is shown in figure 7:

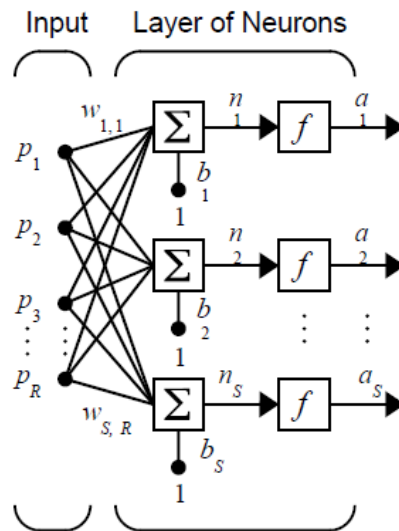


Figure 7: Single Layer Perceptron Network [59]

The output vector can be determined using the following equation,

$$n_s = \sum_{j=1}^R w_{sj} p_j + b_s \quad (3.4)$$

$$a_s = f(n_s) = f\left(\sum_{i=1}^S \sum_{j=1}^R w_{ij} p_j + b_s\right) \quad (3.5)$$

R is the number of inputs and S is the number of neurons. The above expression can be written in vector form as:

$$a = Wp + b \quad (3.6)$$

The input, weight, bias and output matrices are given by:

$$p = [p_1 \dots p_R]$$

$$W = \begin{bmatrix} w_{1,1} & \dots & w_{1,R} \\ \vdots & \ddots & \vdots \\ w_{S,1} & \dots & w_{S,R} \end{bmatrix}$$

$$b = [b_1 \dots b_S]$$

$$a = [a_1 \dots a_S]$$

3.4.2 Multilayer Feed forward Neural Network

A multilayer feed-forward network or multi-layer perceptron (MLP) consists of multiple layers of neurons. Each layer may consist of the same or different numbers of neurons. In between input and output layers exists single or multiple hidden layers. Each layer may have the same or different activation functions and weight vectors. The output of every layer becomes an input for next hidden layer until it reaches output layer. Figure 8 shows a MLP network with two hidden layers:

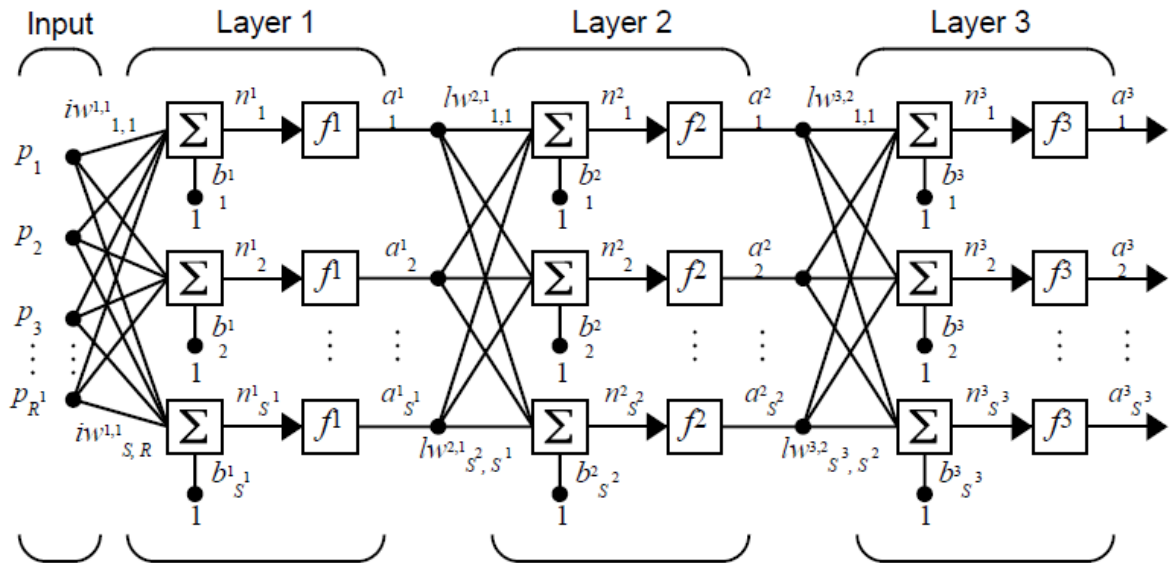


Figure 8: Multi-layer Perceptron Network [59]

Let l denote the number of hidden layers and $l = L$ for the output layer, the equations for the outputs of every hidden layer and the final output is given by following equations,

Hidden layer 1 ($l = 1$):

$$a_{1S^1} = f_1(n_{1S^1}) = f_1\left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{1S^1}\right) \quad (3.7)$$

Here S^l denotes the number of neurons in layer l .

Hidden layer 2 to output layer ($l = 2 \dots L$):

$$a_{lS^l} = f_l(n_{lS^l}) = f_l\left(\sum_{i=1}^{S^l} \sum_{j=1}^{S^{l-1}} w_{ij} a_{(l-1)j} + b_{lS^l}\right) \quad (3.8)$$

Since in the above figure $L = 3$:

$$a_{3S^3} = f_3(n_{3S^3}) = f_3\left(\sum_{i=1}^{S^3} \sum_{j=1}^{S^2} w_{ij} a_{2j} + b_{3S^3}\right) \quad (3.9)$$

$$a_{3S^3} = f_3\left(\sum_{i=1}^{S^3} \sum_{j=1}^{S^2} w_{ij} f_2\left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij} f_1\left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{1S^1}\right) + b_{2S^2}\right) + b_{3S^3}\right) \quad (3.10)$$

3.4.3 Cascaded Neural Network

The cascaded neural network (CNN) architecture is a type of MLP network in which each input and the output of every hidden layer is cascaded to next layer. Figure 9 demonstrates a 3-layered CNN:

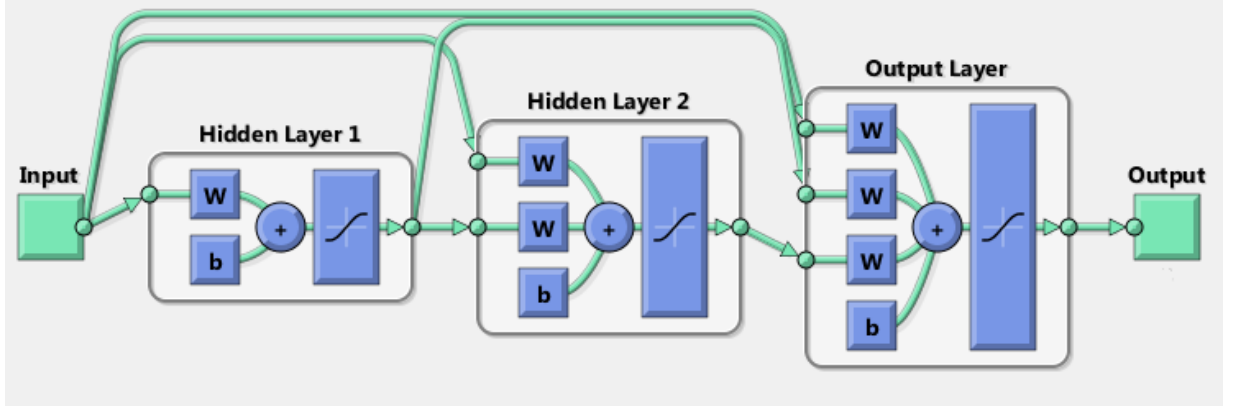


Figure 9: 3-layered Cascaded Neural Network

The equations for the outputs of all the layers are as follows:

Hidden layer 1 ($l = 1$):

$$a_{1S^1} = f_1(n_{1S^1}) = f_1\left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij}p_j + b_{1S^1}\right) \quad (3.11)$$

Hidden layer 2 to last hidden layer ($l = 2$):

$$a_{2S^2} = f_2(n_{2S^2}) = f_2\left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij}a_{1j} + \sum_{i=1}^{S^2} \sum_{j=1}^R w_{ij}p_j + b_{2S^2}\right) \quad (3.12)$$

a_{2S^2} may be treated as the output of a CNN with one hidden layer.

Output Layer ($l = L = 3$):

$$a_{3S^3} = f_3(n_{3S^3}) = f_3\left(\sum_{i=1}^{S^3} \sum_{j=1}^{S^2} w_{ij}a_{2j} + \sum_{i=1}^{S^3} \sum_{j=1}^{S^1} w_{ij}a_{1j} + \sum_{i=1}^{S^3} \sum_{j=1}^R w_{ij}p_j + b_{3S^3}\right) \quad (3.13)$$

$$\begin{aligned}
 a_{3S^3} = & f_3 \left(\sum_{i=1}^{S^3} \sum_{j=1}^{S^2} w_{ij} f_2 \left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij} f_1 \left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{1S^1} \right) + \sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{2S^2} \right) \right. \\
 & \left. + \sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij} f_1 \left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{1S^1} \right) + \sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{3S^3} \right)
 \end{aligned}
 \tag{3.14}$$

3.4.4 Recurrent Neural Network

A recurrent neural network (RNN) contains a feedback path from the output to the input along with the associated weights. There are many types of RNN discussed in the literature which are used for prediction. Basically there are two types of feedback paths, a local feedback and a global feedback. Local feedback is achieved by the recurrent path within the hidden layer, while a global feedback is produced by connecting output to input of the network [60]. Figures 10 and 11 show both local and global RNNs:

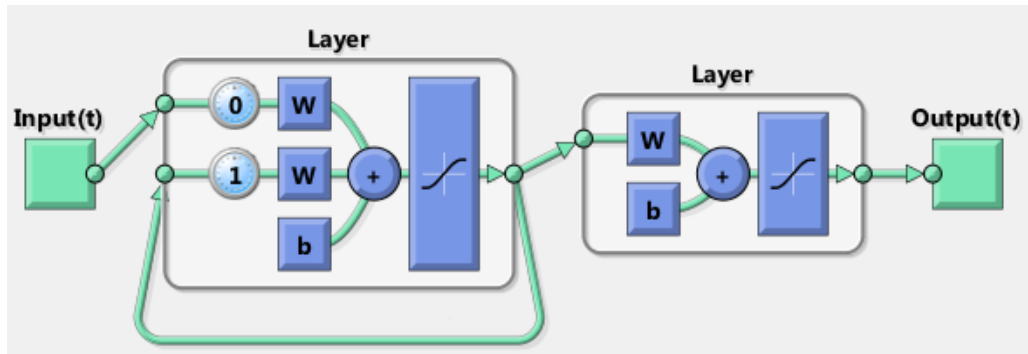


Figure 10: Recurrent Neural Network with Local feedback

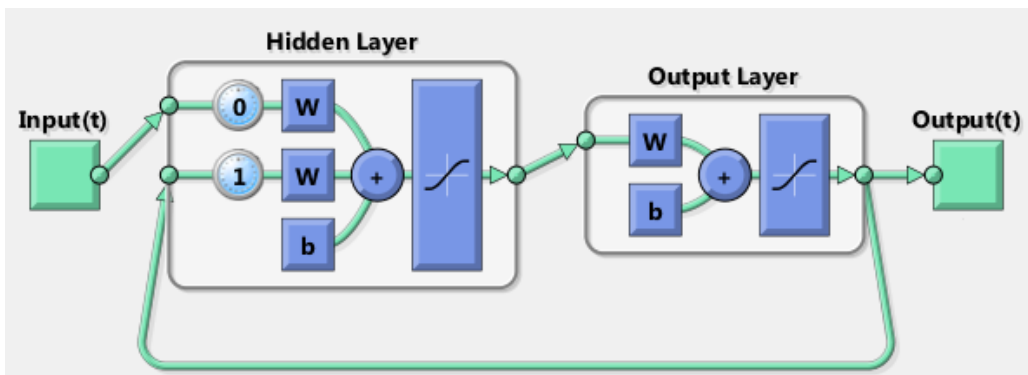


Figure 11: Recurrent Neural Network with Global feedback

Let t be current time. The output of Recurrent NN with Local feedback (RNNL) is given by :

$$a_{1S^1}(t) = f_1(n_{1S^1}) = f_1\left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij}p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^2} w_{ij}a_{1j}(t-1) + b_{1S^1}\right) \quad (3.15)$$

$$a_{2S^2}(t) = f_2(n_{2S^2}) = f_2\left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij}a_{1j}(t) + b_{2S^2}\right) \quad (3.16)$$

$$a_{2S^2}(t) = f_2\left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij}f_1\left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij}p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^2} w_{ij}a_{1j}(t-1) + b_{1S^1}\right) + b_{2S^2}\right) \quad (3.17)$$

And the output of Recurrent NN with Global feedback (RNNG) is given by following equations:

$$a_{1S^1}(t) = f_1(n_{1S^1}) = f_1\left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij}p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^3} w_{ij}a_{2j}(t-1) + b_{1S^1}\right) \quad (3.18)$$

$$a_{2S^2}(t) = f_2(n_{2S^2}) = f_2\left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij}a_{1j}(t) + b_{2S^2}\right) \quad (3.19)$$

$$a_{2S^2}(t) = f_2\left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij}f_1\left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij}p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^3} w_{ij}a_{2j}(t-1) + b_{1S^1}\right) + b_{2S^2}\right) \quad (3.20)$$

3.4.5 Hybrid Neural Network Architectures

Hybrid Neural Network architectures that are introduced in this thesis combine a CNN with both recurrent networks. These networks contain a cascaded connection of input to output layer as well as local or global feedback paths. The network which combines the CNN with RNNL is called Cascaded – Recurrent NN with Local feedback (CRNNL). Figure 12 shows its architecture:

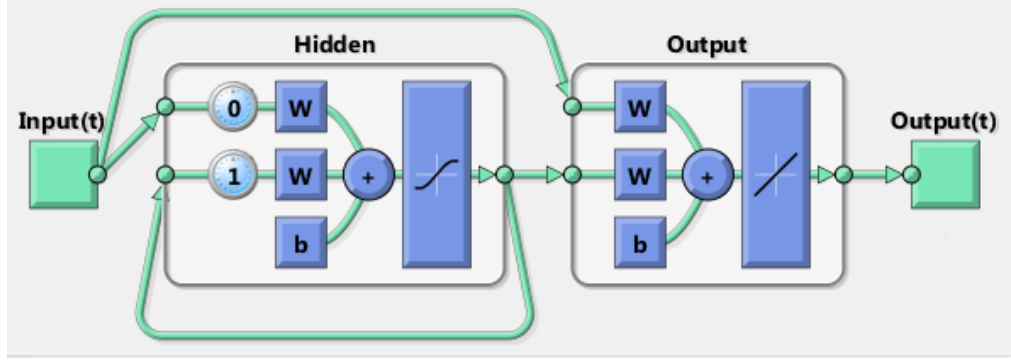


Figure 12: Cascaded Recurrent Neural Network with Local feedback

The output equation of the CRNNL is derived as follows:

$$a_{1S^1}(t) = f_1(n_{1S^1}) = f_1 \left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^2} w_{ij} a_{1j}(t-1) + b_{1S^1} \right) \quad (3.21)$$

$$a_{2S^2} = f_2(n_{2S^2}) = f_2 \left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij} a_{2j} + \sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{2S^2} \right) \quad (3.22)$$

$$a_{2S^2} = f_2 \left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij} f_1 \left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^2} w_{ij} a_{1j}(t-1) + b_{1S^1} \right) + \sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{2S^2} \right) \quad (3.23)$$

Moving forward the second hybrid network combines the CNN with RNNG to create a Cascaded – Recurrent NN with Global feedback CRNNG network shown in Figure 13:

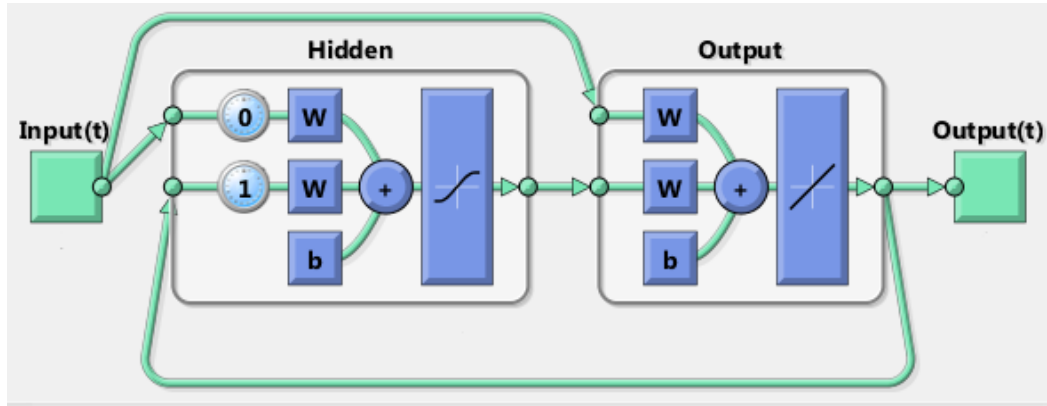


Figure 13: Cascaded Recurrent Neural Network with Global feedback

The output of CRNNG is given by:

$$a_{1S^1}(t) = f_1(n_{1S^1}) = f_1 \left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^3} w_{ij} a_{2j}(t-1) + b_{1S^1} \right) \quad (3.24)$$

$$a_{2S^2} = f_2(n_{2S^2}) = f_2 \left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij} a_{2j} + \sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{2S^2} \right) \quad (3.25)$$

$$a_{2S^2} = f_2 \left(\sum_{i=1}^{S^2} \sum_{j=1}^{S^1} w_{ij} f_1 \left(\sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + \sum_{i=1}^{S^1} \sum_{j=1}^{S^3} w_{ij} a_{2j}(t-1) + b_{1S^1} \right) + \sum_{i=1}^{S^1} \sum_{j=1}^R w_{ij} p_j + b_{2S^2} \right) \quad (3.26)$$

3.5 Network Training

The training of a neural network is the process of modifying network architecture and updating the connection weights and biases so that a network response may follow a particular behavior. To achieve this task different learning algorithms are used. The purpose of a learning algorithm is to minimize the error function by altering network parameters by utilizing previous information as samples. When a solution is reached the finalized parameters are stored which means that network has learned how to respond if new information is received for a particular application.

There are two types of learning algorithms; supervised and unsupervised. In supervised training the response of a network is known. The network is simulated by applying the input patterns. The output result is compared with the desired response and an error signal is generated. This error signal is used to direct changes in the network parameters so as to minimize it and the network is taught to produce the desired response. Conversely, unsupervised training is used in situations when desired response of a network is unknown. In this thesis, supervised learning is used.

In order to train a network to reach a certain target d , output a is compared with it to obtain the error signal e . The error signal e is the difference between the target d and output o :

$$e_{uv}(k) = d_{uv} - a_{uv} \quad (3.27)$$

u is the index of patterns, from 1 to U , where U is the number of input patterns. v is the index of outputs, from 1 to V , where V is the number of outputs. e_{uv} is the training error at the output v when applying input pattern u . For evaluating training process the Least Mean Squared (LMS) error for all the input patterns and the resulting outputs is defined as:

$$E(U, V) = \frac{1}{2} \sum_{u=1}^R \sum_{v=1}^S e_{uv}^2 \quad (3.28)$$

The average squared error is given by:

$$E_{ave} = \frac{1}{U * V} \sum_{u=1}^R \sum_{v=1}^S E(u, v) \quad (3.29)$$

E_{ave} represents the cost function which is a measure of learning performance. The objective of a learning process is to minimize it by adjusting network parameters. The learning algorithm used in this thesis is the error back propagation which is discussed next.

3.5.1 Error Back Propagation Algorithm

The error back propagation (EBP) algorithm is the most popular learning mechanism for ANN especially when it is used to perform STLF. The algorithm is derived from delta learning algorithm and updates the weights by applying a correction $\Delta w_{ij}(k)$

proportional to the partial derivative called gradient. This method is also named as gradient-descent method [61].

Step 1: Initialization

The first step of training process is to initialize the weights by selecting any random weights from uniform distribution with zero mean.

Step 2: Forward Computation

A training sample is presented to the network to perform forward computation in order to compute error signal. Consider the MLP network of figure 8, we rewrite equation (3.4) for n which is sum of product at the input of activation function:

$$n_j(k) = \sum_{i=0}^R w_{ji}(k) a_j(k) \quad (3.30)$$

k is the index of iterations. i and j are the indices of weights, from 1 to N , where N is the total number of weights. R is the total inputs applied to neuron i . The weight w_{i0} is equal to the bias b_i applied to neuron i . The output is given by:

$$o_j(k) = f_j(n_j(k)) \quad (3.31)$$

Now from equation (3.21) the error signal can be achieved.

$$e_j(k) = d_j - a_j \quad (3.32)$$

Let C be the total number of neurons in the output layer equation (3.22) becomes:

$$E(K) = \frac{1}{2} \sum_{j=1}^c e_j^2(k) \quad (3.33)$$

Step 3: Backward Computation

Case 1 : Neuron j is situated at Output Layer

The error signal is propagated backward. The weight correction $\Delta w_{ij}(k)$ can be represented as a gradient $\frac{dE(k)}{dw_{ij}}$

$$\Delta w_{ji}(k) = \frac{dE(k)}{dw_{ji}(k)} \quad (3.34)$$

By applying the chain rule we can calculate the partial derivative.

$$\frac{dE(k)}{dw_{ji}(k)} = \frac{dE(k)}{de_j(k)} \frac{de_j(k)}{da_j(k)} \frac{da_j(k)}{dn_j(k)} \frac{dn_j(k)}{dw_{ji}(k)} \quad (3.34)$$

$$\frac{dE(k)}{de_j(k)} = e_j(k) \quad (3.35)$$

$$\frac{de_j(k)}{do_j(k)} = -1 \quad (3.36)$$

$$\frac{da_j(k)}{dn_j(k)} = f'_j(n_j(k)) \quad (3.37)$$

$$\frac{dn_j(k)}{dw_{ji}(k)} = a_i(k) \quad (3.38)$$

Finally, using equations (3.28) – (3.31):

$$\frac{dE(k)}{dw_{ji}(k)} = -e_j(k) f'_j(n_j(k)) a_i(k) \quad (3.39)$$

The correction weight is given by:

$$\Delta w_{ji}(k) = -\eta \frac{dE(k)}{dw_{ji}(k)} \quad (3.40)$$

η is called the learning-rate parameter of back propagation algorithm. By using equations (3.32 and 3.33) we get:

$$\Delta w_{ij}(k) = -\eta \delta_j(k) \quad (3.41)$$

$\delta_j(k)$ is the gradient and is defined as:

$$\delta_j(k) = \frac{dE(k)}{dw_{ji}(k)} \quad (3.42)$$

Case 2 : Neuron j is situated at Hidden Layer

When a neuron is located at the hidden layer the desired response is not specified. An error signal for a hidden layer needs to be evaluated in terms of the error signal of all neurons to which hidden neuron is connected directly. The gradient is given by:

$$\delta_j(k) = \frac{dE(k)}{dn_j(k)} = \frac{dE(k)}{da_j(k)} \frac{da_j(k)}{dn_j(k)} = \frac{dE(k)}{da_j(k)} f'_j(n_j(k)) a_i(k) \quad (3.43)$$

From equation (3.22) we have:

$$E(k) = \frac{1}{2} \sum_{m \in C} e_m^2(k) \quad (3.44)$$

m is the output node and C is all the neurons in the output layer. Here the output neurons are denoted by a different letter to avoid confusion. Differentiating it with respect to $a_j(k)$ we get:

$$\frac{dE(k)}{da_j(k)} = \sum_m e_m(k) \frac{de_m(k)}{da_j(k)} \quad (3.45)$$

Applying the chain rule:

$$\frac{dE(k)}{da_j(k)} = \sum_m e_m(k) \frac{de_m(k)}{dn_m(k)} \frac{dn_m(k)}{da_j(k)} \quad (3.46)$$

Also:

$$e_m(k) = d_m(k) - a_m(k) = d_m(k) - f_m(n_m(k)) \quad (3.47)$$

Hence:

$$\frac{de_m(k)}{dn_m(k)} = -f'_m(v_m(k)) \quad (3.48)$$

From equation (3.24) $n_m(k)$ is given as:

$$n_m(k) = \sum_{j=0}^m w_{mj}(k) a_j(k) \quad (3.49)$$

Differentiating equation (3.43) with respect to $a_j(k)$:

$$\frac{dn_m(k)}{da_j(k)} = w_{mj}(k) \quad (3.50)$$

Equation (3.40) yields,

$$\frac{dE(k)}{da_j(k)} = - \sum_m e_m(k) f'_m(v_m(k)) w_{mj}(k) = - \sum_m \delta_m(k) w_{mj}(k) \quad (3.51)$$

Using equation (3.45) in (3.37) we can get the local gradient $\delta_j(k)$:

$$\delta_j(k) = - f'_j(n_j(k)) a_i(k) \sum_m \delta_m(k) w_{mj}(k) \quad (3.52)$$

Finally, the weight update is given by:

$$w(k+1) = w(k) + \Delta w_{ji}(k) = w(k) - \eta \delta_j(k) \quad (3.53)$$

Equation (3.47) describes the weight update in the BP algorithm. It has some problems which are related to the slow convergence. One of the main reasons for this is the occurrence of local minima. As the algorithm always changes weights towards steepest gradient where the curvature of error surface is low but it might briefly have to rise before reaching the general descent also referred as global minima. Sometimes the algorithm gets stuck at these points, thus slowing or inhibiting convergence [63]. This situation is demonstrated in Figure 14:

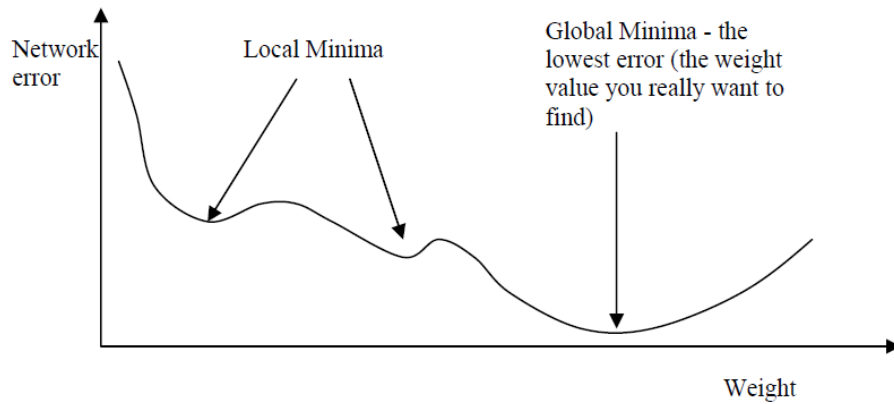


Figure 14: Local and Global Minima [63]

Furthermore the selection of learning rate or step size also plays an important role in convergence. When the gradient is steep, a small step size needs to be selected so that any minimum point which might occur should not be missed. Also when the gradient change is gentle, a small step size would slow the training process. But if a big step size is selected it might result in missing some minima [62]. This is demonstrated in Figure 15:

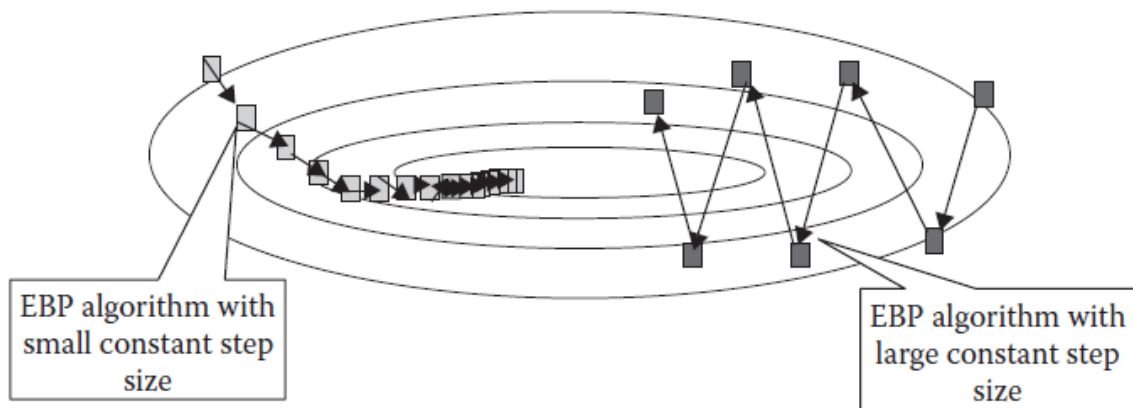


Figure 15: Effect of learning rate coefficient on learning curve [62]

These problems are addressed by improvements to the BP algorithm and making use of second order differential equations. Next some advanced algorithms are discussed.

3.5.2 Newton Algorithm

It is assumed that the local gradient is a function of weights which are linearly independent from each other:

$$\delta_j(k) = F_j(w_1(k), w_2(k), \dots, w_N(k)) \quad (3.54)$$

Starting with Taylor series we get:

$$\begin{aligned} \delta_1(k) &\approx \delta_{1,0}(k) + \frac{d\delta_1(k)}{dw_1(k)} \Delta w_1(k) + \frac{d\delta_1(k)}{dw_2(k)} \Delta w_2(k) + \dots + \frac{d\delta_1(k)}{dw_N(k)} \Delta w_N(k) \\ \delta_2(k) &\approx \delta_{2,0}(k) + \frac{d\delta_2(k)}{dw_1(k)} \Delta w_1(k) + \frac{d\delta_2(k)}{dw_2(k)} \Delta w_2(k) + \dots + \frac{d\delta_2(k)}{dw_N(k)} \Delta w_N(k) \\ &\vdots \\ \delta_N(k) &\approx \delta_{N,0}(k) + \frac{d\delta_N(k)}{dw_1(k)} \Delta w_1(k) + \frac{d\delta_N(k)}{dw_2(k)} \Delta w_2(k) + \dots + \frac{d\delta_N(k)}{dw_N(k)} \Delta w_N(k) \end{aligned} \quad (3.55)$$

Differentiating again with respect the weight,

$$\frac{d\delta_j(k)}{dw_i(k)} = \frac{\frac{dE(k)}{dw_j(k)}}{dw_i(k)} = \frac{d^2E(k)}{dw_i(k)dw_j(k)} \quad (3.56)$$

Equation (3.49) becomes:

$$\begin{aligned} \delta_1(k) &\approx \delta_{1,0}(k) + \frac{d^2E(k)}{dw_1^2(k)} \Delta w_1(k) + \frac{d^2E(k)}{dw_1(k)dw_2(k)} \Delta w_2(k) + \dots + \frac{d^2E(k)}{dw_1(k)dw_N(k)} \Delta w_N(k) \\ \delta_2(k) &\approx \delta_{2,0}(k) + \frac{d^2E(k)}{dw_2(k)dw_1(k)} \Delta w_1(k) + \frac{d^2E(k)}{dw_2^2(k)} \Delta w_2(k) + \dots + \frac{d^2E(k)}{dw_2(k)dw_N(k)} \Delta w_N(k) \\ &\vdots \\ \delta_N(k) &\approx \delta_{N,0}(k) + \frac{d^2E(k)}{dw_N(k)dw_1(k)} \Delta w_1(k) + \frac{d^2E(k)}{dw_N(k)dw_2(k)} \Delta w_2(k) + \dots + \frac{d^2E(k)}{dw_N^2(k)} \Delta w_N(k) \end{aligned} \quad (3.57)$$

To minimize the total error function E , the left hand sides of above equations should be zero. The above equations will become:

$$\begin{aligned}
-\delta_{1,0}(k) &\approx \frac{d^2 E(k)}{dw_1^2(k)} \Delta w_1(k) + \frac{d^2 E(k)}{dw_1(k)dw_2(k)} \Delta w_2(k) + \dots + \frac{d^2 E(k)}{dw_1(k)dw_N(k)} \Delta w_N(k) \\
-\delta_{2,0}(k) &\approx + \frac{d^2 E(k)}{dw_2(k)dw_1(k)} \Delta w_1(k) + \frac{d^2 E(k)}{dw_2^2(k)} \Delta w_2(k) + \dots + \frac{d^2 E(k)}{dw_2(k)dw_N(k)} \Delta w_N(k) \\
&\vdots \\
-\delta_{N,0}(k) &\approx + \frac{d^2 E(k)}{dw_N(k)dw_1(k)} \Delta w_1(k) + \frac{d^2 E(k)}{dw_N(k)dw_2(k)} \Delta w_2(k) + \dots + \frac{d^2 E(k)}{dw_N^2(k)} \Delta w_N(k)
\end{aligned} \tag{3.58}$$

In matrix form we can write the above equation as:

$$\begin{bmatrix} -\delta_1(k) \\ -\delta_2(k) \\ \vdots \\ -\delta_N(k) \end{bmatrix} = \begin{bmatrix} -\frac{dE(k)}{dw_1(k)} \\ -\frac{dE(k)}{dw_2(k)} \\ \vdots \\ -\frac{dE(k)}{dw_N(k)} \end{bmatrix} = \begin{bmatrix} \frac{d^2 E(k)}{dw_1^2(k)} & \frac{d^2 E(k)}{dw_1(k)dw_2(k)} & \dots & \frac{d^2 E(k)}{dw_1(k)dw_N(k)} \\ \frac{d^2 E(k)}{dw_2(k)dw_1(k)} & \frac{d^2 E(k)}{dw_2^2(k)} & \dots & \frac{d^2 E(k)}{dw_2(k)dw_N(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{d^2 E(k)}{dw_N(k)dw_1(k)} & \frac{d^2 E(k)}{dw_N(k)dw_2(k)} & \dots & \frac{d^2 E(k)}{dw_N^2(k)} \end{bmatrix} \times \begin{bmatrix} \Delta w_1(k) \\ \Delta w_2(k) \\ \vdots \\ \Delta w_N(k) \end{bmatrix} \tag{3.59}$$

The square matrix is called the Hessian matrix.

$$H(k) = \begin{bmatrix} \frac{d^2 E(k)}{dw_1^2(k)} & \frac{d^2 E(k)}{dw_1(k)dw_2(k)} & \dots & \frac{d^2 E(k)}{dw_1(k)dw_N(k)} \\ \frac{d^2 E(k)}{dw_2(k)dw_1(k)} & \frac{d^2 E(k)}{dw_2^2(k)} & \dots & \frac{d^2 E(k)}{dw_2(k)dw_N(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{d^2 E(k)}{dw_N(k)dw_1(k)} & \frac{d^2 E(k)}{dw_N(k)dw_2(k)} & \dots & \frac{d^2 E(k)}{dw_N^2(k)} \end{bmatrix} \tag{3.60}$$

Equation (3.53) becomes:

$$\delta(k) = H(k)\Delta w(k) \tag{3.61}$$

$$\Delta w(k) = -H^{-1}(k)\delta(k) \tag{3.62}$$

$$w(k+1) = w(k) - H^{-1}(k)\delta(k) \tag{3.63}$$

3.5.3 Gauss – Newton Algorithm

The Newton method requires the calculations of second order derivatives. In order to calculate Hessian matrix the second-order derivatives of the total error function should be computed, Jacobean matrix is introduced in Gauss – Newton method for this purpose.

$$J(k) = \begin{bmatrix} \frac{de_1(k)}{dw_1(k)} & \frac{de_1(k)}{dw_2(k)} & \dots & \frac{de_1(k)}{dw_N(k)} \\ \frac{de_2(k)}{dw_1(k)} & \frac{de_2(k)}{dw_2(k)} & \dots & \frac{de_2(k)}{dw_N(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{de_j(k)}{dw_1(k)} & \frac{de_j(k)}{dw_2(k)} & \dots & \frac{de_j(k)}{dw_N(k)} \end{bmatrix} \quad (3.64)$$

Integrating equations (3.27) and (3.37):

$$\delta_j(k) = \frac{dE(k)}{dw_i(k)} = \frac{d\frac{1}{2}\sum_{j=1}^s e_j^2(k)}{dw_i(k)} = \sum_{m=1}^c \frac{de_m(k)}{dw_i(k)} e_m(k) \quad (3.65)$$

$$\delta(k) = Je(k) \quad (3.66)$$

where C is the total number of neurons in the layer and m is index of e and:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_c \end{bmatrix} \quad (3.67)$$

The element in Hessian matrix is given as:

$$h_{ij} = \frac{d^2E(k)}{dw_i(k)dw_j(k)} = \sum_{m=1}^c \frac{de_m(k)}{dw_i(k)} \frac{de_m(k)}{dw_j(k)} + \sum_{m=1}^c \frac{d^2e_m(k)}{dw_i(k)dw_j(k)} e_m(k) \quad (3.68)$$

According to Newton method the later term is closed to zero [62]. Hessian matrix can be expressed in terms of Jacobean matrix as:

$$H(k) = J^T(k)J(k) \quad (3.69)$$

$$w(k+1) = w(k) - (J^T(k)J(k))^{-1} J(k)e(k) \quad (3.70)$$

3.5.4 Levenberg-Marquardt Back Propagation Algorithm

The Gauss–Newton algorithm simplifies the process of learning as calculations of the second order derivative of total error function are replaced by the Jacobean matrix. This method seems to have fast convergence if error surface is likely to have a quadratic approximation but otherwise it might prove divergent [62].

The Levenberg–Marquardt Back Propagation (LMBP) algorithm combines the Error Back Propagation (EBP) method with Gauss–Newton algorithm. In this way it achieves the fast convergence of Gauss–Newton algorithm and steadiness of EBP algorithm. LMBP is a little slower in convergence than Gauss-Newton algorithm but it ensures convergence even for a complex error surface. The LMBP uses EBP algorithm to learn the complex curvature of error surface to find a local curvature for which making a quadratic approximation is possible. It then switches to the Gauss–Newton algorithm to speed up convergence. The LMBP presents another approximation of the Hessian matrix to confirm that approximation made in Gauss–Newton method is always invertible:

$$H(k) \approx J^T(k)J(k) + \mu I \quad (3.71)$$

μ is called combination coefficient, its value is always positive and I is identity matrix.

From above equation it is evident that values of diagonal elements are greater than zero hence approximated Hessian matrix is invertible. Equation (3.65) yields:

$$w(k + 1) = w(k) - (J^T(k)J(k) + \mu I)^{-1} J(k)e(k) \quad (3.72)$$

The LMBP switches between Gauss–Newton and EBP algorithms; value of combination coefficient μ is used for this purpose. When it has a very small value equation (3.67) imitates (3.65) as Gauss–Newton, on the other hand when μ is made very large, equation

(3.67) and can be interpreted as (3.48). μ can be expressed in terms of learning rate coefficient η of the EBP algorithm as:

$$\eta = \frac{1}{\mu} \quad (3.73)$$

This thesis implements LMBP for training the neural networks. A step by step implementation of this algorithm is stated as follows:

1. Generate the least mean square error by applying randomly generated initial weights.
2. Update the weights using equation (3.67).
3. Evaluate the total error after applying the new weights.
4. If current total error is found more than previous one than reset the weights to previous values and increase the value of μ by a specified factor and return to step 2 and try again.
5. If the current total error is found lesser than previous one than make the updated weights as current and decrease the value of μ by a specified factor.
6. Go to step 2 and repeat the whole process until the value of the total error becomes smaller than the required value.

The flowchart of LMBP training described above is demonstrated in Figure 16 .

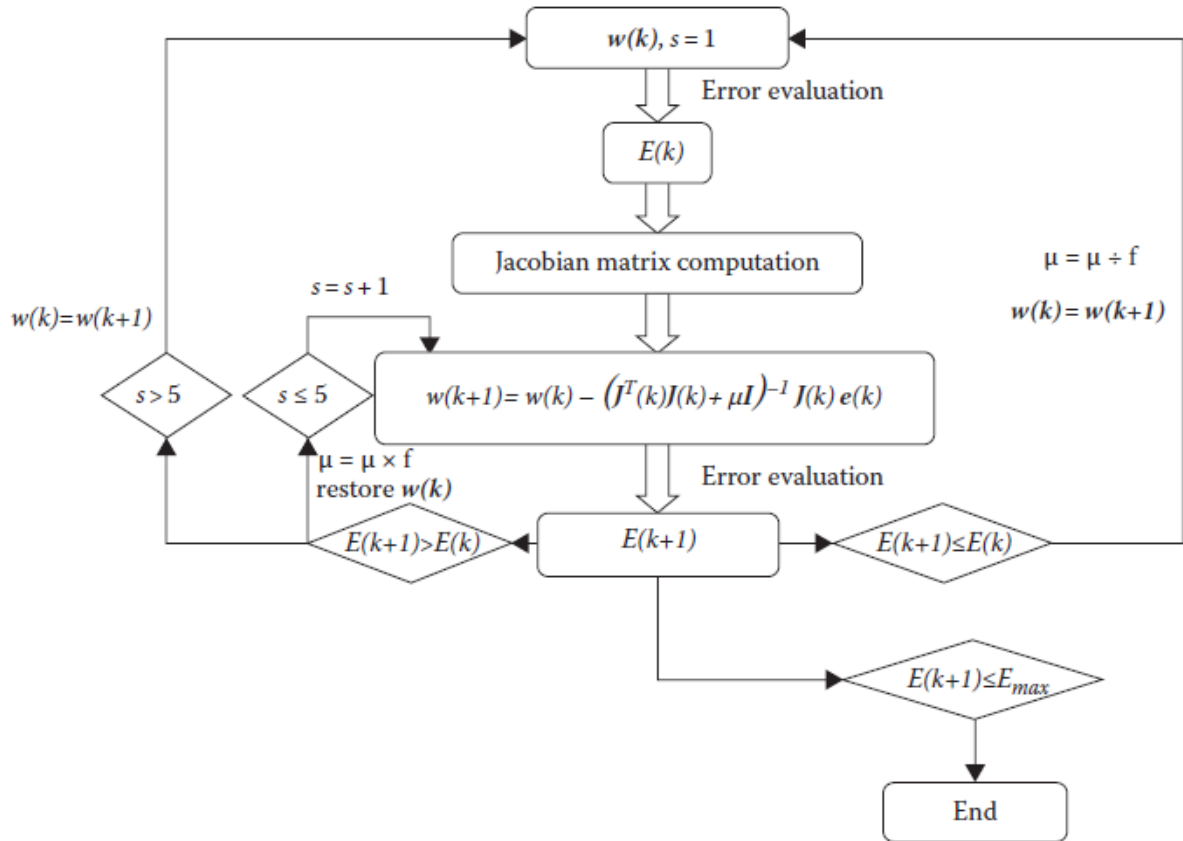


Figure 16: Flowchart of LMBP algorithm [62]

In Figure:

$w(k)$ = current weight

$w(k+1)$ = updated weight

$E(k+1)$ = current total error

$E(k)$ = previous total error

s = slope of activation function and

f = specified factor for changing the value of μ .

E_{max} = required value of total error (E)

Chapter 4: Implementation of ANN for STLF

To implement ANN to perform STLF, three Canadian provinces; Nova Scotia (NS), New Brunswick (NB) and Ontario were selected. The data for each jurisdiction was gathered via online resources [64 - 67]. The load data from June 1 to August 30, 2013 was collected through utility company websites; for Nova Scotia from NS Power Oasis database [65], for New Brunswick NB Power Operations data [66] and for Ontario Independent System Electricity (IESO) Power data [67]. Only temperature was selected as exogenous variable as it is the most influential of all meteorological parameters. Temperature data for each province was downloaded from Government of Canada Climate Database [65]. After gathering all data following procedures were followed to achieve forecasts.

4.1 Statistical Analysis and selection of Input parameters

In order to study load response and factors affecting it a statistical analysis was performed after gathering the data. The last week of August from August 24 to 30, 2013 was selected to be forecasted. The remaining data points from June 1 to August 23 were selected for training. The performance of the algorithms depends on the choices of training and testing intervals. Data of four weeks prior to the forecasting week was given preference for input selection. To study relationship between loads of previous hours of same day and previous days in a week on current load, a graph was plotted. The week selected for plot was dated from July 30 to August 4 which is the first week of August. Load curves of each day were plotted on graph of that week to observe the relationship between load demand at any specific hour and load at previous hours and previous day

same hour. The graphs containing load curves for Nova Scotia, New Brunswick and Ontario for week mentioned are demonstrated below:

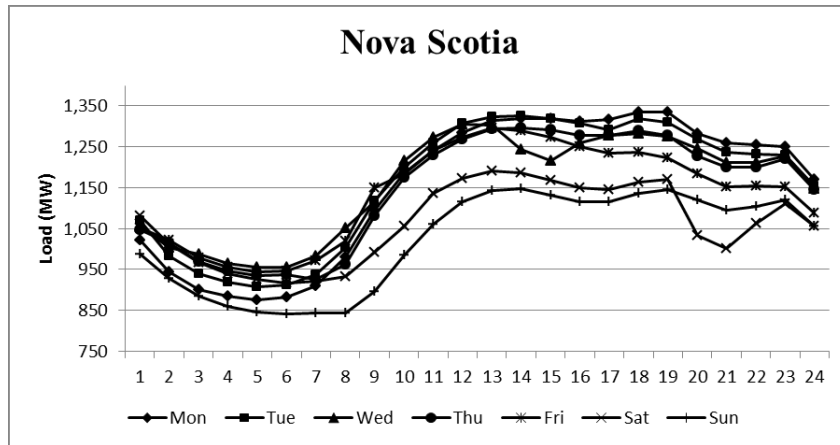


Figure 17: Daily Load Curves for Nova Scotia

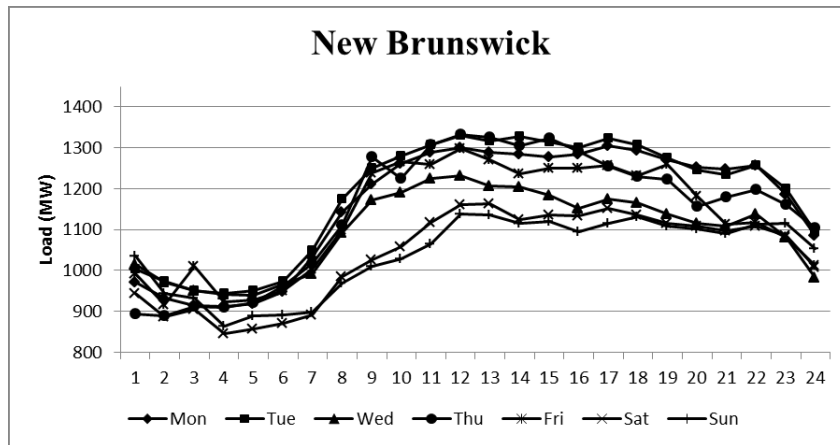


Figure 18: Daily Load Curves for New Brunswick

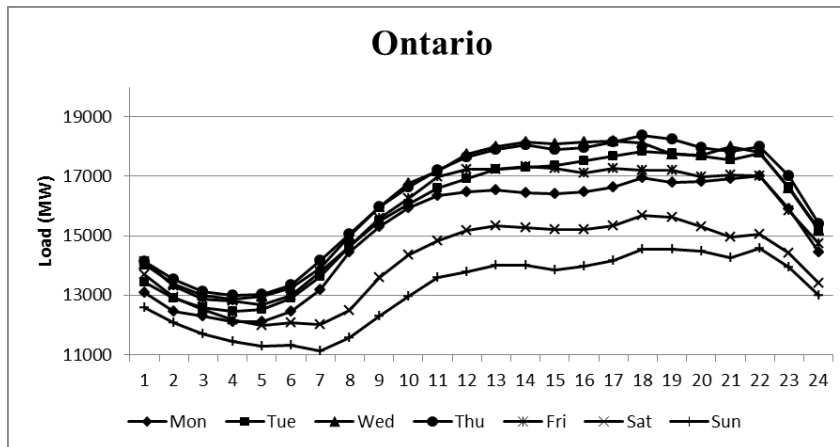


Figure 19: Daily Load Curves for New Brunswick

For all three provinces it can be seen that load values of previous hour and previous days match each other closely. The slopes almost replicate each other. Also load behavior of weekend days Saturday and Sunday is different than other days. Hence, it was decided to forecast weekdays and weekends separately. With the help of these graphs four inputs were selected for Weekdays and one for Weekend forecast. If $y(h)$ is considered to be the variable to represent present load then the inputs for weekdays include load for previous hour of same day – $y(h-1)$, same hour of previous day – $y(h-24)$, same hour two days before – $y(h-48)$ and same hour three days before – $y(h-72)$. For weekend only $y(h-1)$ was selected as their load curve generates a unique response.

Furthermore, to study the effect of the load series of previous weeks on present load a graph for each jurisdiction was produced. A sample of data was selected comprising of load data for four weeks from July 22 to August 18, 2013 with week 1 starting from July 22, week 2 starting from July 29, week 3 from August 5 and week 4 from August 12 respectively. Following figures presents graphs containing load curves of four weeks stated for three selected provinces.

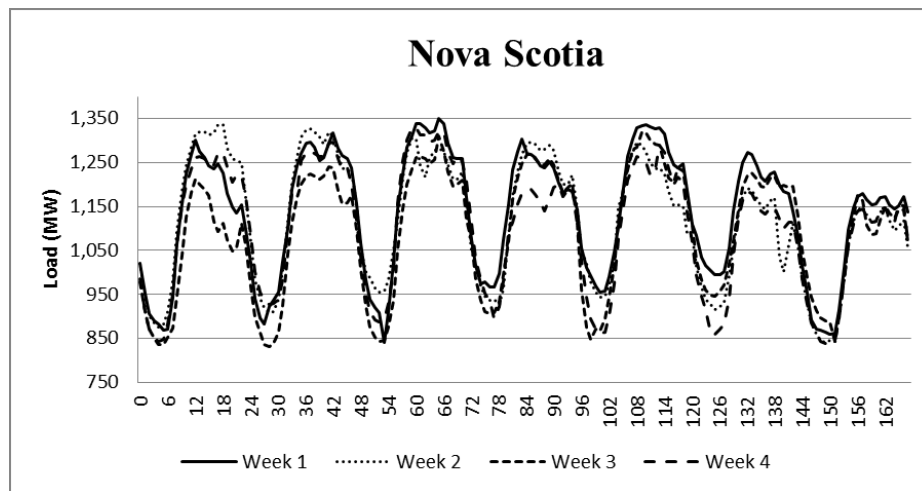


Figure 20: Weekly Load Curves for Nova Scotia

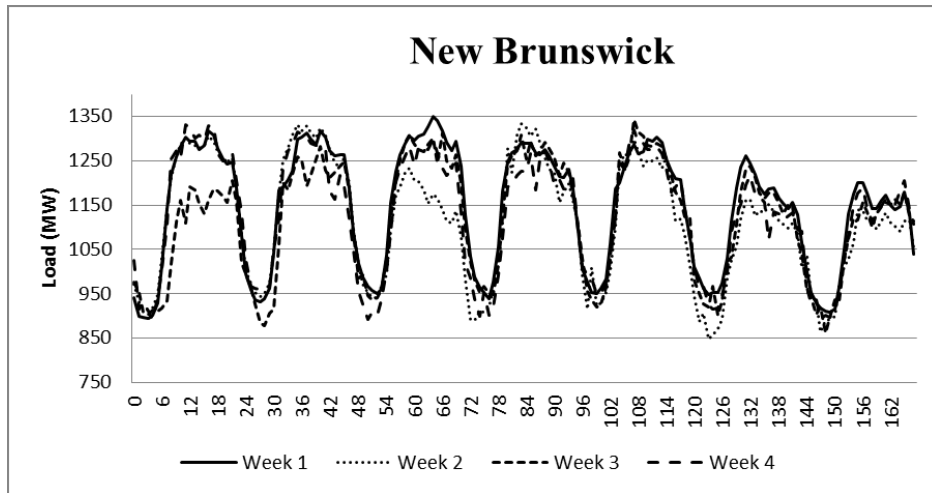


Figure 21: Weekly Load Curves for New Brunswick

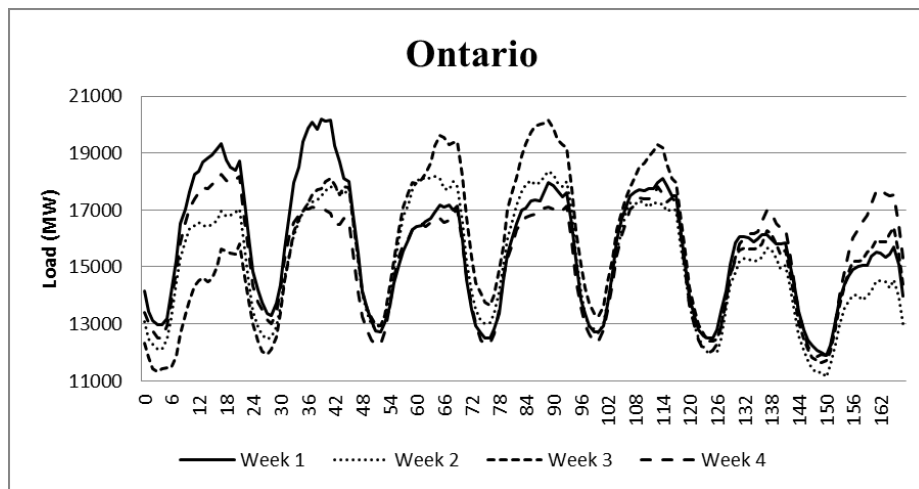


Figure 22: Weekly Load Curves for Ontario

It can be observed that load curves of four weeks are closely related to each other. Data points including first day (Monday) week 3 of Nova Scotia, Monday week 3 and third day (Wednesday) week 2 of New Brunswick, first day (Monday) all weeks, second day (Tuesday) week 1 and four of last five days week 3 of Ontario do not match closely. Therefore, they were treated as exceptions. It can also be seen that for weekends the response is more similar to each other than weekdays. Thus two inputs for weekday forecasts and three inputs for weekends were selected. The inputs selected for weekdays include the load for same hour previous week – $y(h-168)$ and same hour two weeks

before – y (h-336). For weekends the load of same hour three weeks before – y(h-504) was selected in addition of former two.

In order to study the effect of temperature on load the following sets of graphs were plotted for each province by using data points of week 1. The first graph consists of load and temperature curves superimposed on each other depicting the relationship between both variables. The second graph plots the temperatures of each day of the week.

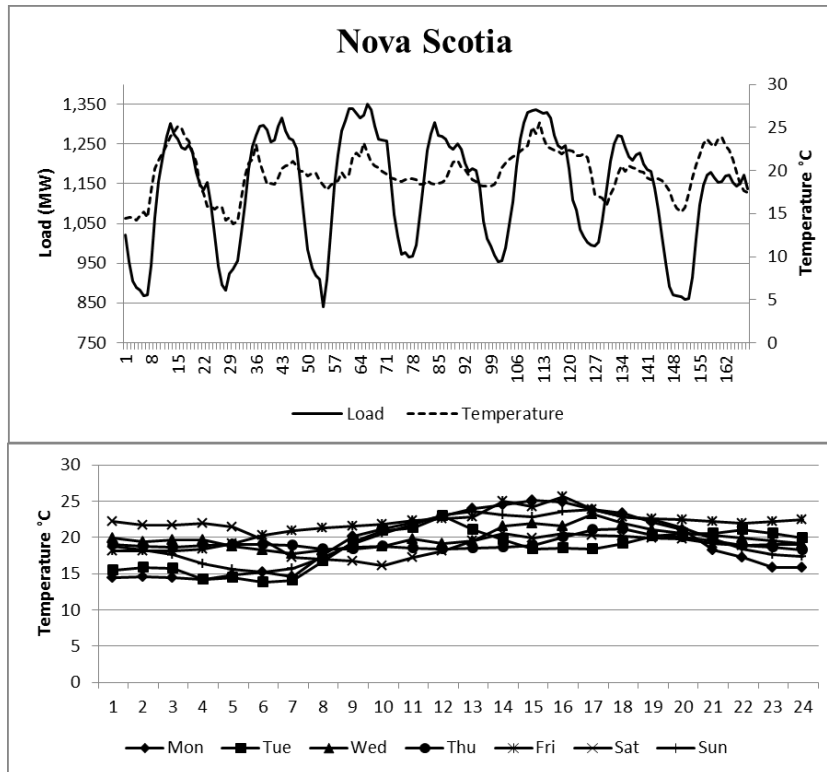


Figure 23: Weekly load and temperature (above), Daily temperature (below) for NS

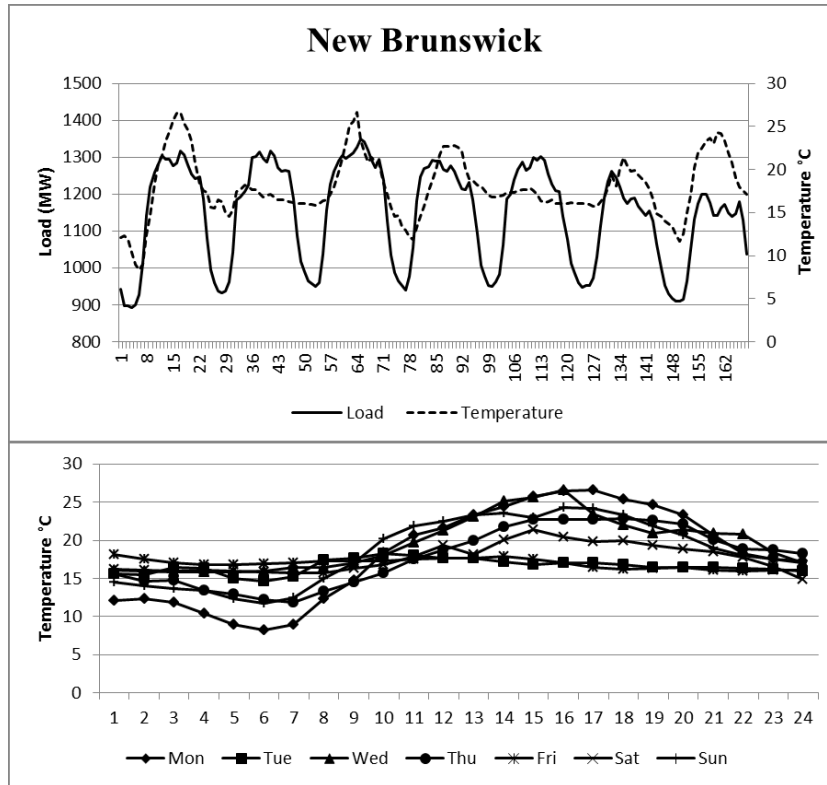


Figure 24: Weekly load and temperature (above), Daily temperature (below) for NB

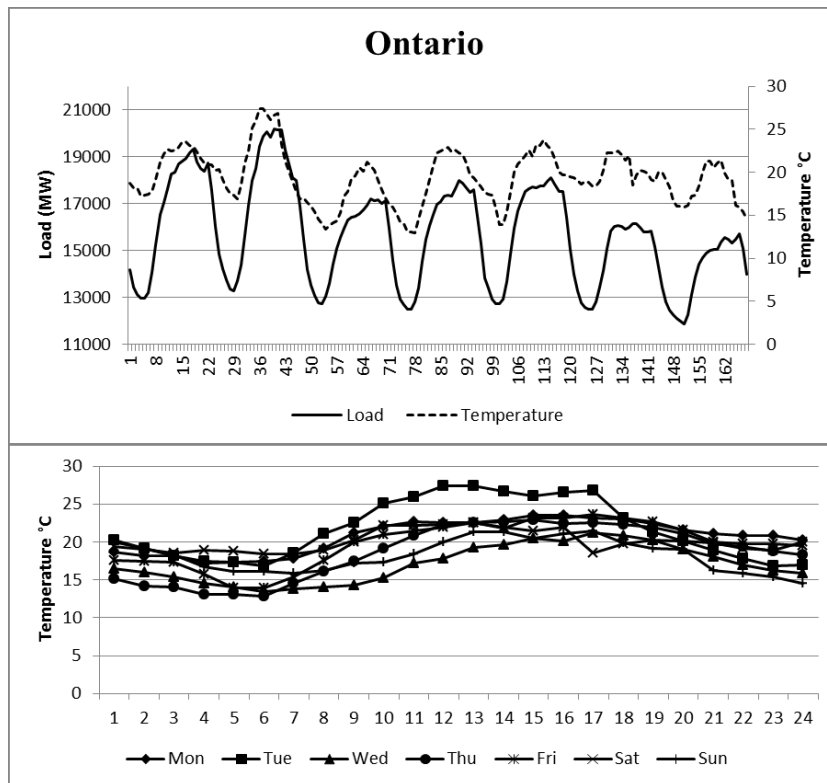


Figure 25: Weekly load and temperature (above), Daily temperature (below) for ON

A close association between variations in temperature and respective change in load can be seen. Also temperatures of previous days are similar to each other. Henceforth, three temperature variables were selected for weekends which include current temperature denoted by $t(h)$, temperature of pervious hour – $t(h-1)$ and temperature of same hour of the previous day – $t(h-24)$, Since the load response of each day of weekend is unique for temperature variables of the same day ($t(h)$ and $t(h-1)$) were selected as inputs. In total, nine inputs for weekdays forecast and six inputs for weekend forecasting were selected. They are summarized in table 2:

Table 2: Summary of selected Inputs for Weekdays and Weekend forecasts

Inputs	Denoted by	Weekdays	Weekends
Previous Hour Load	$y(h-1)$	✓	✓
Previous Day same Hour Load	$y(h-24)$	✓	x
Same Hour Load 2 days before	$y(h-48)$	✓	x
Same Hour Load 3 days before	$y(h-72)$	✓	x
Previous Week same Hour Load	$y(h-168)$	✓	✓
Same Hour Load 2 weeks before	$y(h-336)$	✓	✓
Same Hour Load 3 weeks before	$y(h-504)$	x	✓
Current Temp	$t(h)$	✓	✓
Previous Hour Temp	$t(h-1)$	✓	✓
Previous Day Same Hour Temp	$t(h-24)$	✓	x

4.2 Data Pre-Processing

As mentioned in the previous section the data was separated into weekdays and weekends for separate forecasts. Moreover, all data points were normalized between values of 0 and 1 in order to simplify the calculations and to quantify the effect of all temperature variables. It was done by using the following formula for load and tempter:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

where,

x_{norm} is the normalised data point

x is the actual data point

x_{min} is the minimum data point in the series

x_{max} is the maximum data point in the series

Data normalization was carried out in a separate MATLAB script prior to the separation into weekdays and weekends. Then the input matrices were created in same script respectively.

4.3 Training of Neural Networks

Further separation of data into training and testing datasets was performed. The training dataset consist of data points from June 1 to August 23, 2013 whereas the testing dataset includes data from August 24 to 30, 2013. The neural network architectures which are mentioned in next section were trained using the training dataset and the Levenberg-Marquardt Back Propagation (LMBP) algorithm discussed in section 3.5.4. Each network took different time period to train. The time took for training each network is mentioned in the next chapter.

4.4 Finalised networks used for Forecasting

As discussed in section 3.4, regarding the theoretical background of ANN architectures, this section illustrates the implementation of networks used to perform STLF in this thesis. Four new networks are introduced and three advanced architectures are explored. The results of these seven network architectures are compared with the

most commonly used feed forward neural network (FFNN) with one hidden layer. For weekdays load forecast FFNN is demonstrated in Fig. 26

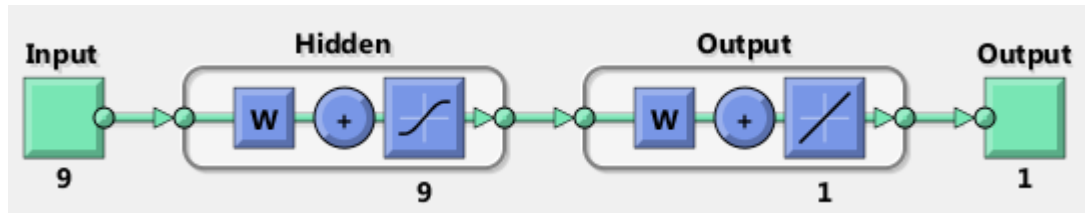


Figure 26: Feed forward Neural Network (FFNN)

For nine inputs nine neurons are used in the hidden layer while one neuron is used in output layer. Linear transfer function is used for the output layer while all prior hidden layers use hyperbolic tangent sigmoid. This result is achieved after trial and error. For fastest convergence equal number of neurons as number of inputs was found to be the best combination. It was also found that with no bias networks generated accurate forecasts therefore the value of bias was set to zero. This section will present network architectures to forecast weekday loads used in this thesis. For weekend forecasts only the number of inputs is changed to six. The performance of each network will be evaluated in the next chapter. Furthermore the Cascaded Neural Network (CNN) architecture with one hidden layer is shown in Fig. 27:

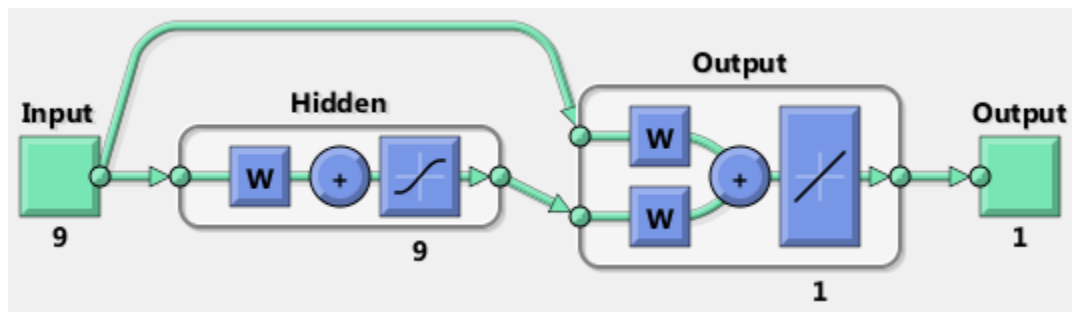


Figure 27: Cascaded Neural Network (CNN)

CNN is considered as advanced architecture which is very seldom used for STLF. In this architecture the input is cascaded to neuron of output layer with weight connected

to it. Moreover, Recurrent Neural Network with Local feedback (RNNL) and Recurrent Neural Network with Global feedback (RNNG) are also explored in this thesis besides CNN. The RNNL architecture consists of a feedback loop for output of hidden layer to its input. This feedback path is called local feedback. For RNNG a weighted feedback is taken from output of output layer to connect from input of hidden layer. Both these networks consist of one hidden layer and are presented in Fig. 28 and 29:

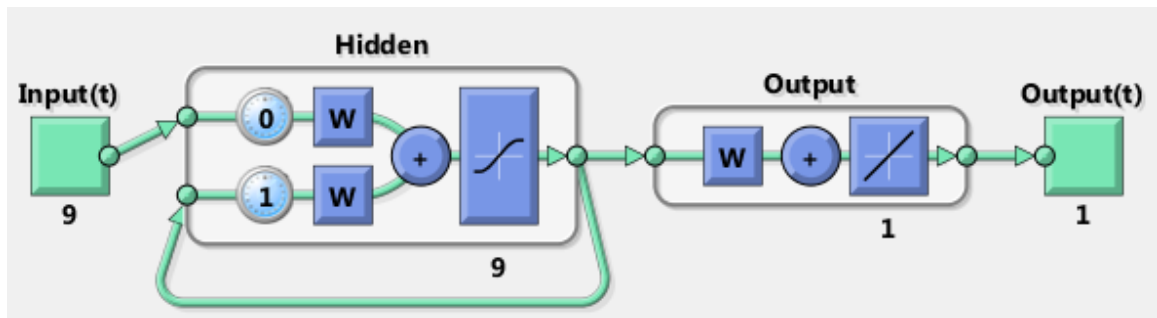


Figure 28: Recurrent Neural Network with Local feedback (RNNL)

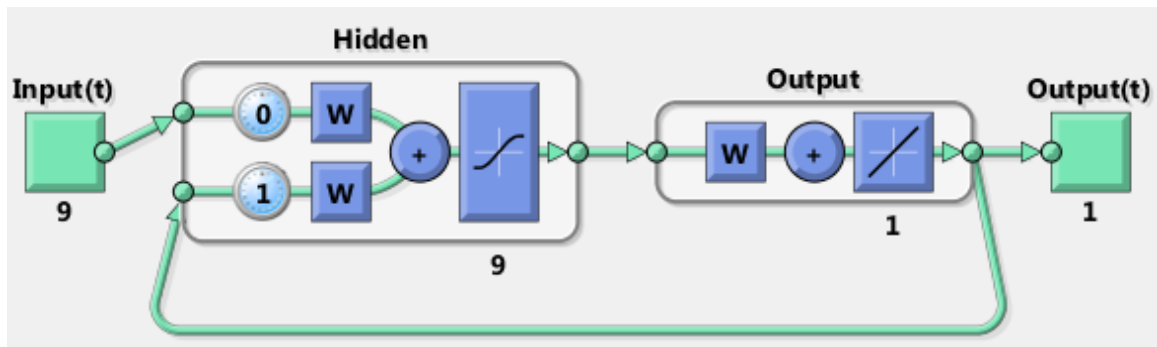


Figure 29: Recurrent Neural Network with Global feedback (RNNG)

Two hybrid ANN architectures are introduced. One combines CNN with RNNL and is named as Cascaded – Recurrent Neural Network with Local feedback (CRNNL). Whereas other combine CNN with RNNG and called Cascaded – Recurrent Neural Network with Global feedback (CRNNG). The architectures of both hybrid networks are given in Fig. 30 and 31:

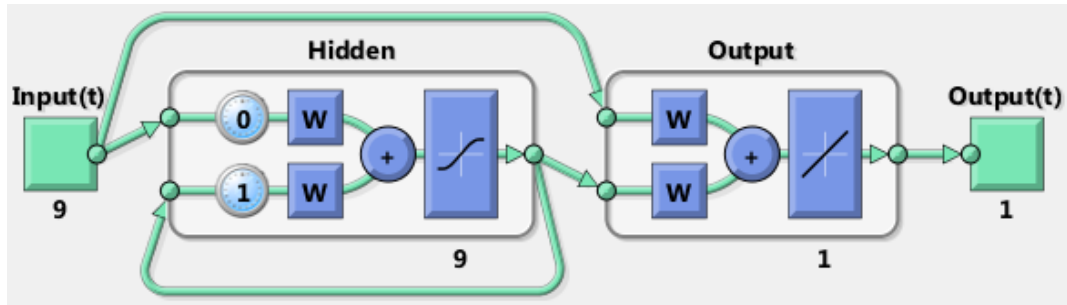


Figure 30: Cascaded – Recurrent Neural Network with Local feedback (CRNNL)

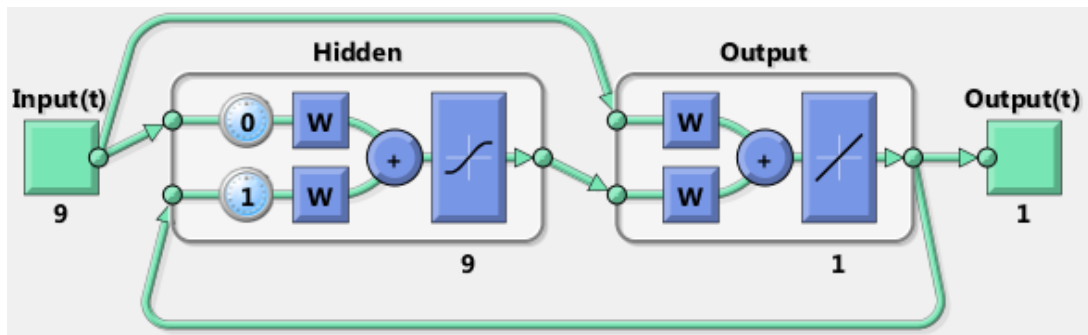


Figure 31: Cascaded – Recurrent Neural Network with Global feedback (CRNNG)

The FFNN and CNN networks are modified and tested with an increased hidden layer. The 3-layered FFNN denoted by FFNN (3L) consist of 2 hidden layers with number of neurons in first layer equal to number of inputs. The number of neurons in second hidden layer was kept double the number of inputs as it was found to produce accurate results. 3-layered CNN or CNN (3L) is much like FFNN (3L) but input is cascaded into both hidden and output layers with connection weights. The output of first hidden layer is cascaded to output layers as well with a weight connection of its own. Both network architectures are demonstrated as shown in Fig. 32 and 33.:

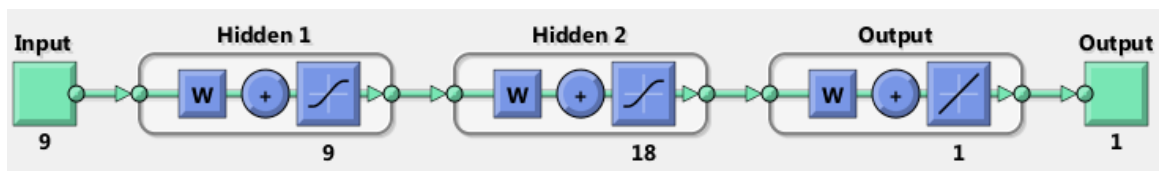


Figure 32: 3-layered feed forward Neural Network (FFNN (3L))

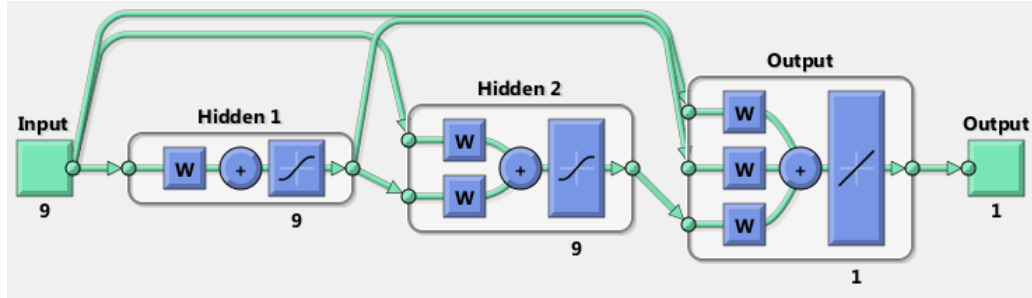


Figure 33: 3-layered cascaded Neural Network (CNN (3L))

4.5 Error Analysis and Performance Evaluation

After getting forecasting results two parameters were used to compare the results and evaluate the advantages and disadvantages of each network. Error analysis was done on the basis of Mean Absolute Percentage Error (MAPE). Most authors in literature use this parameter as a measure of performance of a forecasting method. Another parameter calculated for error analysis is the Root Mean Squared Error (RMSE). The choice between MAPE and other error criteria depends on the error statistics which is system dependent. Following formulae were used to calculate both error parameters:

$$MAPE = \frac{100}{N} \sum_{i=1}^N \frac{|y_{predicted} - y_{actual}|}{y_{actual}} \quad (4.2)$$

$$RMSE = \sqrt{\frac{(y_{predicted} - y_{actual})^2}{N}} \quad (4.3)$$

N is the total number of points forecasted.

4.6 Use of Software

Microsoft Excel was used to generate all the graphs in this Thesis. The figures of neural network architectures were produced using the MATLAB's neural network toolbox and training of networks, forecasting results as well as data processing steps were performed using MATLAB script in conjunction neural networks toolbox.

Chapter 5: Results and Discussion

This chapter is divided into four parts. The first three parts present forecasting results of three Canadian jurisdictions namely; Nova Scotia, New Brunswick and Ontario for weekdays and weekends using eight different neural network architectures introduced in section 4.4. Graphs of actual versus predicted values are followed by a table summarizing and discussing results. Last part of this chapter brings up all results together and discusses the observed outcome achieved.

5.1 Nova Scotia

The load of Nova Scotia ranges from 745 MW to 1373MW during selected time period. Though weekdays and weekend data is separated but since the province is not much industrialized, load curves throughout the week remain consistent. Load and temperature data from June 1 to August 25, 2013 is used for training all networks tested. This data was gathered from online resources published by the Government of Canada, climate database [64] and Nova Scotia power, Oasis Database [65]. The 24 hour ahead and week ahead forecasts are presented next for weekdays followed by weekend's forecasts.

5.1.1 24 hours ahead forecast for Weekday

The 24 hours ahead NS load forecast using eight network architectures are given for a weekday. The day selected for forecast is August 26, 2013. It is the last Monday of the month. The architectures of networks tested are discussed in section 4.4. The following figures show a comparison between actual and forecasted load as well as the variations in forecasting error throughout the day for each network implemented.

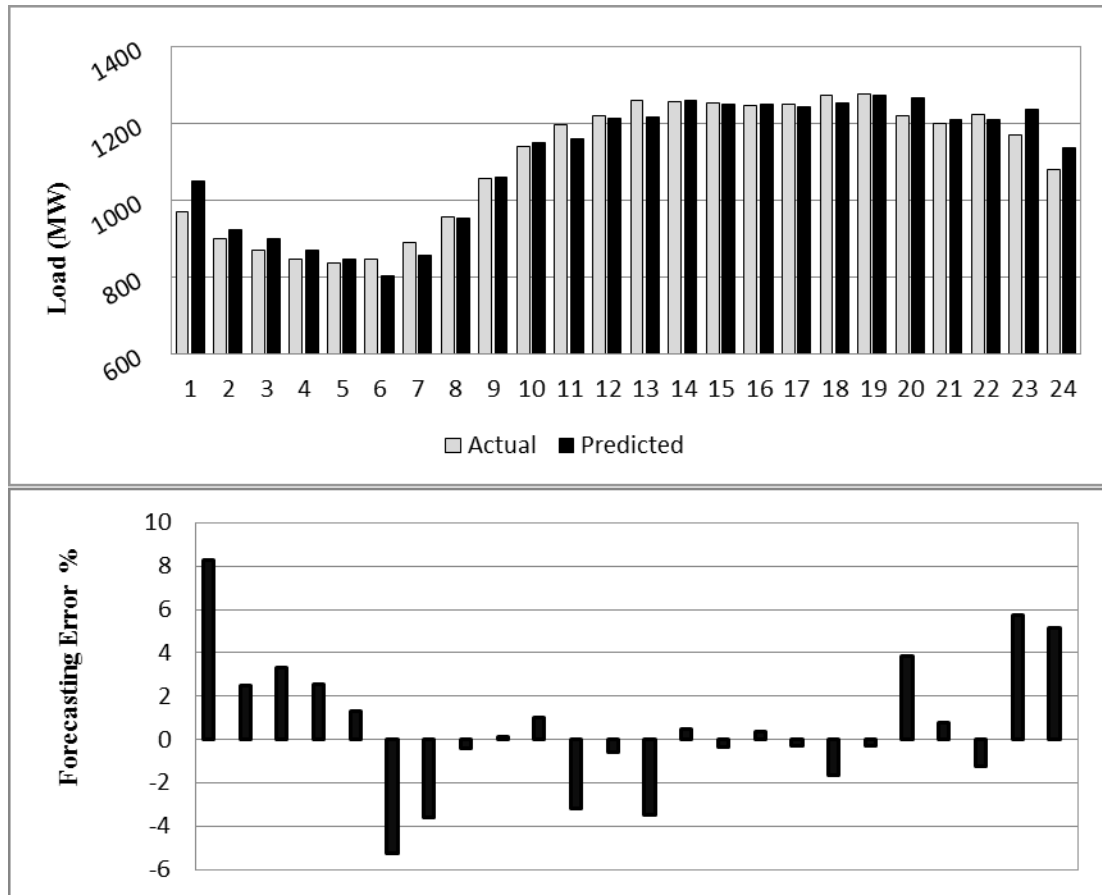


Figure 34: Day ahead forecast results for FFNN

Fig. 34 shows the actual and predicted loads for FFNN. The forecasting error ranges from 8.26% to -5.24%. Both these errors occur during first 6 hours of the day. Once load curve becomes steady the forecasting error is also reduced to as low as 0.13%. It can also be observed that as the load starts decreasing towards end of the day, the error starts increasing. For the last 2 hours error increases above 5%. The training process for this architecture took 5 seconds to generate results. The MAPE of this network configuration was 2.32%.

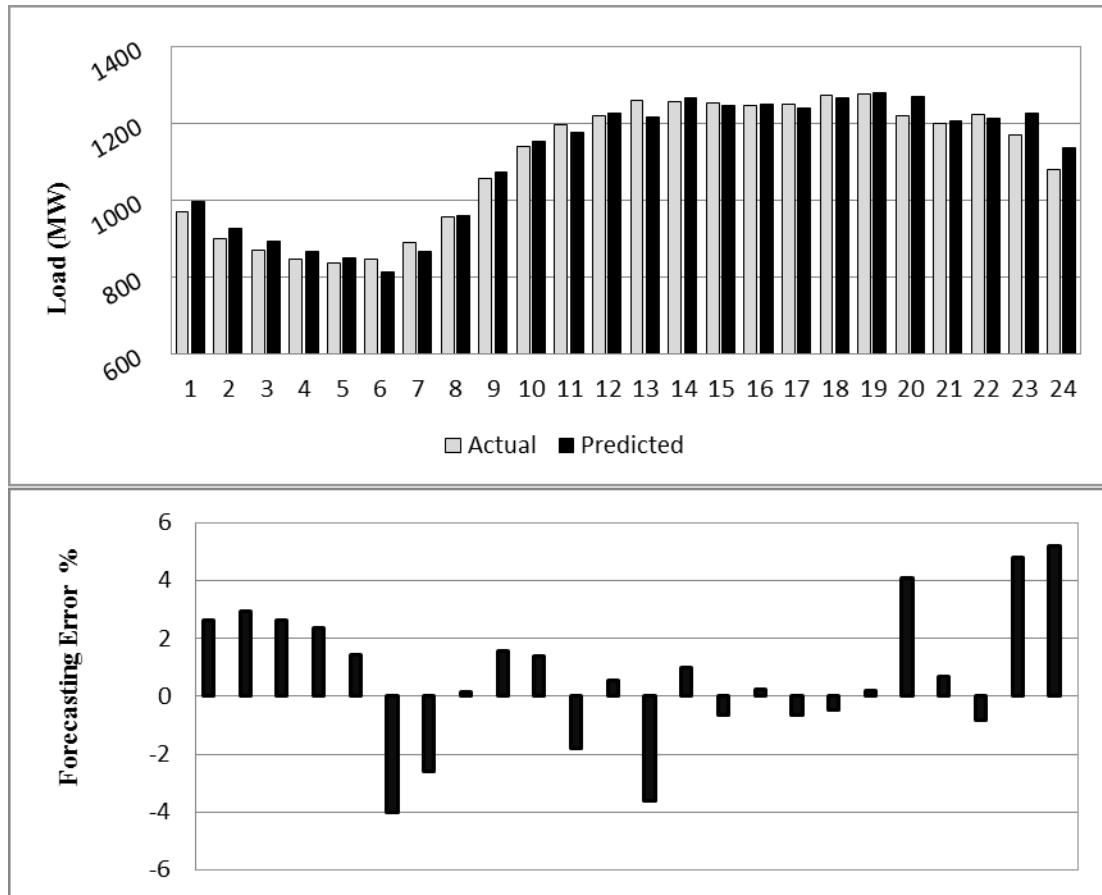


Figure 35: Day ahead forecast results for CNN

The forecasting error for the CNN ranges from 5.19% to -4.03%. The CNN forecast starts with the error around 2% for the first 5 hours but a negative spike of -4.03 occurs at the 6th hour. The forecast is quite stable throughout except during the last 4 hours. The error increases to its maximum during the last hour. As compared to the FFNN the overall error distribution seems to be more stable. The training took 6 seconds and the curve for predicted load follows the actual load consistently except during the last 4 hours. The MAPE was also improved to 1.93% as compared to the FFNN.

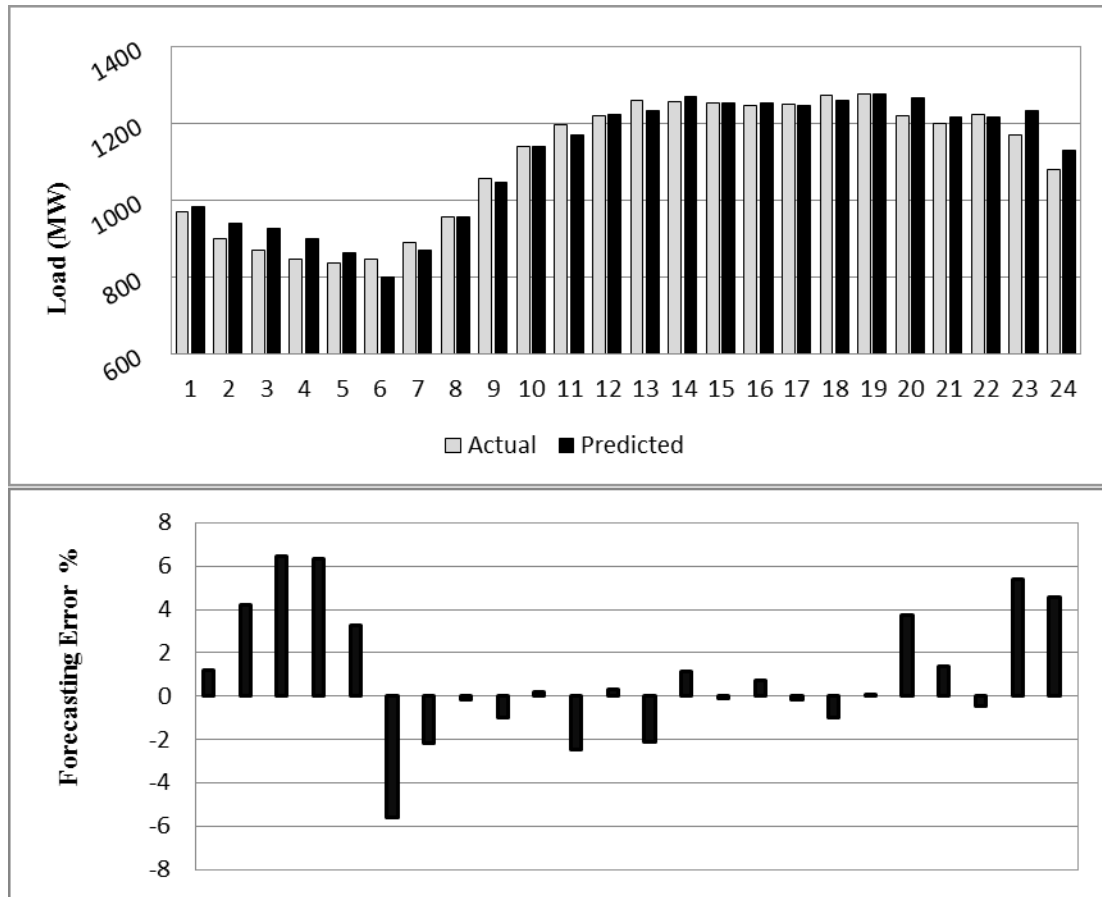


Figure 36: Day ahead forecast results for RNNL

The forecasting results for RNNL are shown in Fig. 36. Since the network contains a delay in the feedback path, training took time to produce forecasts. The total training time was 55 seconds long. Forecasting error for first hour was low (2.18%) but the maximum error peak occurs at the 3rd hour with a value of 6.45%. A negative error peak of -5.6% occurs at the 6th hour. The performance afterwards seems to be quite accurate and the error floats between -1% and 1%. The error performance during peak hours is very good and the error falls to its minimum of 0.048% during the 19th hour. The error increases again during the last 4 hours. Due to high peaks during start and end, MAPE for this network goes to 2.26% but the actual versus forecasted curves during the peak hours matches very closely with each other.

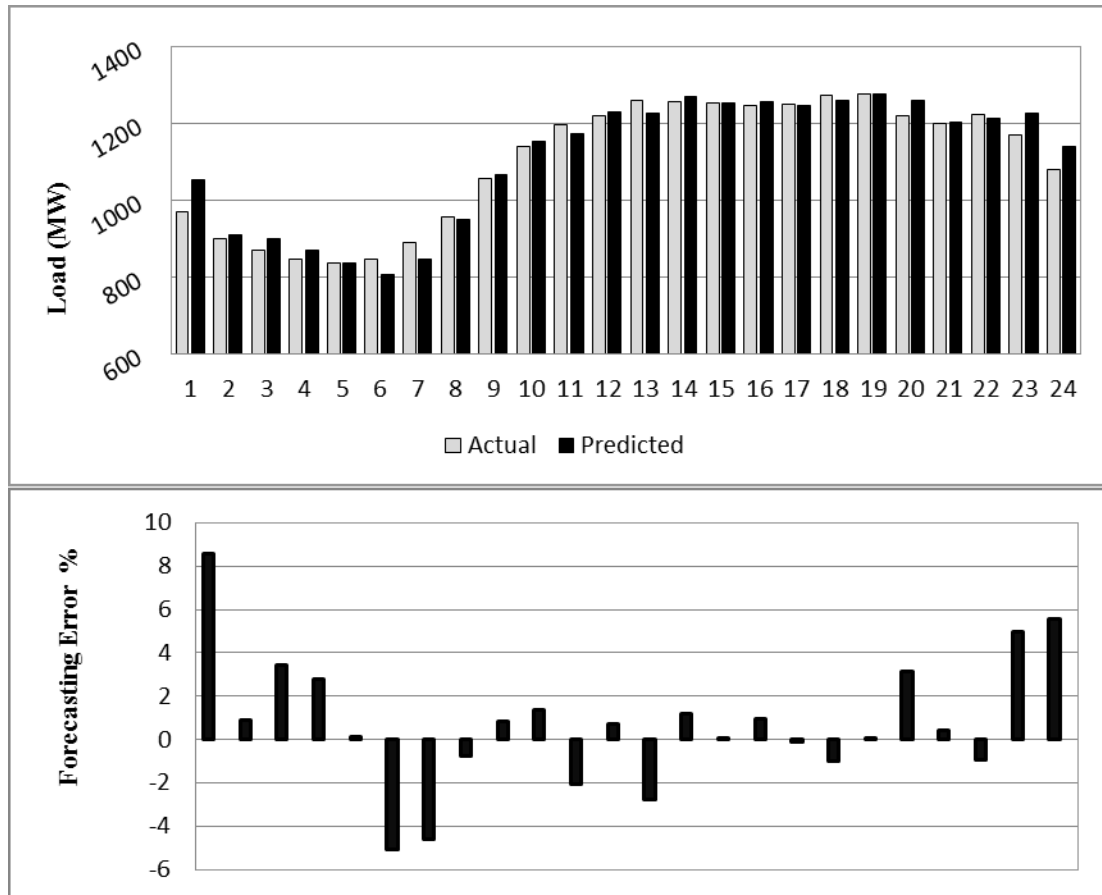


Figure 37: Day ahead forecast results for RNNG

The performance of RNNG has more resemblance to that of the FFNN. The training time of network is 63 seconds. It starts with a high forecasting error of 8.55% and negative peak goes to -5.06%. Just like the FFNN both these error peaks occur during first 6 hours. After being steady during peak hours forecast ends with errors around 5% during last 2 hours. Because a few error peaks occurs during the forecast MAPE and drops down to 2.18% as compared to that of the FFNN.

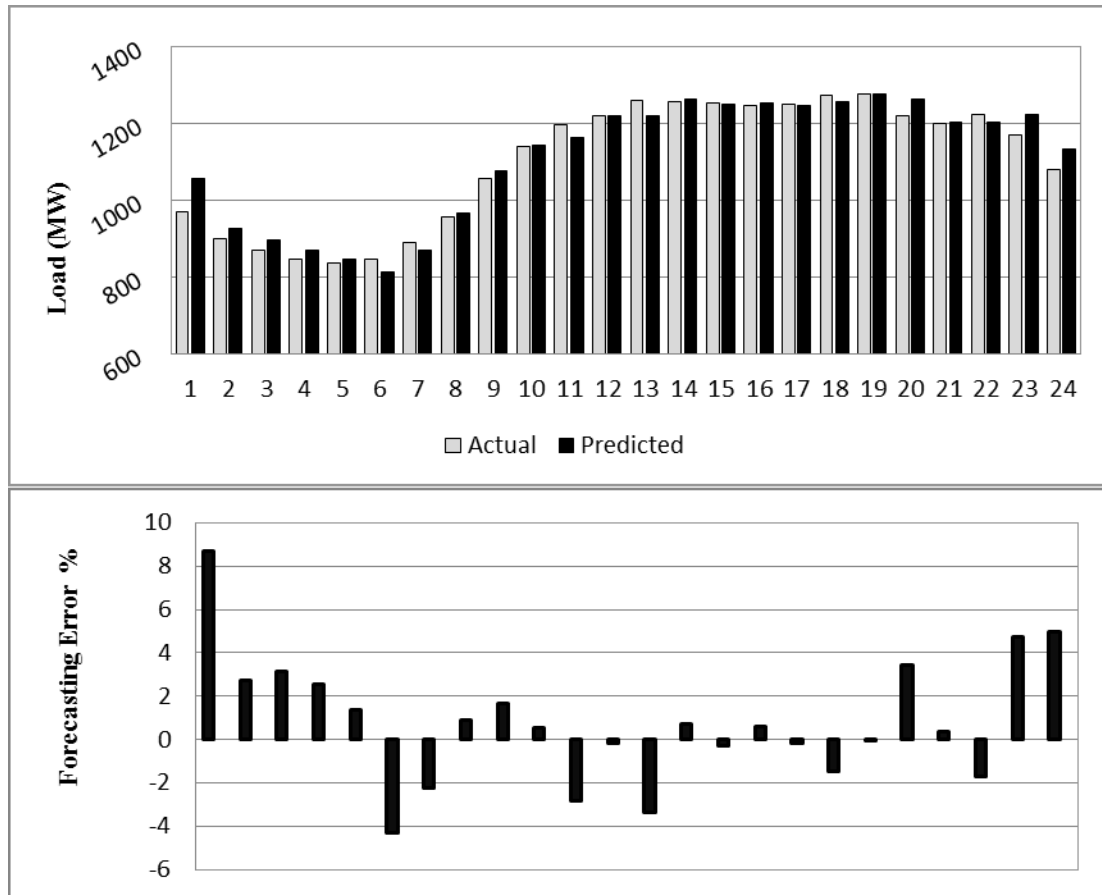


Figure 38: Day ahead forecast results for CRNNL

The forecasts of the hybrid network architecture CRNNL is shown in Figure 38. The MAPE drops down to 2.21% as compared to FFNN (2.32%). The training time of around 58 second is notable. It is like a trade-off between accuracy and time. For short term load forecasting this seems to be acceptable. The response is almost similar to FFNN with a starting error of 8.6% and a negative peak of -4.3% starting in 6 hours. The error rises towards the ending hours just like other networks discussed. In total there are 6 instances where error goes above 3% both on positive and negative sides as compared to 8 times for FFNN.

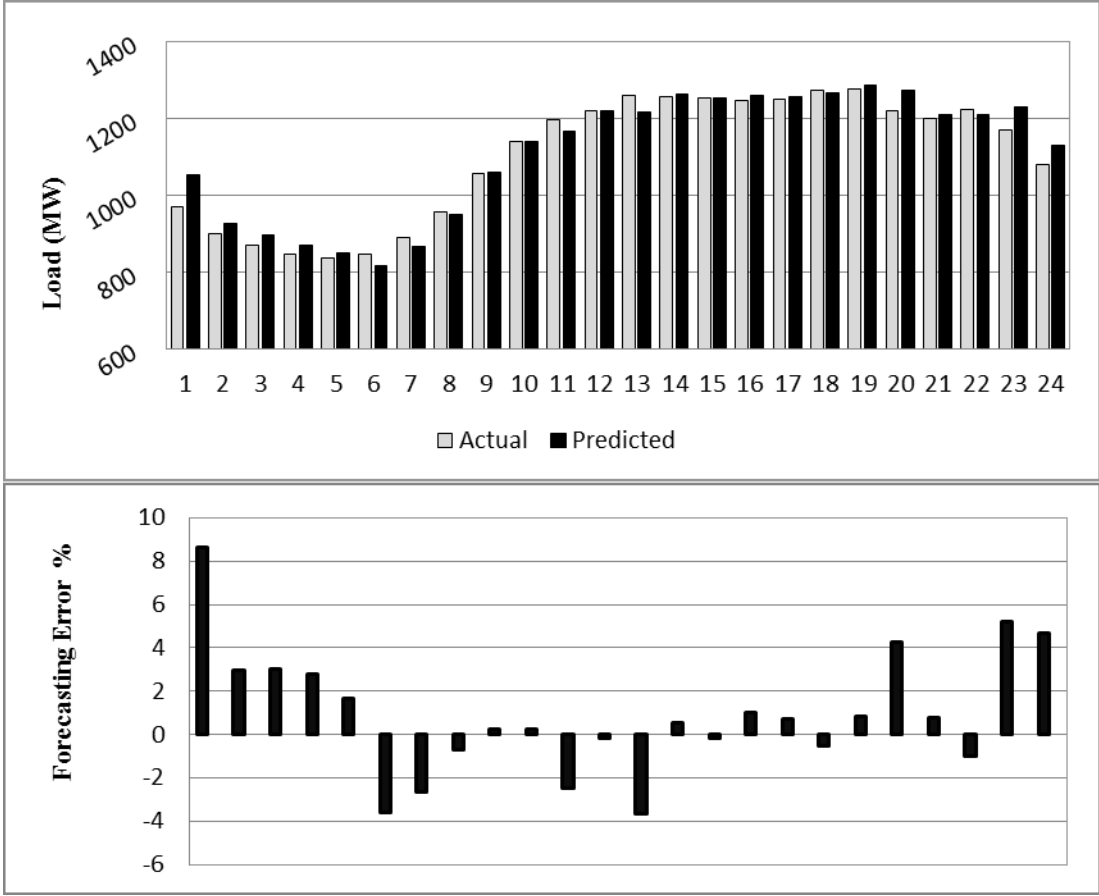


Figure 39: Day ahead forecast results for CRNNG

The CRNNG produces improved results than CRNNL with slightly less time for training. Network trains in approximately 49 seconds and gives an overall MAPE of 2.18% which is similar to the performance of the RNNG network. The forecasting error ranges from -3.6% to 8.6% which is a better outcome as compared to maximum negative error of -5.06% of RNNG and -4.03 of CNN. The error curve during peak hours is steady like other networks and towards negative error side is suppressed more than other networks.

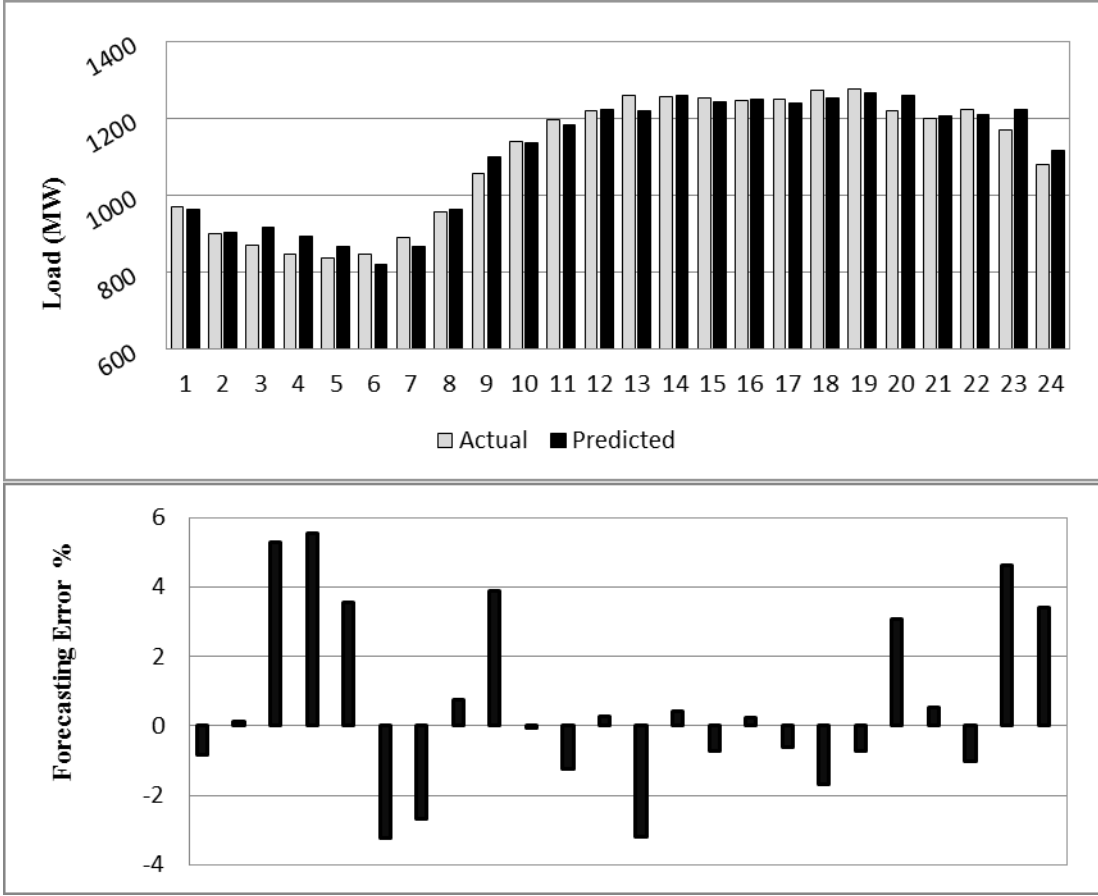


Figure 40: Day ahead forecast results for 3-layered FFNN

The performance of 3-layered FFNN presented in Figure 40 look much alike RNNL with low error to start with. The error response looks quite erratic but maximum positive and negative errors are between 5.53% and -3.32. The minimum error is 0.04%. The convergence time is 6 seconds with MAPE of 1.99%. Though the overall performance looks better than most of the networks but from error curve it is observed that the forecast performance during peak hours produces some error spike.

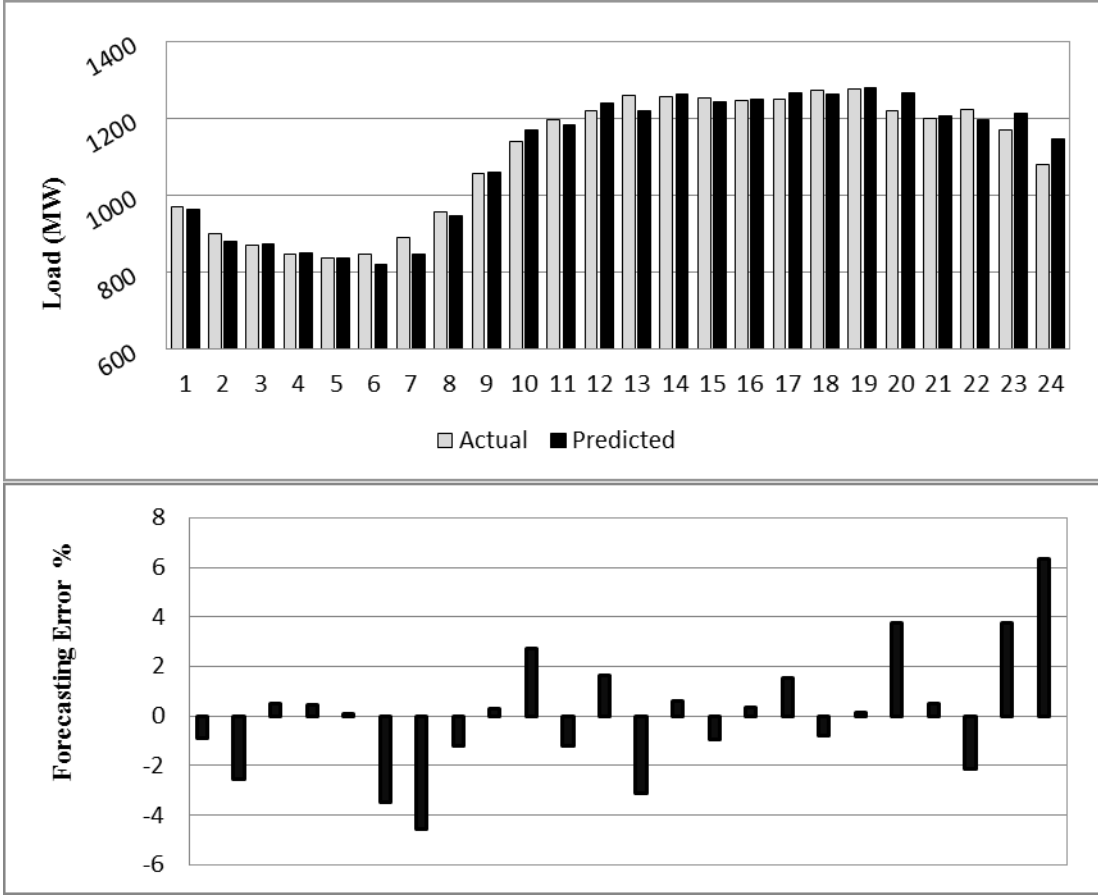


Figure 41: Day ahead forecast results for 3-layered CNN

The forecast generated by the 3-layered CNN is shown in Figure 41. By far it gives the best performance with a MAPE of 1.81. The error ranges from -4.54% to 6.33%. There are 6 instances where the error goes above 3% on either side but throughout it remains close to 0. The error remains between positive and negative 4% except for the last hour prediction. If we add the fast training period of 7 seconds, 3-layered CNN by far proves to be the best among all eight networks for 24 hour ahead forecasting of weekday for Nova Scotia.

5.1.2 Comparative Analysis of 24 hours ahead forecast for Weekday

The performance of all eight networks has been discussed in previous section. Now we will summarize and give a comparative analysis. Table 3 shows the MAPE and RMSE of each network tested for forecasting the Nova Scotia load for August 26, 2013. The improvement of each network over worst performing network which in this case is FFNN is also presented in the table.

Table 3: 24 hours ahead MAPE and RMSE for Nova Scotia

Architecture	MAPE	RMSE	Improvement
<u>FFNN</u>	<u>2.32</u>	<u>32.66</u>	<u>0.00%</u>
CNN	1.93	26.28	16.68%
RNNL	2.26	30.29	2.80%
RNNG	2.18	31.23	5.86%
CRNNL	2.21	30.92	4.76%
CRNNG	2.18	31.20	5.86%
FFNN (3L)	1.99	26.48	14.42%
<u>CNN (3L)</u>	<u>1.81</u>	<u>26.23</u>	<u>21.88%</u>

The results indicate that the 3-layered CNN gives best overall performance with 21.88% improvement over the FFNN. The performance of CNN, RNNL, RNNG, CRNNL and CRNNG with enhancements of 16.68%, 2.8%, 5.86%, 4.76% and 5.86% have proved to be more stable during peak hours than 3-layered FFNN or CNN network. On the basis of consistency if the forecasting errors during starting and ending hours are acceptable to a utility company than CNN can be the best choice among all. Also networks with recurrent paths takes longer time to converge but gives better error response and closely matches the load curve during peak hours. Following is a graph giving a comparison between worst and best forecasting error performance; FFNN and CNN (3L)

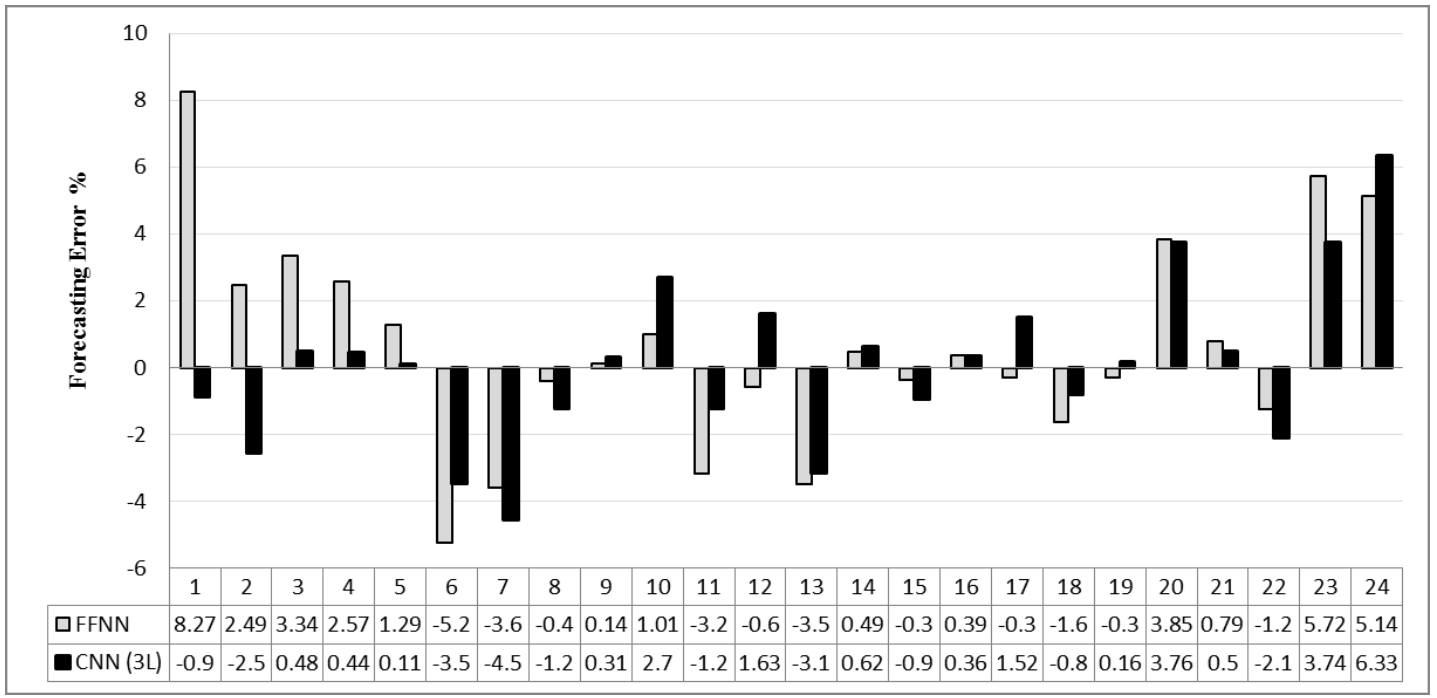
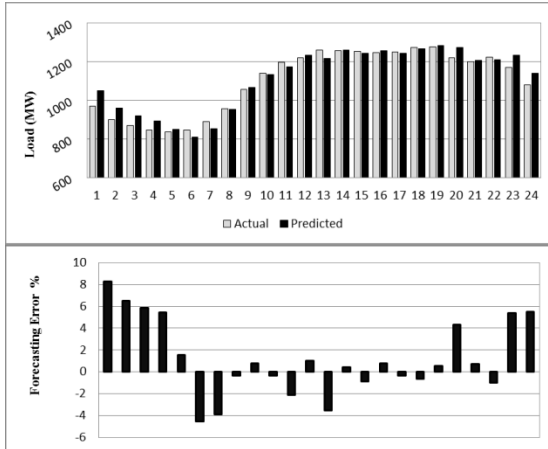


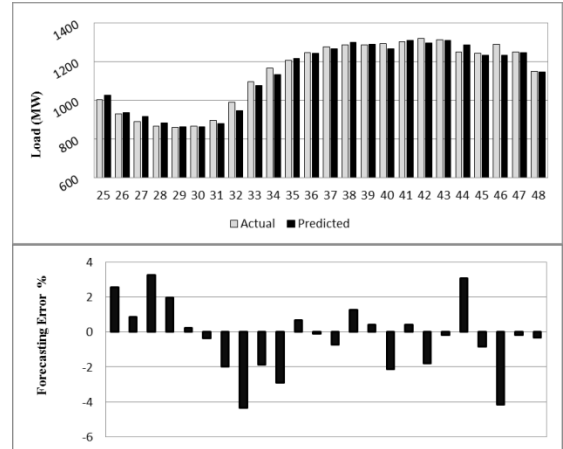
Figure 42: 24 hours ahead forecasting error performance comparison between FFNN and CNN (3L) for Nova Scotia

5.1.3 Week ahead forecast for Weekdays

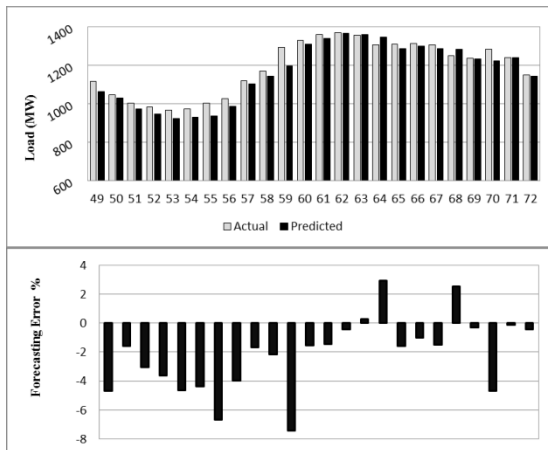
The week ahead NS load forecast for weekdays is discussed in this section. The 5-day week selected to forecast load is from August 26 to 30. These are days from Monday and Friday. The eight networks discussed in section 4.4 are tested and their results explained as follows:



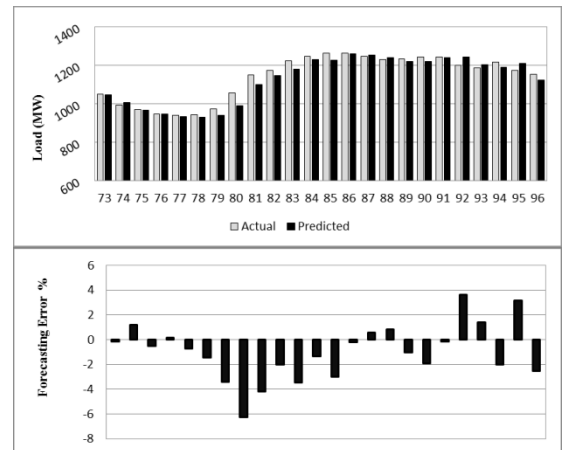
(a) Monday



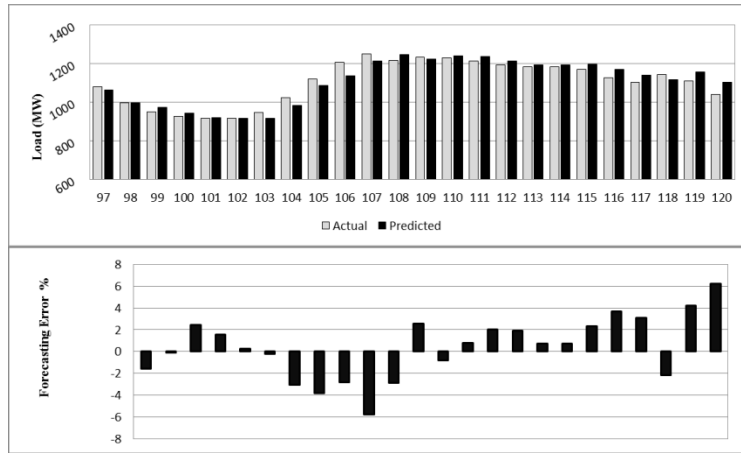
(b) Tuesday



(c) Wednesday



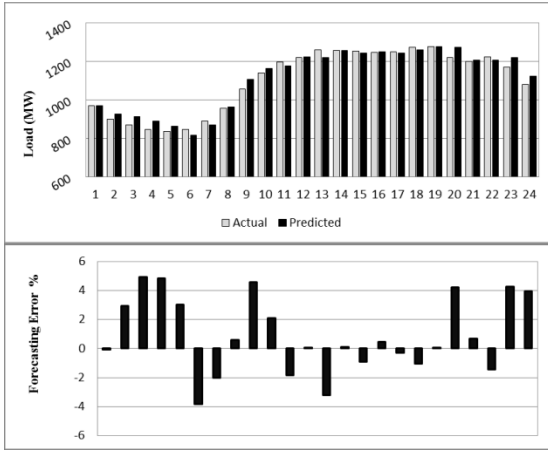
(d) Thursday



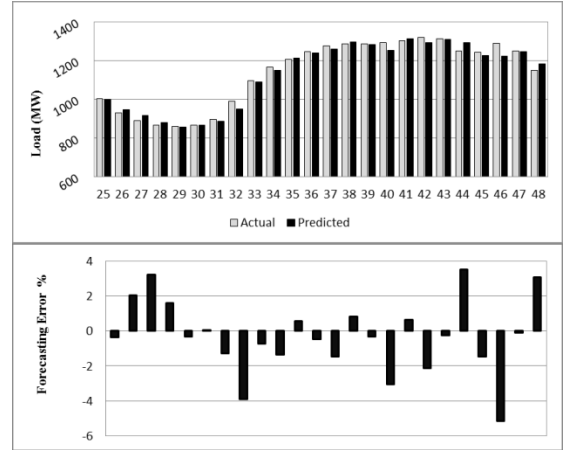
(e) Friday

Figure 43: Week ahead forecast results for FFNN

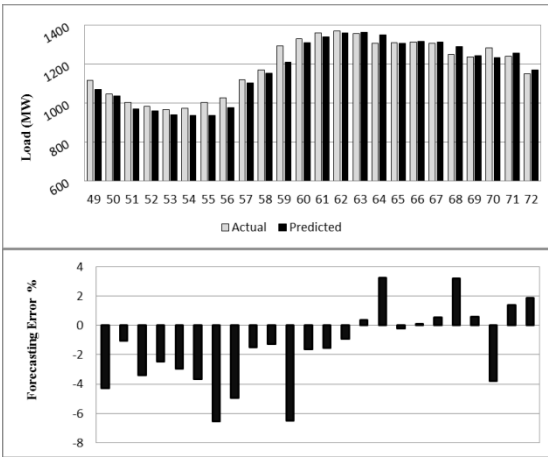
FFNN performs better for longer period of forecast and produce a forecast for week ahead for the days specified with MAPE 2.21. The overall forecasting result may seem reduced but error range is increased. The error varies in between -7.45% to 8.26% with minimum error of 0.07%. The error rises during start and ending hours of a day with most error points towards negative side of the axis.



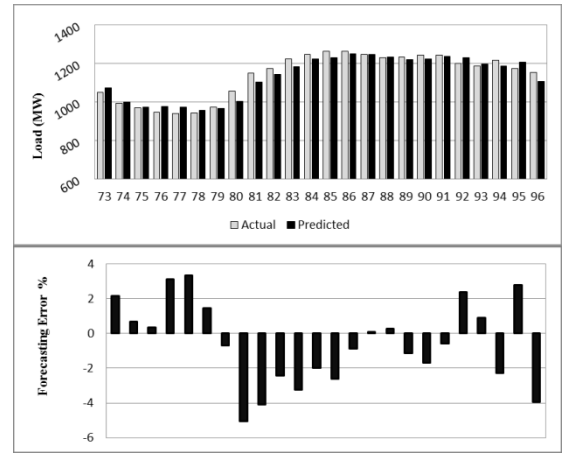
(a) Monday



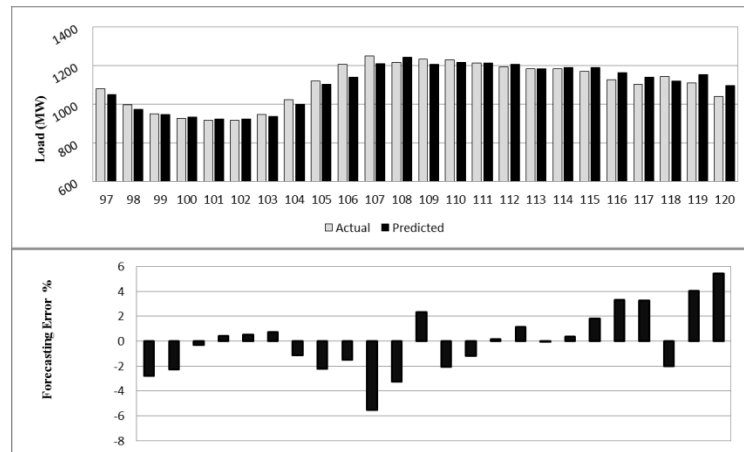
(b) Tuesday



(c) Wednesday



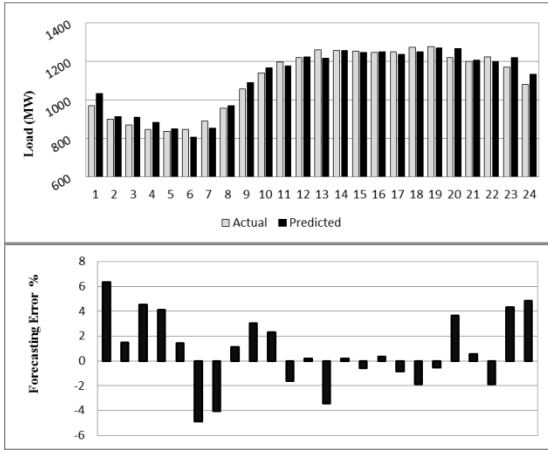
(d) Thursday



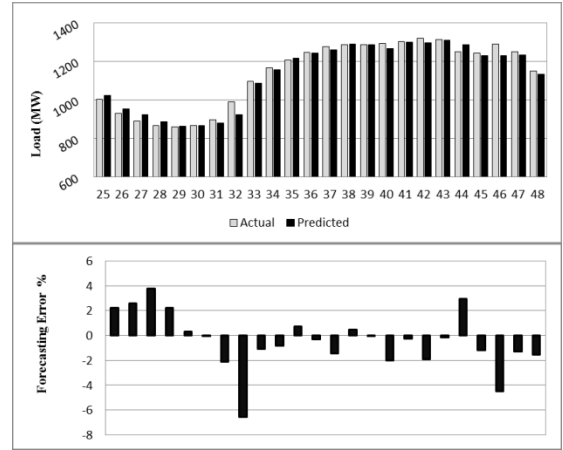
(e) Friday

Figure 44: Week ahead forecast results for CNN

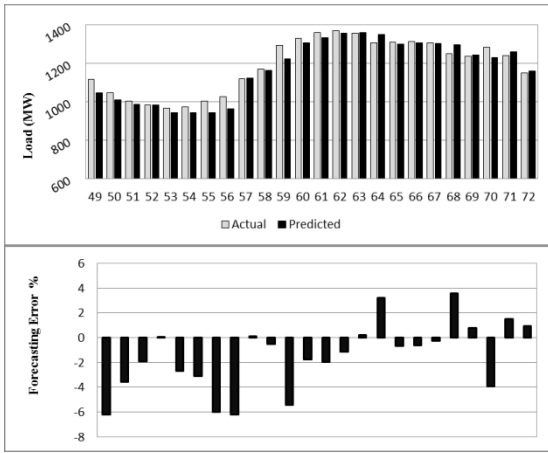
CNN produces improved results as compared to FFNN. The MAPE drops down to 2.03 and the error is found in between -6.56% and 5.45%. The large negative error peak occurring during second and third day (Tuesday and Wednesday) have been reduced. Starting error is also reduced and error curve looks steadier than FFNN. The rise in error seen during starting and ending hours of each day has also improved.



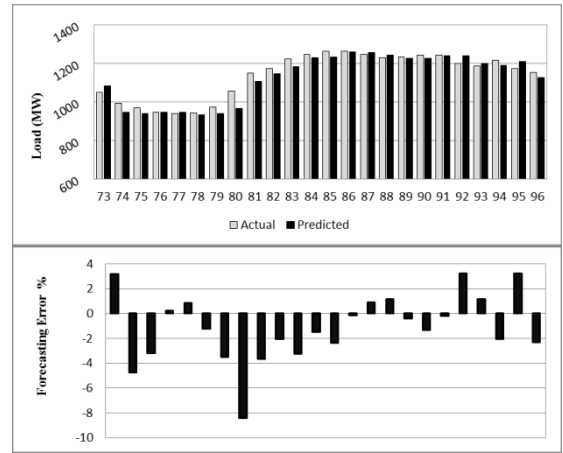
(a) Monday



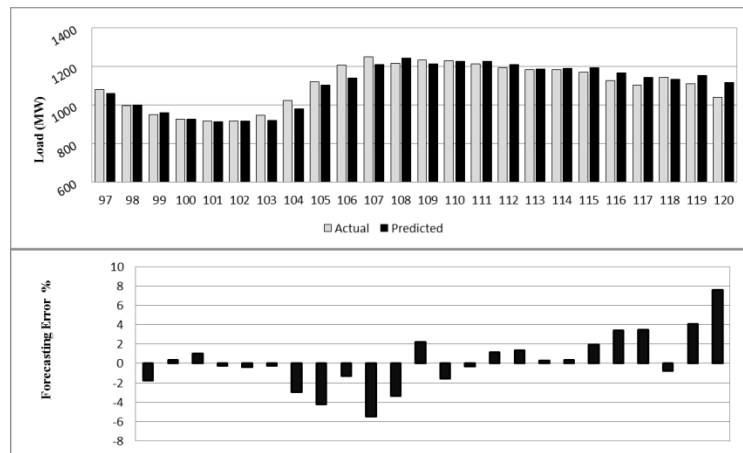
(b) Tuesday



(c) Wednesday



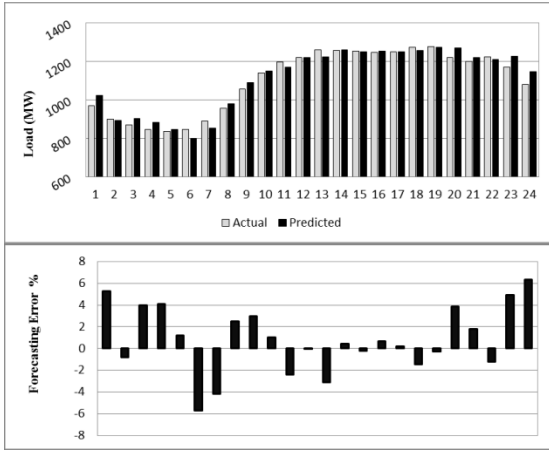
(d) Thursday



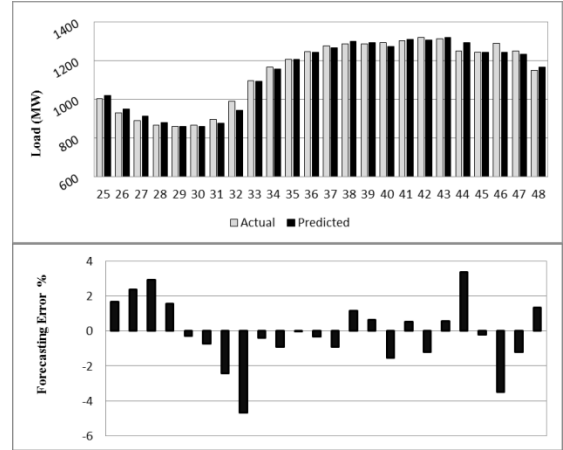
(e) Friday

Figure 45: Week ahead forecast results for RNNL

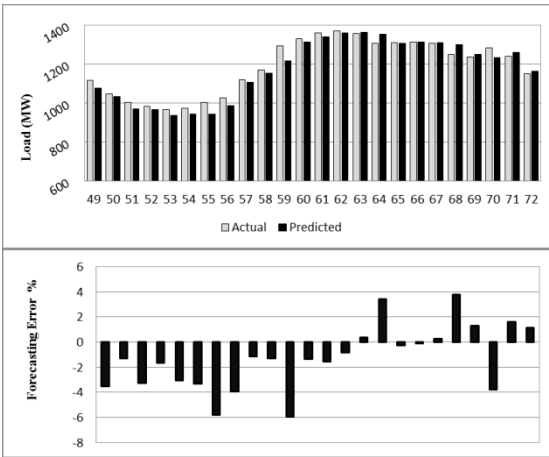
RNNL produces almost same results as FFNN. With a MAPE of 2.16 the performance is also not much improved. Considering time it takes to train a recurrent network the overall performance of FFNN can be regarded to be better. During peak hours the predicted power curve closely follows the actual curve. Also the error during the starting hours of third and fourth days is greatly reduced. The reason for a reduced overall error performance is the high errors occurring throughout the forecast ranging from -8.4% to 7.61%.



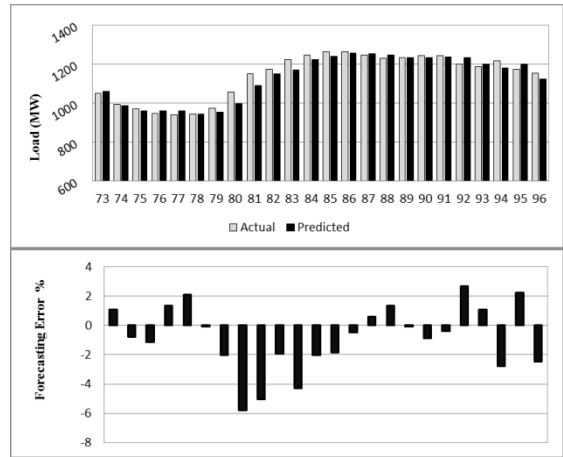
(a) Monday



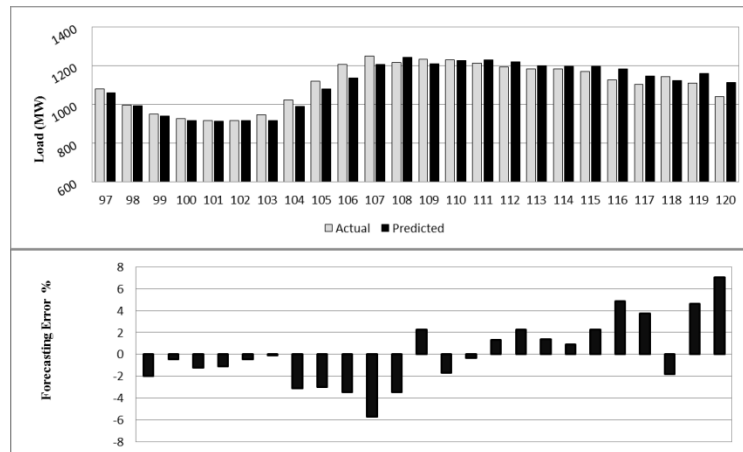
(b) Tuesday



(c) Wednesday



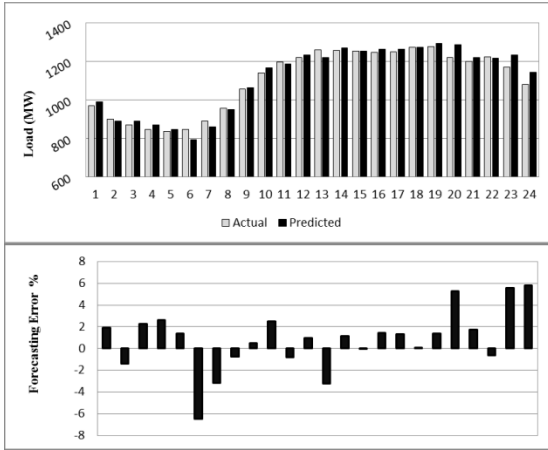
(d) Thursday



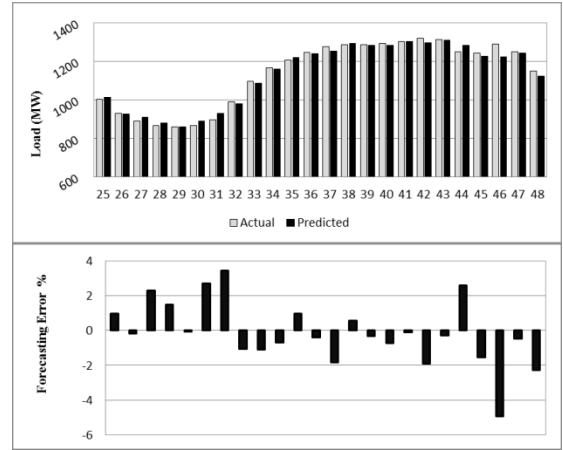
(e) Friday

Figure 46: Week ahead forecast results for RNNG

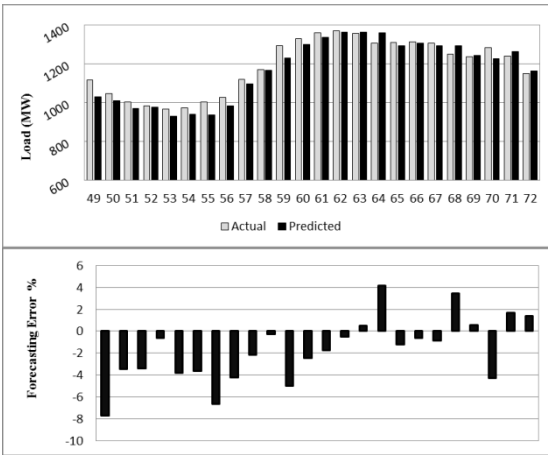
With the forecasting error ranging from -5.94% to 7.06%, RNNG produces a better result with a MAPE of 2.08%. The predicted load follows the actual load curve throughout with a total of only 10 spikes of error 5% above and below the axis. The distribution of error is consistent through the days and starting and ending hours do not result in increasing error.



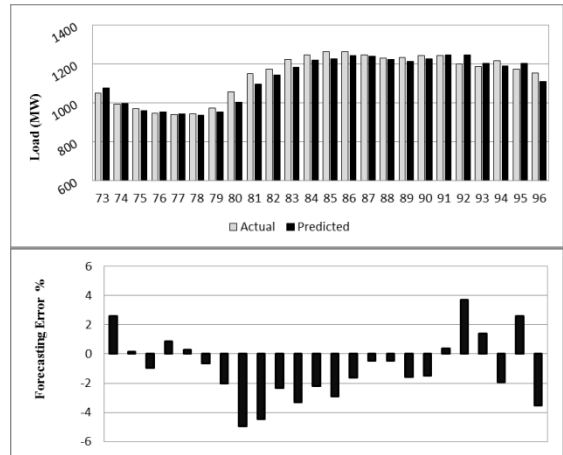
(a) Monday



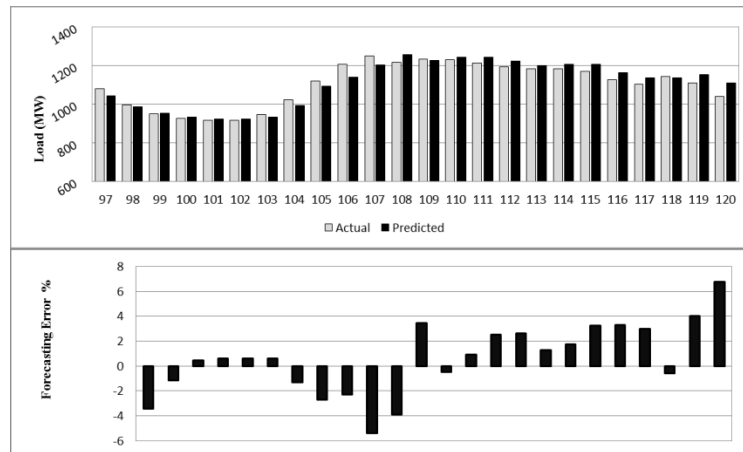
(b) Tuesday



(c) Wednesday



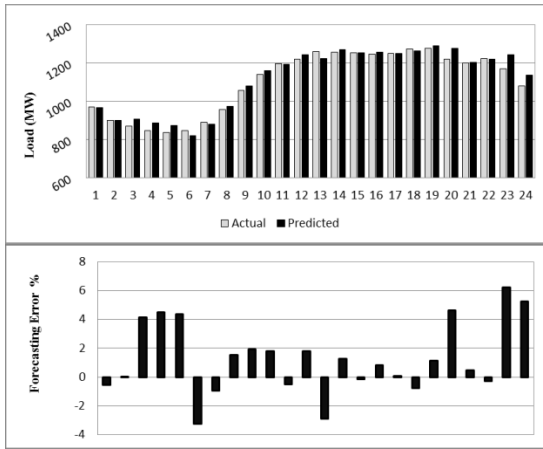
(d) Thursday



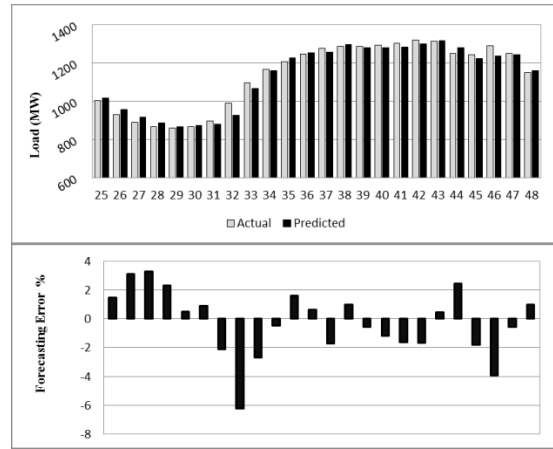
(e) Friday

Figure 47: Week ahead forecast results for CRNNL

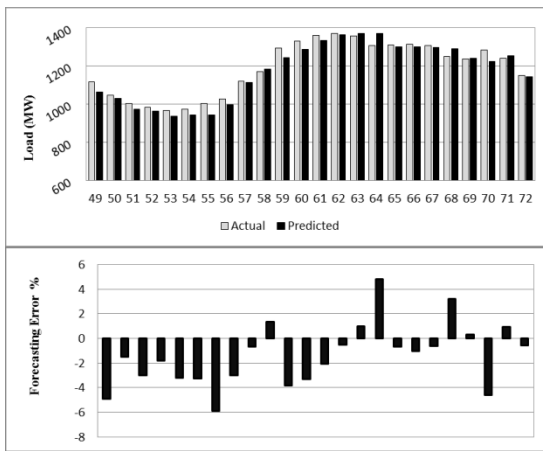
Fig. 47 shows the forecast made by using hybrid CRNNL network. The result generates better forecasts as compared to CNNL network. It is seen that cascaded connection in architecture have improved the outcome. The error ranges from -7.71% to 6.74% with a MAPE of 2.1%. The negative error increases greatly during forecast of second and fourth day but do not affect the overall performance.



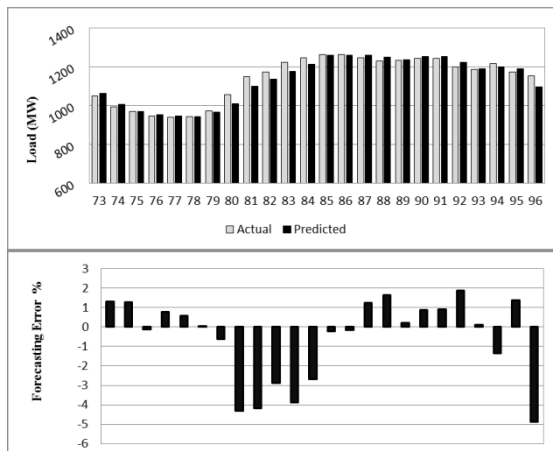
(a) Monday



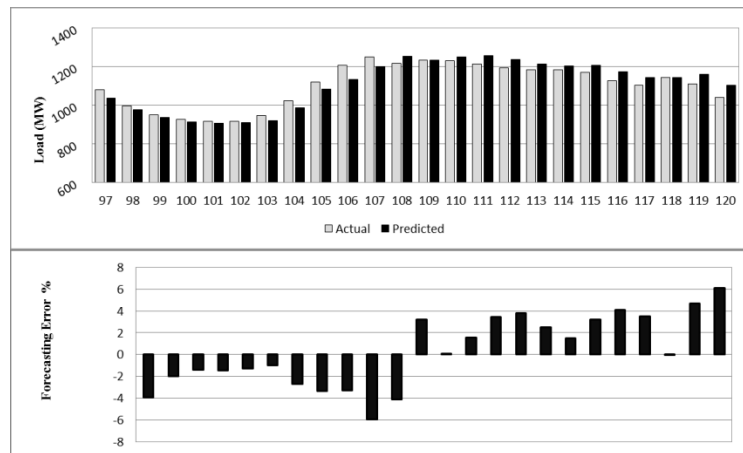
(b) Tuesday



(c) Wednesday



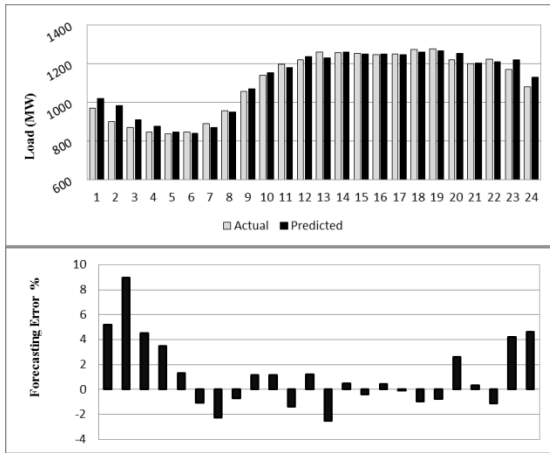
(d) Thursday



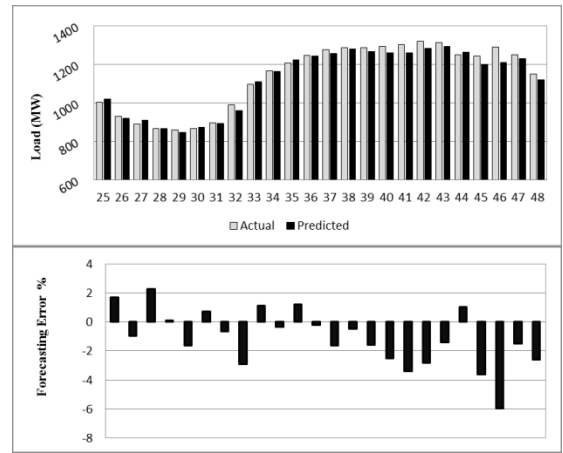
(e) Friday

Figure 48: Week ahead forecast results for CRNG

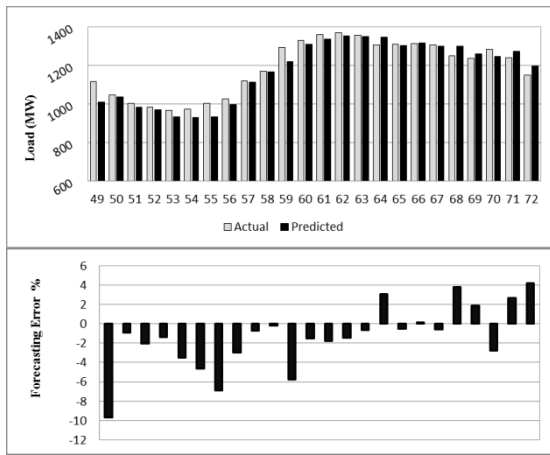
The CRNNG produces steady error distribution. Only 8 times error curve crosses 5%. The MAPE is 2.12 and error range -6.23 to 6.21%. The result generated by this network is most consistent as compared to the network performance described before this. The error increases during starting and ending hours and the forecast of the last 36 hours is very inaccurate contributing a great deal in increasing the overall error.



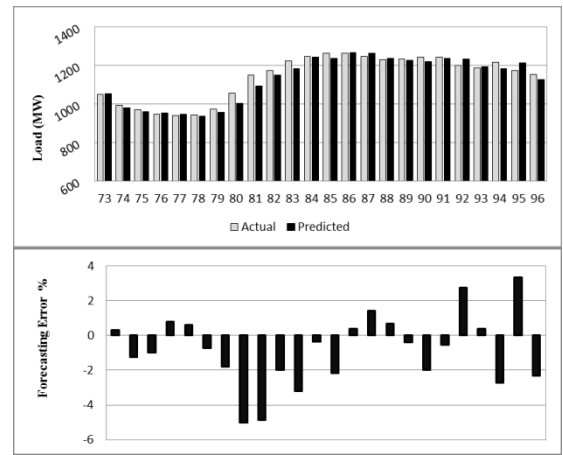
(a) Monday



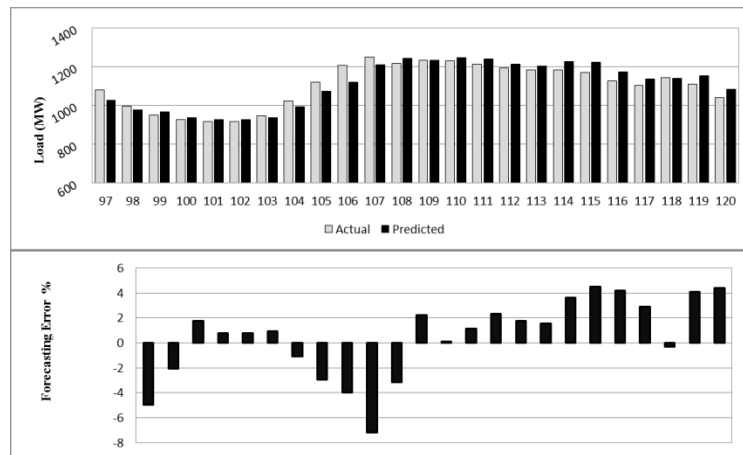
(b) Tuesday



(c) Wednesday



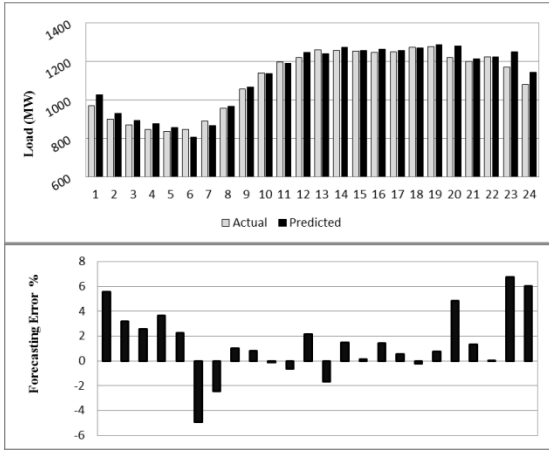
(d) Thursday



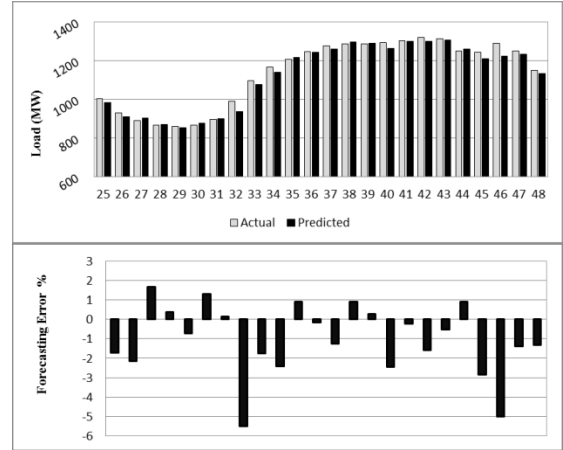
(e) Friday

Figure 49: Week ahead forecast results for 3-layered FFNN

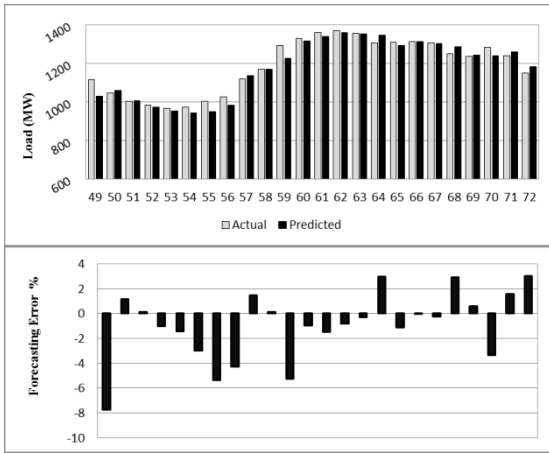
The forecasting results of 3-layered FFNN are presented above. The error curve looks to be near 0 most of the time but the error peaks are very high. The positive and negative maximum errors are 8.94% and -9.66%. The minimum error has gone as low as 0.07%. The error curve shows a stable performance but due to high peaks occurring at the start, end and once during second day forecast, MAPE has gone up to 2.17%.



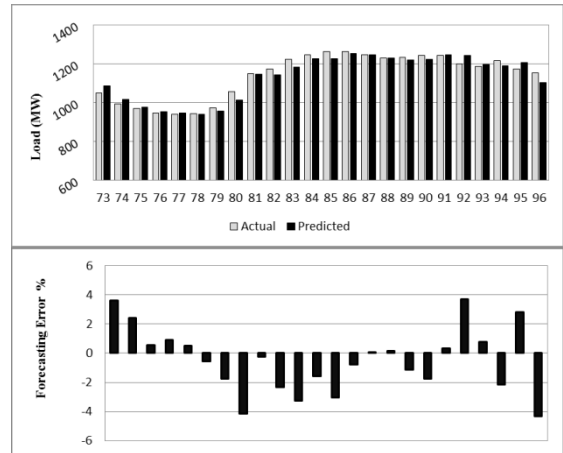
(a) Monday



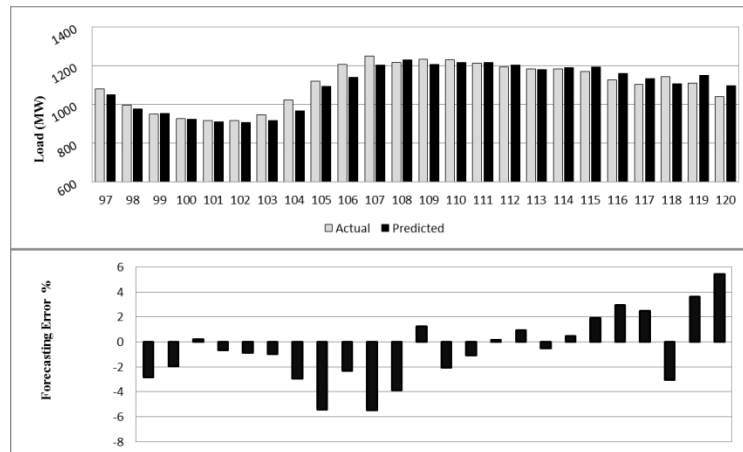
(b) Tuesday



(c) Wednesday



(d) Thursday



(e) Friday

Figure 50: Week ahead forecast results for 3-layered CNN

The 3-layered CNN produces best overall performance with MAPE of 1.99.

Though error rises to above 5% both above and below the axis for 12 times error distribution throughout is uniformly oscillating around -3% to 2.5% and many forecasted points near to 0 have improved MAPE. The error performance during the peak hours is not as good as compared to networks containing recurrent paths but with small variations in error is within acceptable limits.

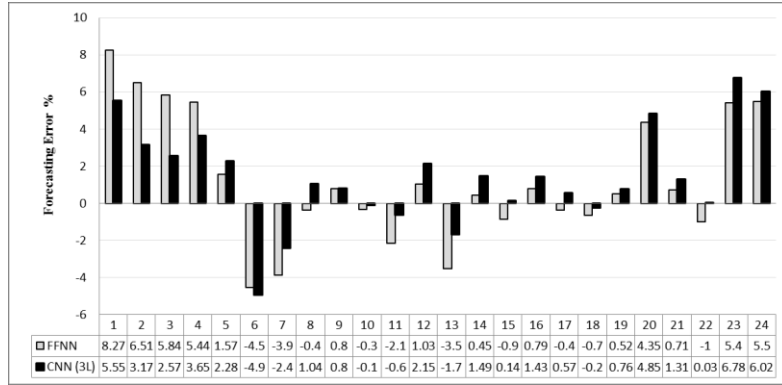
5.1.4 Comparative Analysis of Week ahead forecast for Weekdays

Following the discussion of each individual network for week ahead forecasting, Table 4 presents the MAPE and RMSE of all the networks. Since the FFNN have generated the highest forecasting error so the improvement of other networks are compared with it.

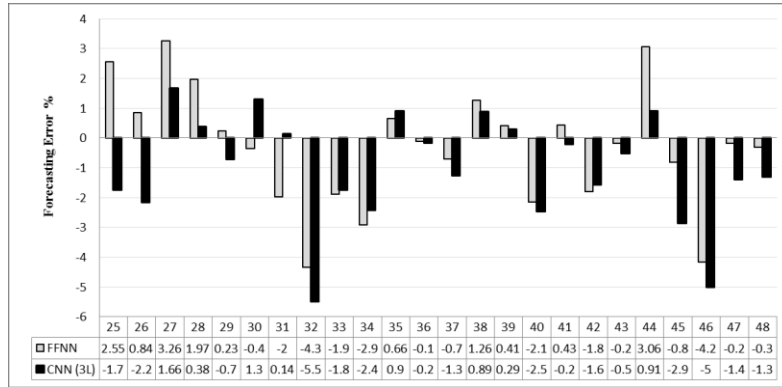
Table 4: Week ahead MAPE and RMSE for Nova Scotia

Architecture	MAPE	RMSE	Improvement
FFNN	2.21	31.45	0.00%
CNN	2.03	28.69	8.26%
RNNL	2.16	30.80	2.32%
RNNG	2.08	29.32	5.83%
CRNNL	2.10	30.24	4.81%
CRNNG	2.12	29.93	4.15%
FFNN (3L)	2.17	31.72	1.76%
CNN (3L)	1.99	29.08	10.04%

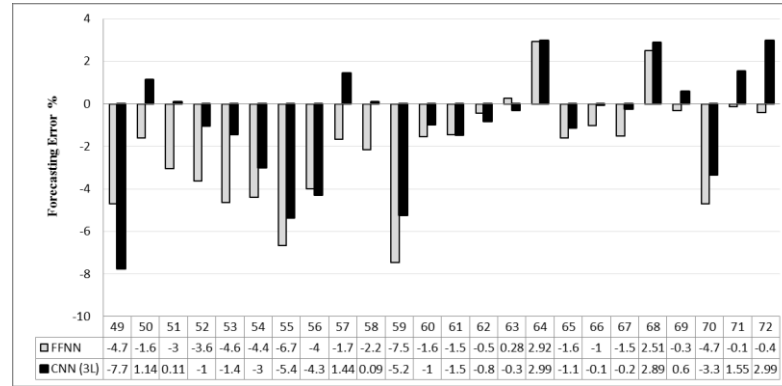
The cascaded networks CNN and 3-layered CNN have proved to be better than other networks with improvements of 8.26% and 10.04% respectively. The other networks have also shown improvement specially the RNNG network (5.83%). The prediction of 3-layered FFNN (1.76% improvement) is much like FFNN but the overall error curve is found to be consistent. Also all recurrent networks perform better during peak hours. The hybrid networks CRNNL (4.81%) and CRNNG (4.15%) also proved to be better than the FFNN and produced lower error during the starting and ending hours during the days of selected week. A comparison of forecasting error performance between FFNN and 3-layered CNN is given in Figure 51:



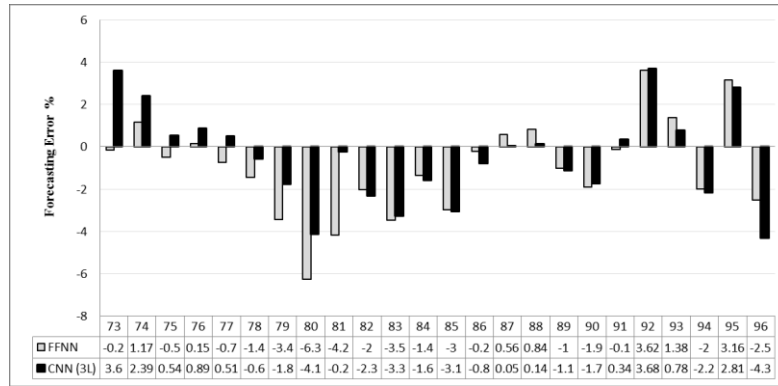
(a) Monday



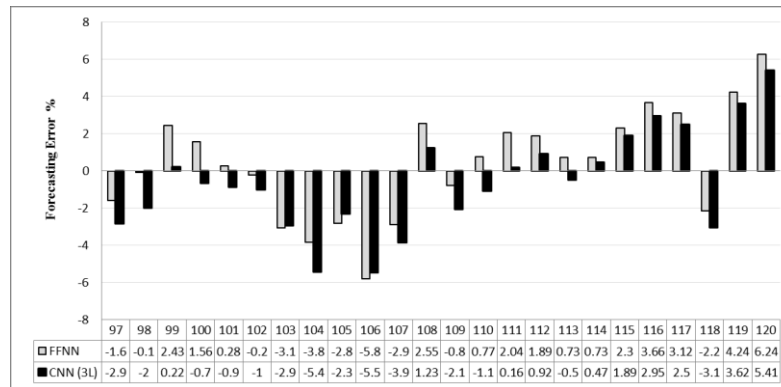
(b) Tuesday



(c) Wednesday



(d) Thursday

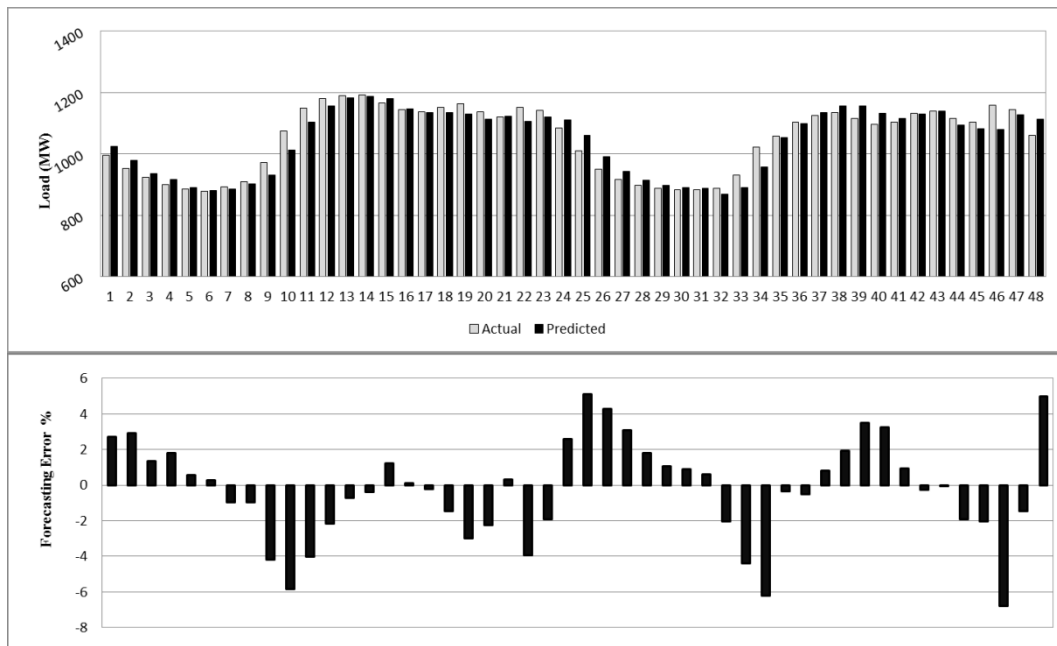


(e) Friday

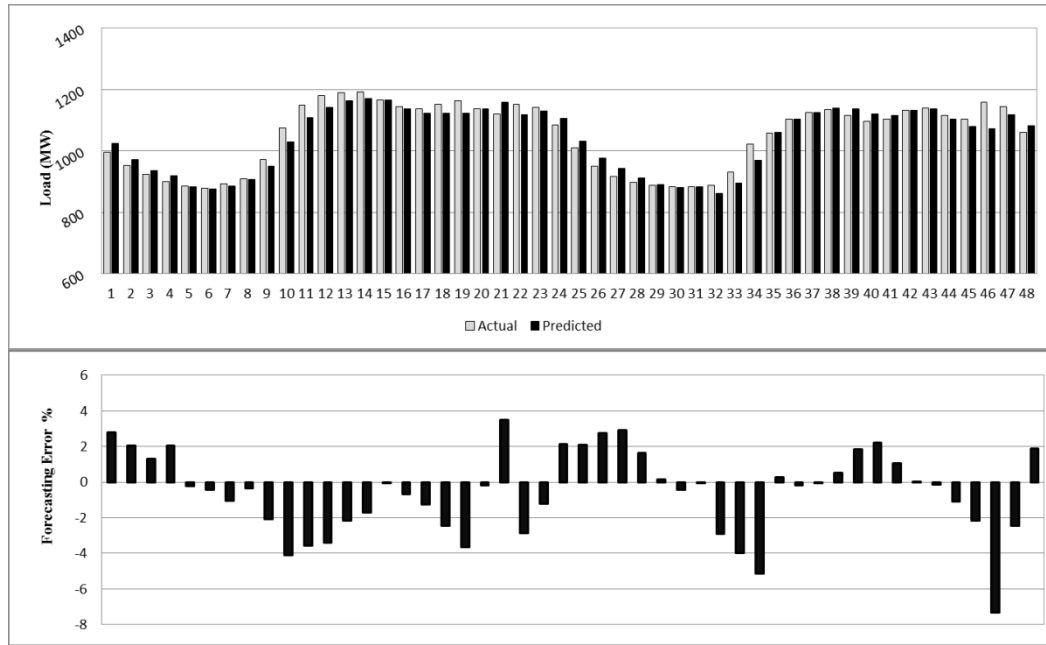
Figure 51: Week ahead forecasting error performance of comparison between FFNN and CNN (3L) for Nova Scotia

5.1.5 Load forecast for Weekend

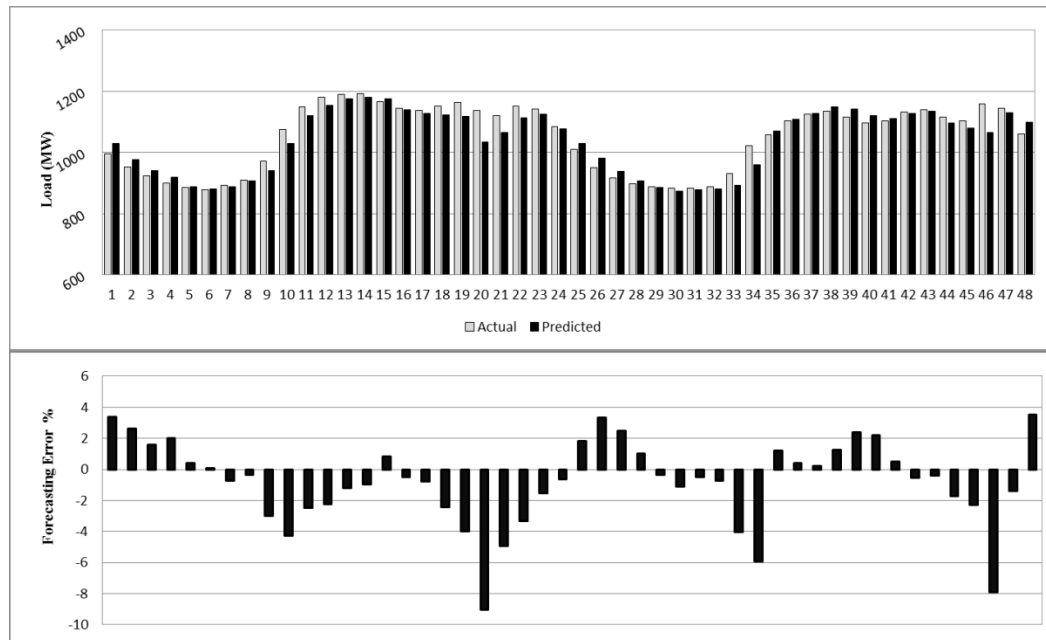
After forecasting the load for day ahead and week ahead, the eight network architectures are used to forecast weekend load. This section elaborates the performance of each network. The weekend days selected to forecast are August 24-25, 2013. It is the last weekend of the month. A comparison between the actual and forecasted load along with the variations in error that occurred during forecast is presented for each network. The results of first four architectures; FFNN, CNN, RNNL and RNNG are demonstrated in Figure 52:



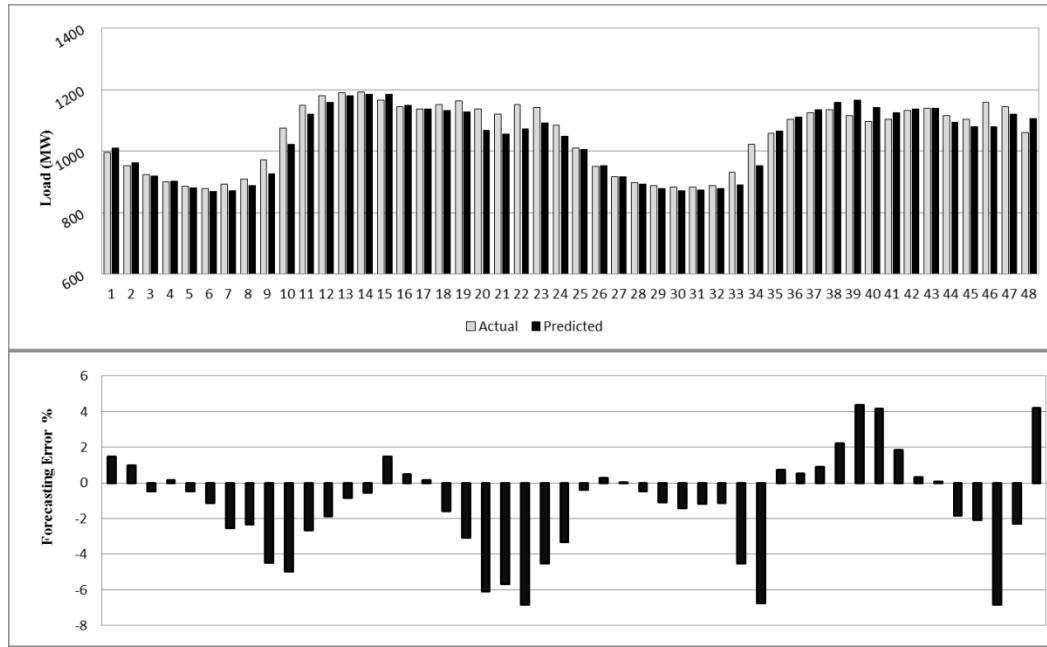
(a) FFNN



(b) CNN



(c) RNNL

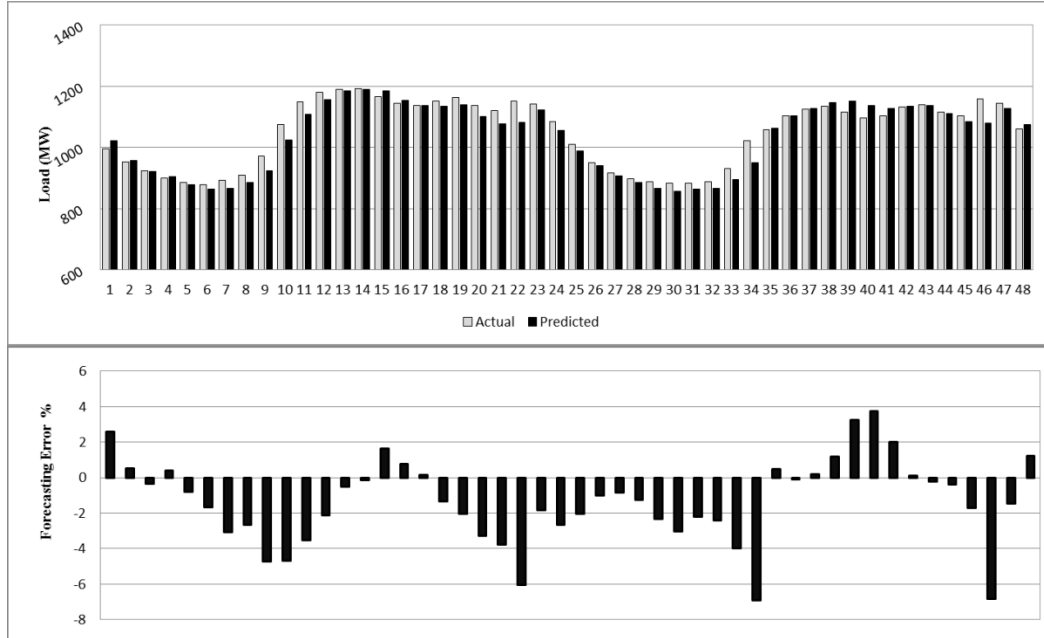


(d) RNNG

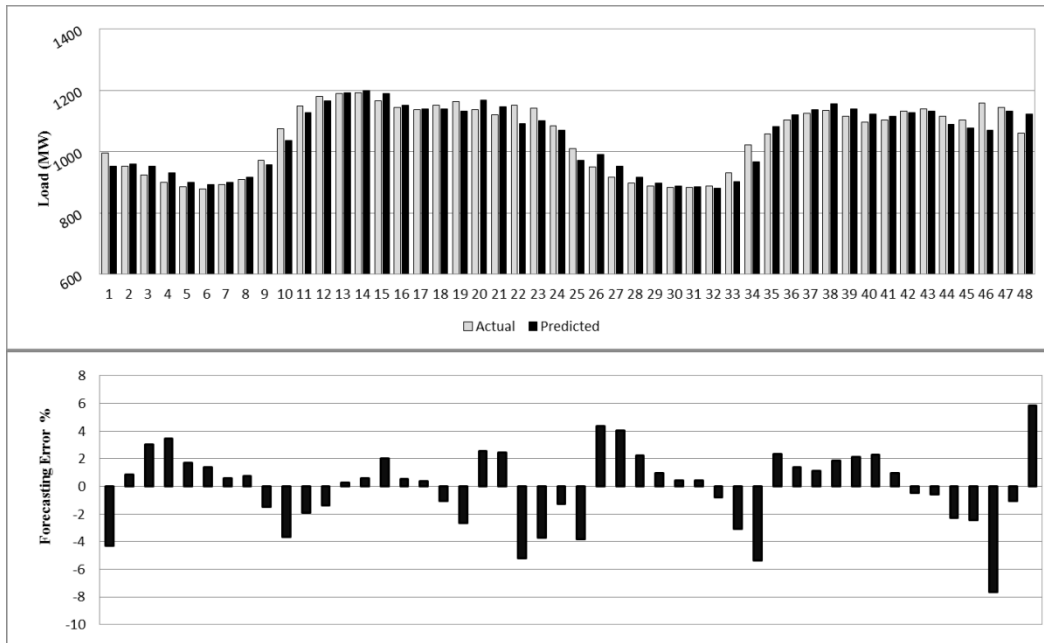
Figure 52: Weekend forecast results (FFNN, CN, RNNL, RNNG)

Above Figure 52 presents the Weekend load forecast for Nova Scotia produced by FFNN, CNN, RNNL and RNNG. It is observed that the performance of FFNN and CNN during peak hours is better than recurrent networks. The forecasting error for FFNN ranges from -6.8% to 5.09% and a minimum error of 0.03%. Its MAPE is 2.16%. For CNN the error varies from -7.32% to 3.48% with the same minimum error as FFNN. It suppresses forecasting errors during the starting and ending hours of both Saturday and Sunday as a result produce lower MAPE of 1.84%. If we compare it with RNNL whose MAPE is 2.09% and error ranges from -9.04% to 3.51%, CNN with added advantage of faster convergence seems a better option. Though the negative error spike of -9.04% occurs during the peak hour on Saturday and a similar error spike results on Sunday at almost the same time due to which MAPE has suffered. The most erroneous performance in all is that of RNNG. Its MAPE is 2.42% and possesses an error range of -6.84% to 4.34%. Throughout the forecast predicted curve keep producing error spikes mostly

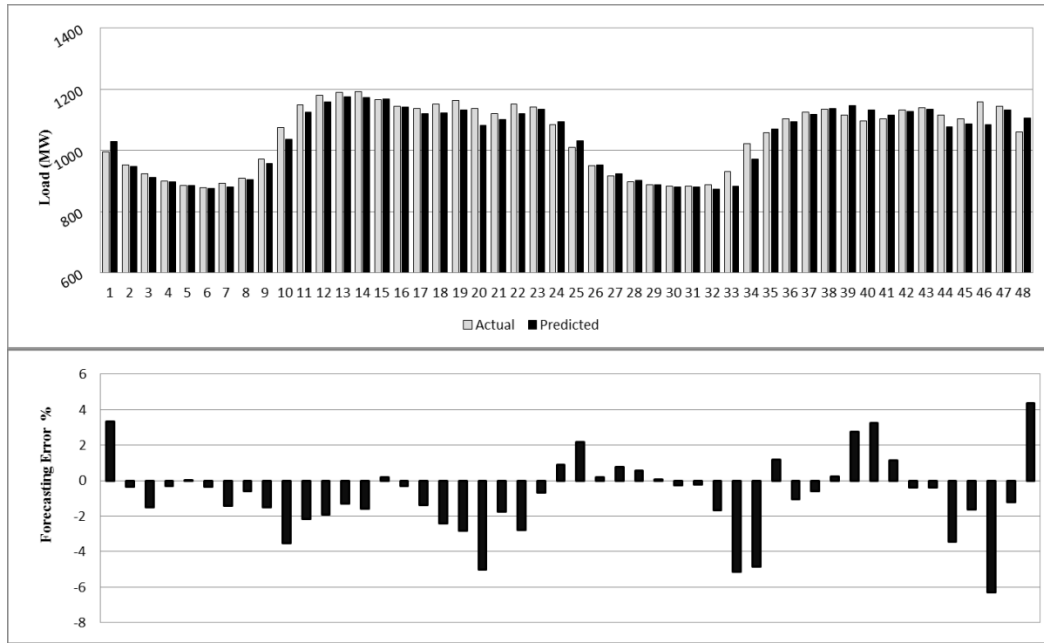
negative and ended up with worst MAPE of all. Furthermore, load forecasts of remaining four networks; CRNNL, CRNNG, 3-layered FFNN and 3-layered CNN are presented in Figure 53:



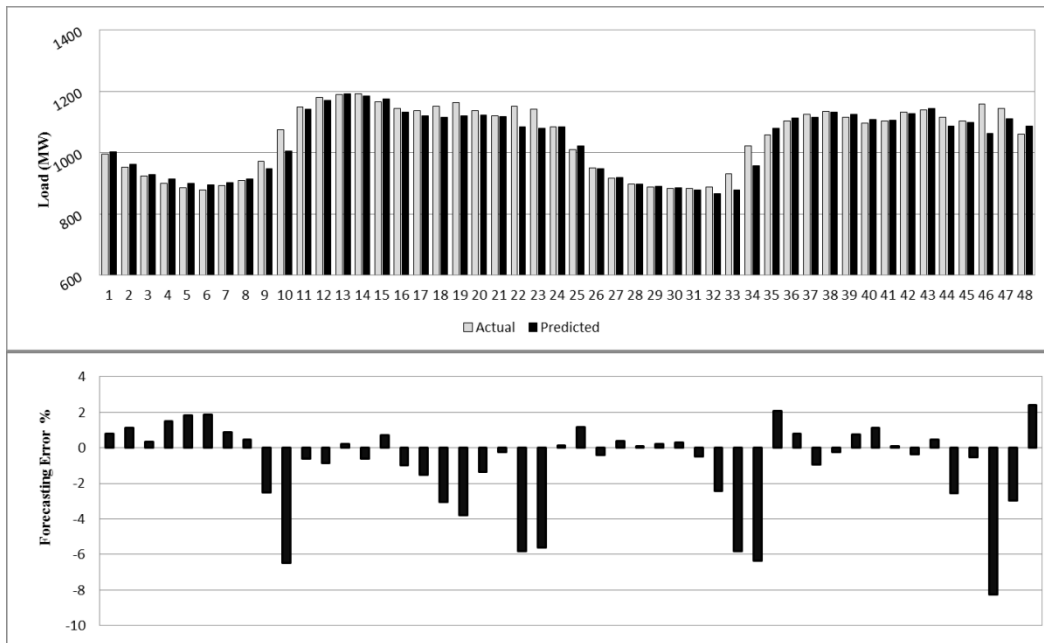
(a) CRNNL



(b) CRNNG



(c) FFNN (3L)



(d) CNN (3L)

Figure 53: Weekend forecast results (CRNNL, CRNG, FFNN (3L), CNN (3L))

CRNNL gives a slightly better performance than RNNL. The error is reduced for the peak hours but increased for starting hours of both days. A similar MAPE of 2.08% is produces. Forecasting error ranges from -6.89% to 3.73% which is a narrow range. The

CRNNG on the other hand offers a big improvement with MAPE 1.76%. It suppresses forecasting errors throughout both days. Its error ranges from -7.62% to 5.85%. The 3-layered FFNN improves the overall performance to a greater extent. Its MAPE is 1.7% and error range is from -6.28 to 4.35% with a minimum error of 0.01%. Finally 3-layered CNN produces MAPE of 1.93% and error range of -8.25% to 2.4% respectively.

5.1.6 Comparative Analysis of Load forecast for Weekend

For weekend forecast the RNNG has given the worst performance among all other architectures. Its performance is taken as a base and an improvement of each network over RNNG is given along with the summary of error performances.

Table 5: Weekend forecast MAPE and RMSE for Nova Scotia

Architecture	MAPE	RMSE	Improvement
FFNN	2.16	29.44	3.46%
CNN	1.84	25.80	17.65%
RNNL	2.09	30.92	6.54%
RNNG	2.24	32.69	0.00%
CRNNL	2.08	28.74	7.07%
CRNNG	1.76	28.77	21.53%
FFNN (3L)	1.70	24.84	23.93%
CNN (3L)	1.93	28.68	13.68%

3-layered FFNN has produced best results with 23.93% improvement over RNNG. CRNNG and CNN with performance enhancements of 21.53% and 17.65% are also worth mentioning. Specifically the CRNNG gives the best performance during peak hours. The remaining networks; 3-layered CNN (13.68%), CRNNL (7.07%), RNNL (6.54) and FFNN (3.46%) have also shown better performance for weekends as they did for weekdays. Overall the performances of all networks for weekends have forecasted better results than weekdays. Figure 54 gives a comparison between the forecasting error performance of RNNG and 3-layered FFNN.

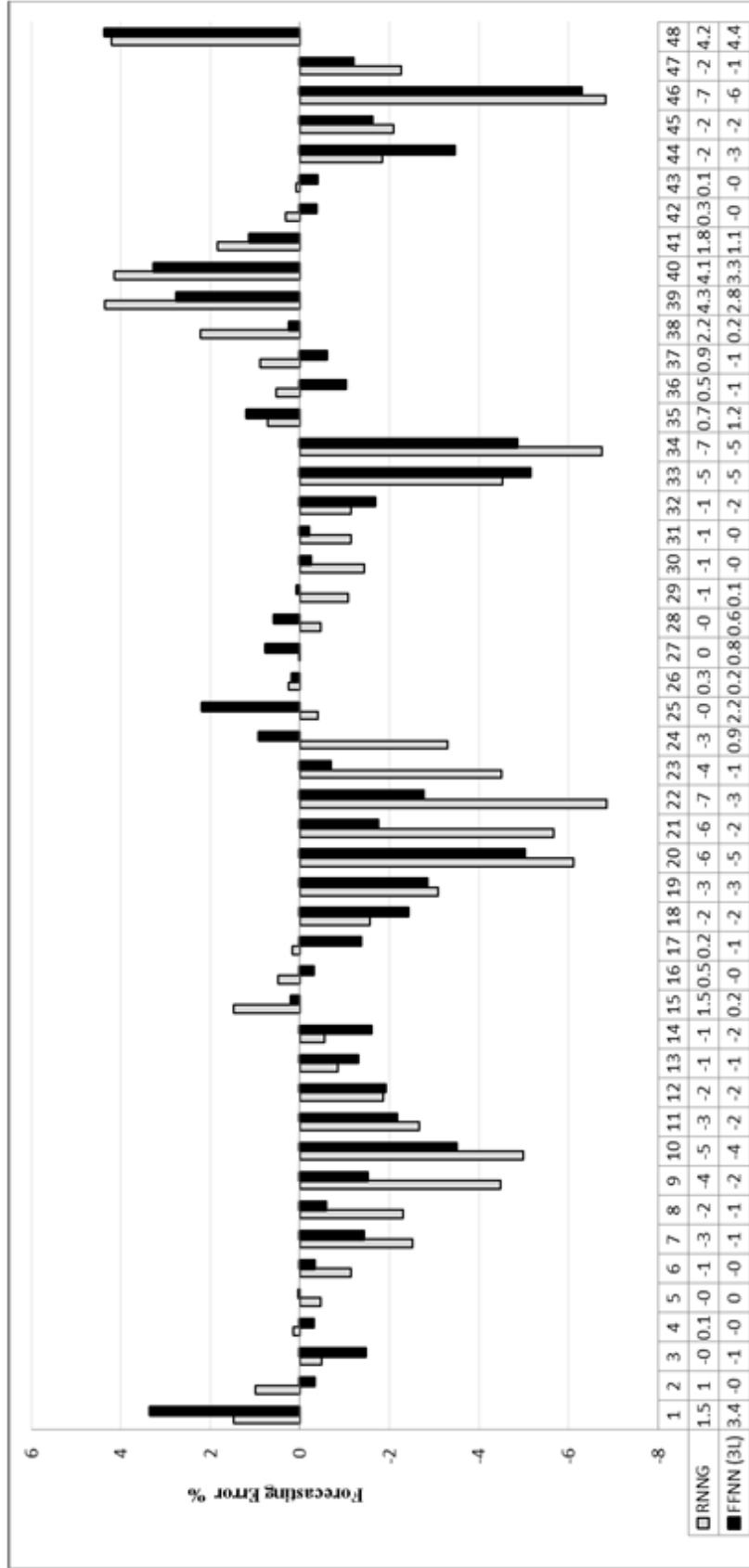


Figure 54: Weekend forecasting error performance comparison between RRNG and FFNN (3L) for Nova Scotia

5.1.7 Summary of Load forecast results for Nova Scotia

Table 6 summarizes all the forecasting results achieved for Nova Scotia followed by a graphical presentation in Figure 55.

Table 6: Summary of Load forecast results for Nova Scotia

Architecture	Day ahead	Week ahead	Weekend
FFNN	2.32	2.21	2.17
CNN	1.93	2.03	1.85
RNNL	2.26	2.16	2.10
RNNG	2.18	2.08	2.24
CRNNL	2.21	2.10	2.09
CRNNG	2.18	2.12	1.76
FFNN (3L)	1.99	2.17	1.71
CNN (3L)	1.81	1.99	1.94

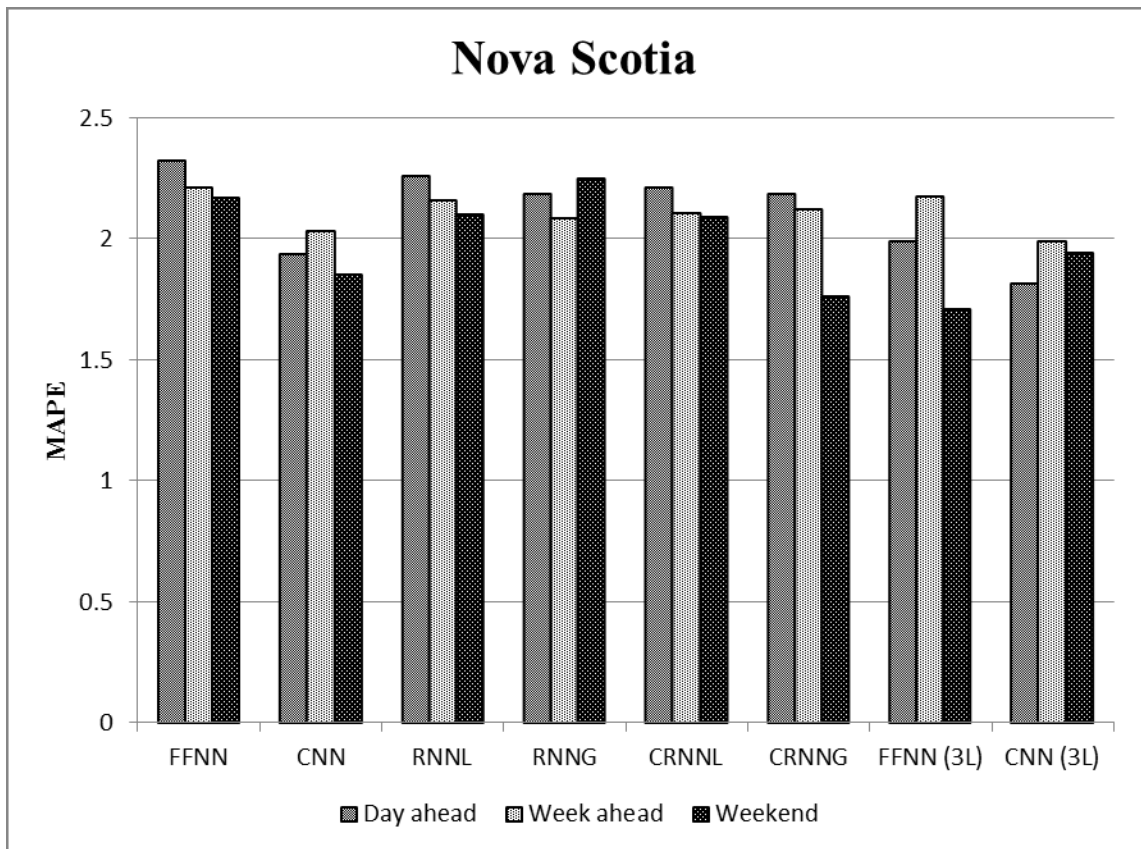


Figure 54: Graphical presentation of load forecast results for Nova Scotia

5.2 New Brunswick

New Brunswick electrical load demand varies from 771 MW to 1450 MW during June to August 2013. The load demand drops down a small amount towards the end of the week. Just like Nova Scotia load and temperature data from June 1 to August 25, 2013 are used for training all the networks. This data was gathered from online resources broadcast by Government of Canada, climate database [64] and New Brunswick power, Operations Data [66]. The 24 hour ahead and week ahead forecasts are presented next for weekdays and weekends.

5.2.1 24 hours ahead forecast for Weekdays

The 24 hours ahead NB load forecast using the eight network architectures discussed in section 4.4 are given for a weekday. The same day is selected for forecast which was selected for testing Nova Scotia load. Figures 56 to 63 present the forecasts produced by each network tested. Each plot gives a comparison between actual and forecasted load as well as forecasting error at each step. In addition, the training and testing data sets for all three jurisdictions is same, hence the network convergence times are also same for New Brunswick and Ontario forecasts.

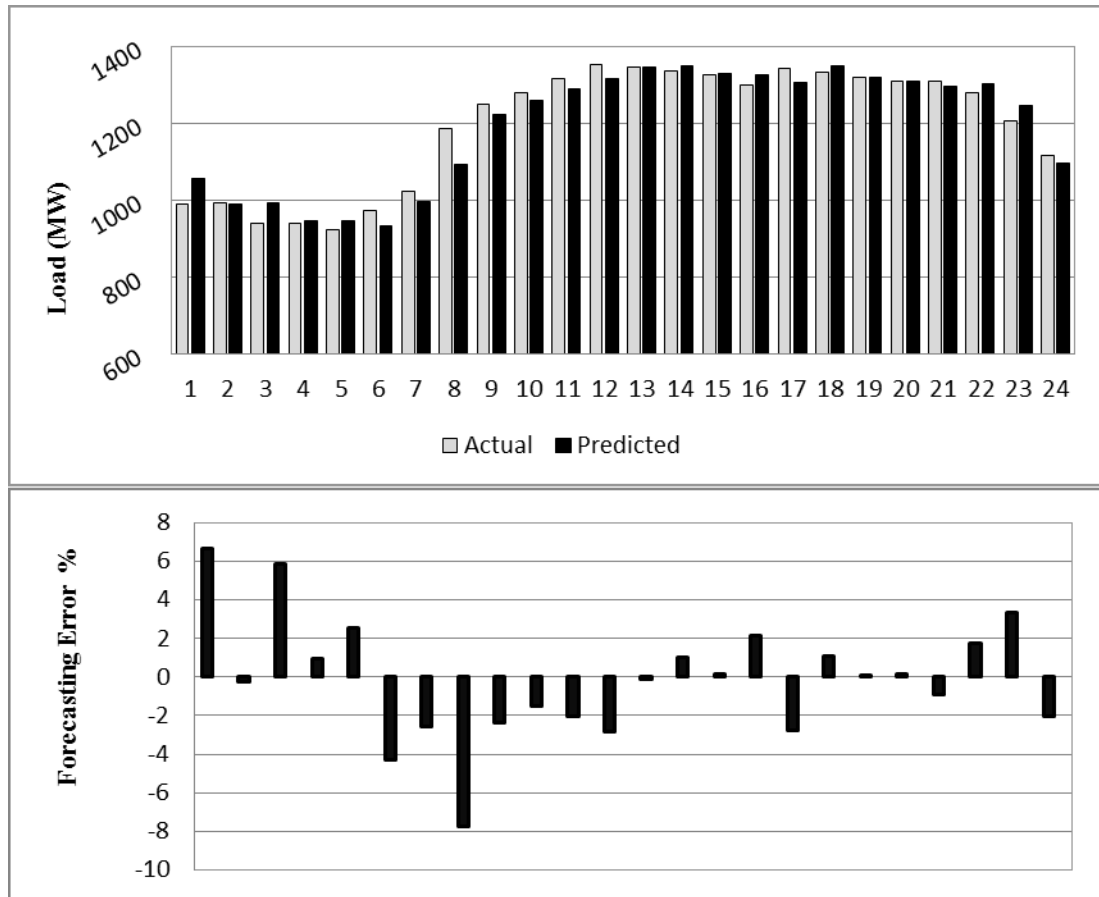


Figure 55: Day ahead forecast results for FFNN

The forecasting results utilizing FFNN for NB load is presented in Figure 56. It can be seen that the predicted curve generates negative error for most part of the forecast but the error seems to be lower than acceptable limits. Overall, 5 times the error falls above or below 3%. The MAPE is 2.3% with maximum positive and negative errors of 6.65% and -7.72%. The forecast from noon to midnight produces accurate results. Hence it can be pointed out that during first half of the day where load demand increases exponentially towards the peak hours, the forecast curve falls short of load demand but once during peak hours the demand becomes stable, network generates accurate forecasts.

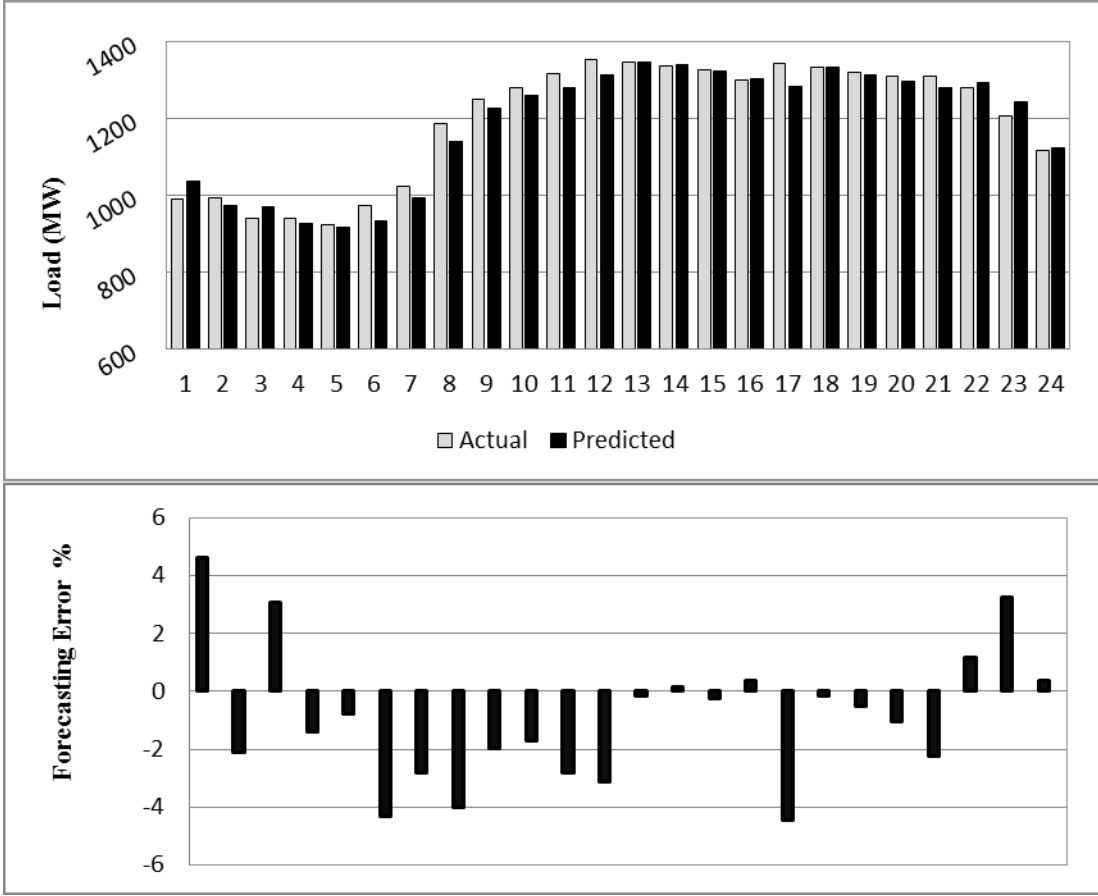


Figure 56: Day ahead forecast results for CNN

The CNN network is implemented and the forecasting results are shown in Figure 57. The accuracy is increased as compared to FFNN. The error ranges from -4.44% to 4.63%. The predicted curve during first half appears similar to FFNN with decreased error. The MAPE for this network is 1.96% which is mainly due to lower peak errors and throughout the day the predicted load closely follows the load demand though for the most part falls below it.

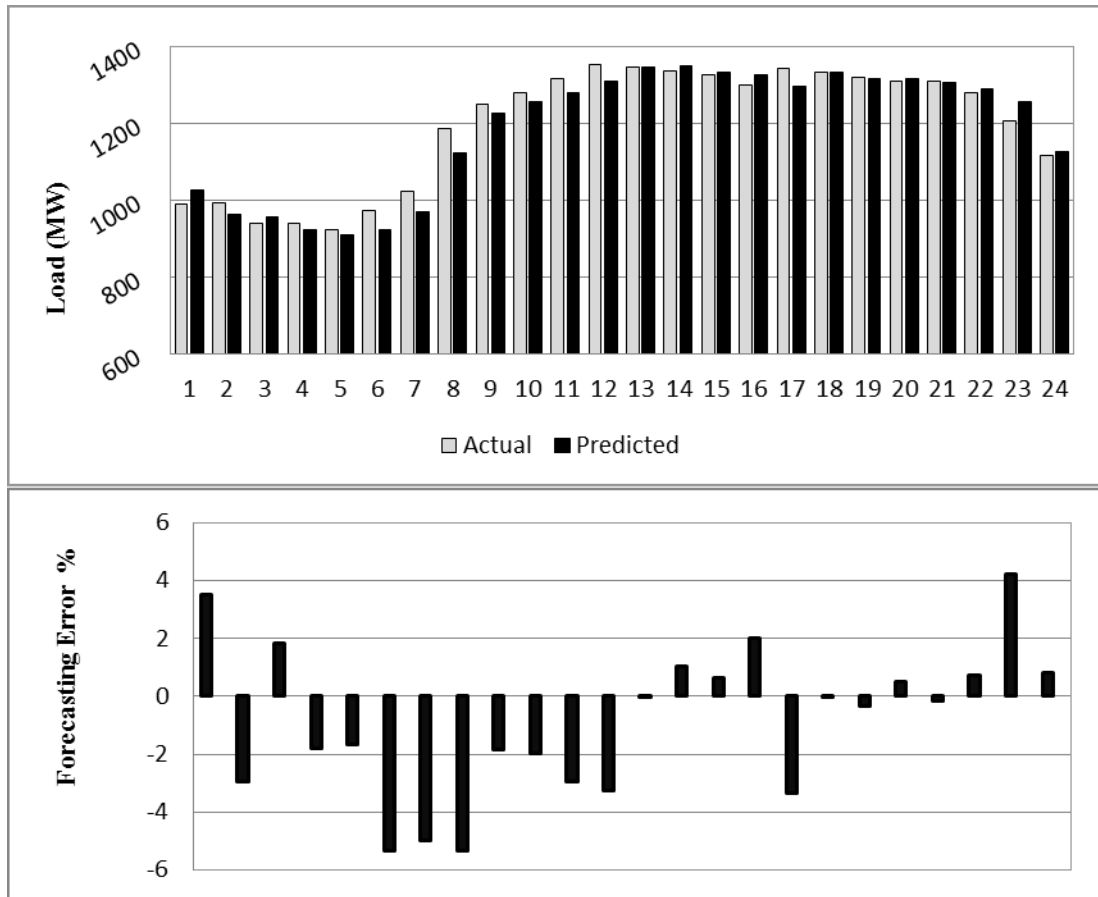


Figure 57: Day ahead forecast results for RNNL

The RNNL also shows improvement over FFNN with a MAPE of 2.14% as compared to 2.3%. The overall response follows the same theme as FFNN with erroneous forecast during first half of the day. Upon close observation it can be noticed that big negative errors occur for most part of the first half. With an error ranging from -5.23% to 4.21% most of errors -2% occurs during first 12 hours with two positive and negative error spikes towards the end.

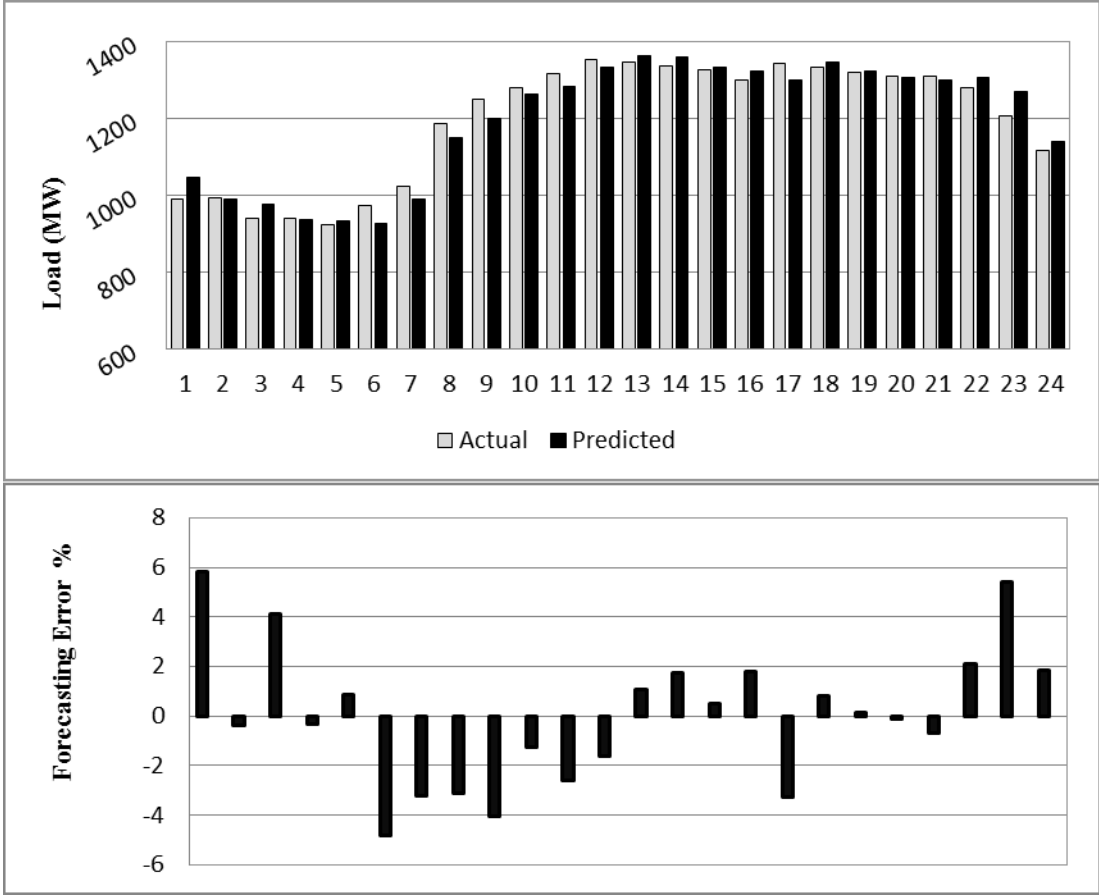


Figure 58: Day ahead forecast results for RNNG

With almost similar overall performance and a MAPE of 2.16%, load forecast made by RNNG is shown in Figure 59. The error distribution is much better than RNNL especially during first 12 hours. Due to high error peaks the overall error performance has suffered. Except first and last couple of hours error curve is quite stable. The maximum errors range from -4.84% to 5.81%.

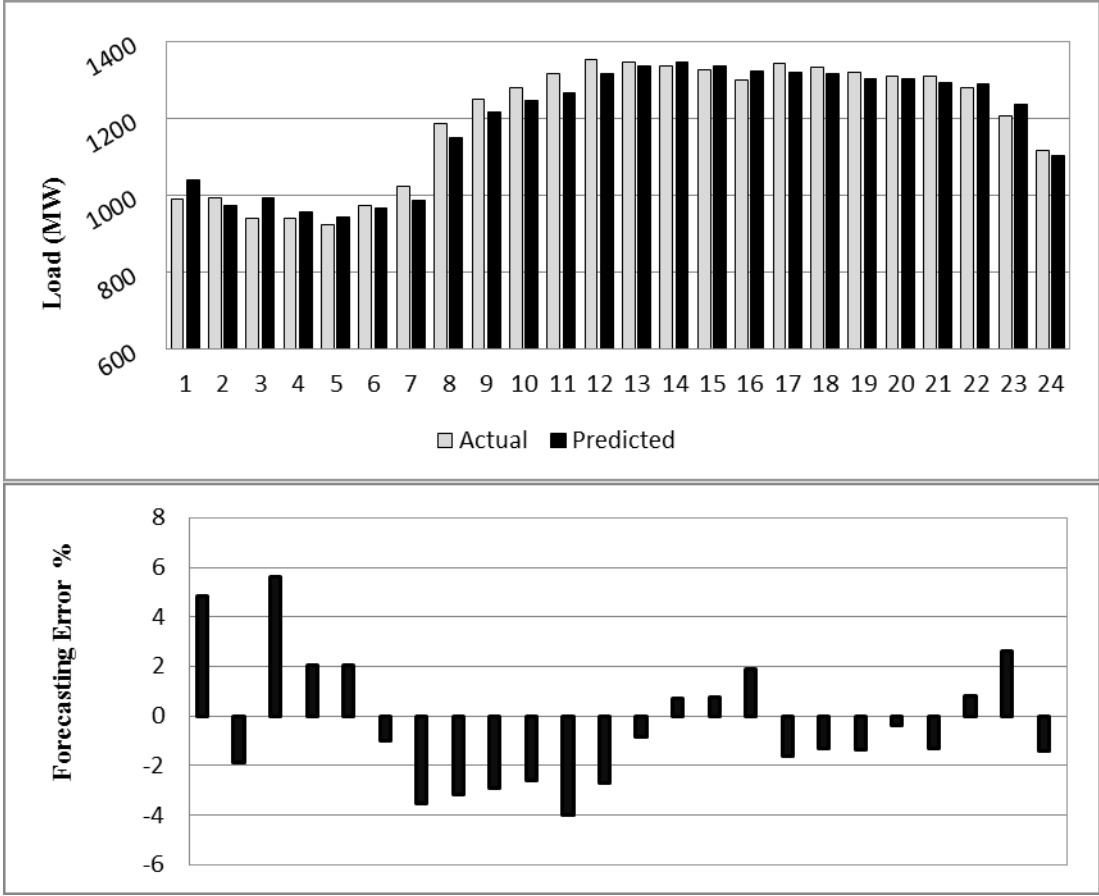


Figure 59: Day ahead forecast results for CRNNL

The hybrid network CRNNL has produced a MAPE of 2.14% which is same as RNNL but negative error has greatly reduced. Another factor worth highlighting is stable curve throughout. Though due to occurrence of positive peaks twice have increase the MAPE. The negative error performance has shown big improvement. This network has produced the least negative error among all the other networks tested yet. The error ranges from -3.97% to 5.6%.

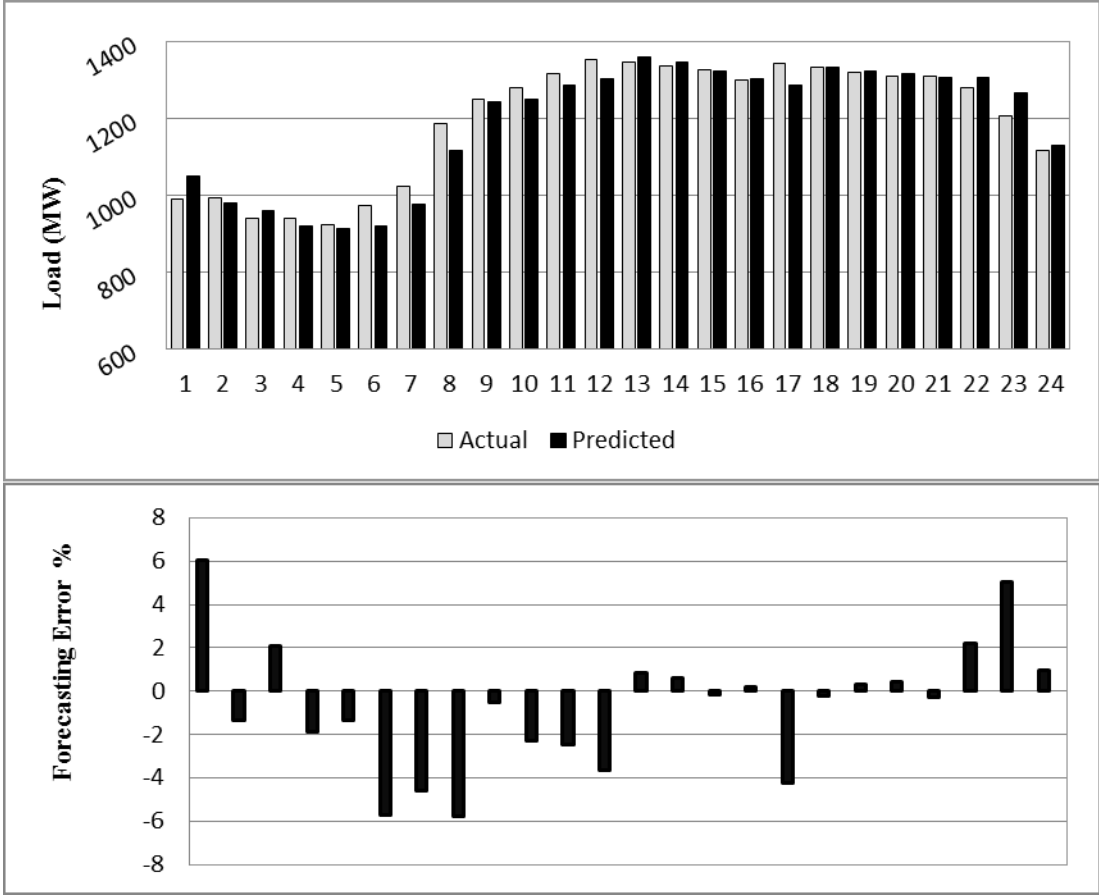


Figure 60: Day ahead forecast results for CRNNG

With MAPE equal to 2.21%, forecasting results for CRNNG are presented above. Though overall error performance of this network is inferior from most of the networks but error distribution is much more stable. Except forecasts for 1st and 4th to 8th hours, results produced for rest of the day are less erroneous than FFNN, RNNL, RNNG or CRNNL. Due to the consistent occurrence of error spikes during the hours mentioned the overall performance has suffered. CRNNG possesses an error range of -5.79% to 6.04% which is the worse range yet.

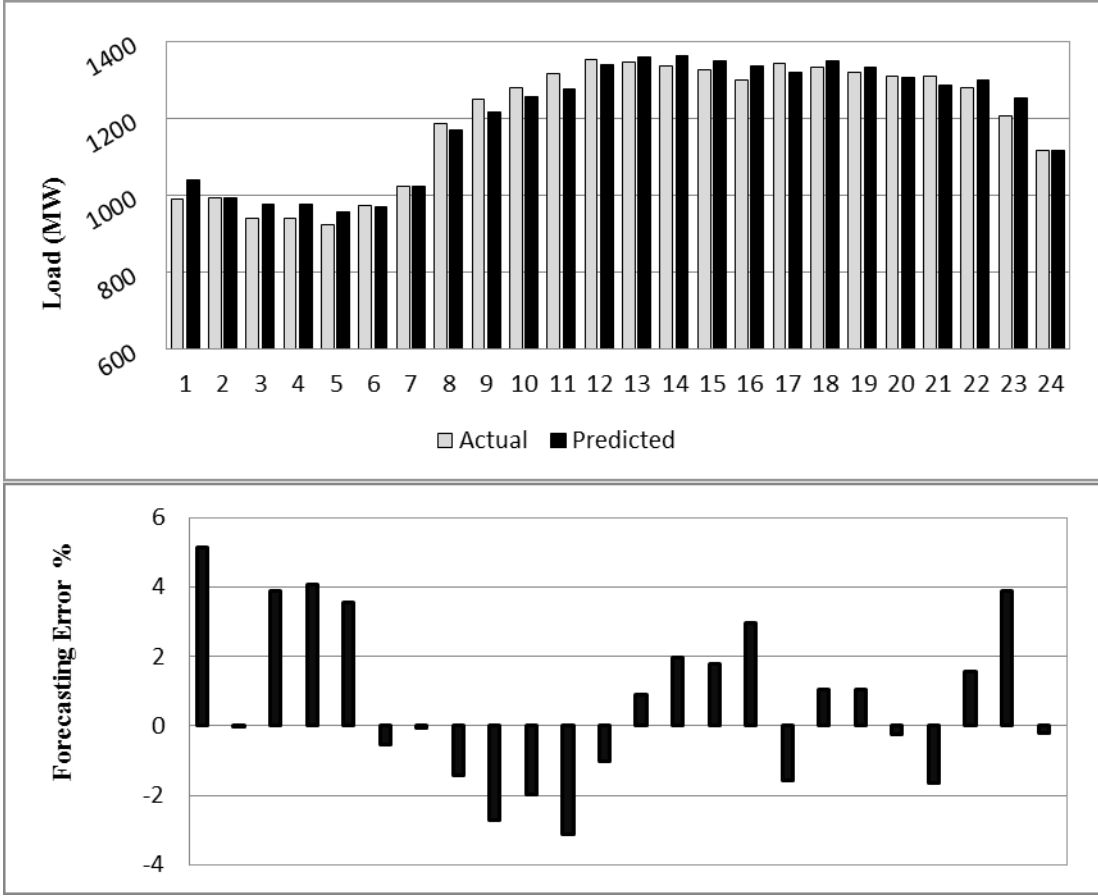


Figure 61: Day ahead forecast results for 3-layered FFNN

With an increment of a layer in the architecture performance of FFNN has improved a great deal. The negative error during first half of the day has greatly reduced. With error peaks of -3.11% to 5.13%, 3-layered FFNN has produced accurate results. The MAPE has also decreased to 1.93% as compared to 2.3% of FFNN. The predicted curve has very closely followed the demand curve and has produced a very stable error distribution.

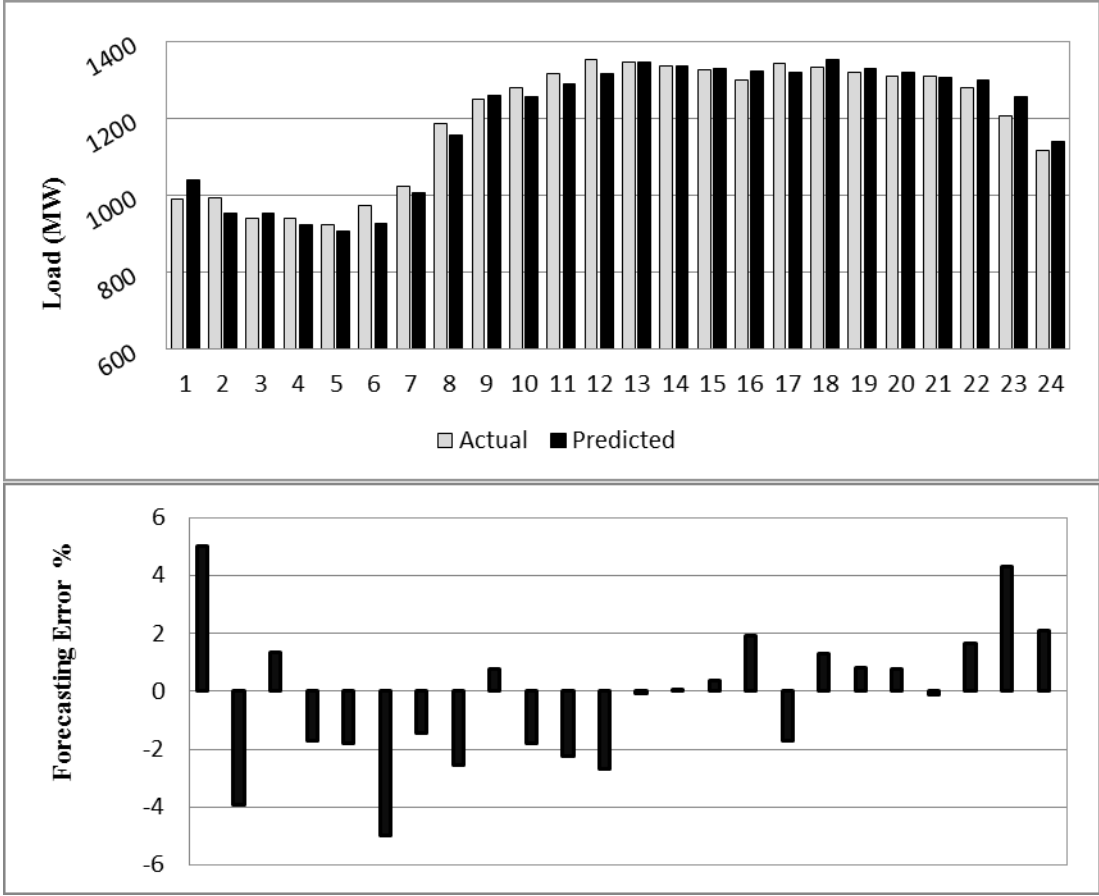


Figure 62: Day ahead forecast results for 3-layered CNN

The forecast generated by the 3-layered CNN is presented in Figure 63. The predicted and demand curve are almost a match with small error throughout. The negative error during first half of the day has greatly reduced. Error ranges from -4.97% to 5.03%. Error curve is also very stable and a MAPE of 1.89% indicates high accuracy maintained throughout the forecast.

5.2.2 Comparative Analysis of 24 hours ahead forecast for Weekdays

For 24 hour NB load forecast for weekdays the FFNN has produced the least accurate results. Table 7 summarizes the error performances of the eight networks and gives the improvement of each individual network over FFNN.

Table 7: 24 hours ahead MAPE and RMSE for New Brunswick

Architecture	MAPE	RMSE	Improvement
FFNN	2.30	34.01	0.00%
CNN	1.96	28.34	14.98%
RNNL	2.14	30.70	7.19%
RNNG	2.16	31.00	6.15%
CRNNL	2.14	28.32	6.89%
CRNNG	2.21	33.63	3.77%
FFNN (3L)	1.93	26.87	15.99%
CNN (3L)	1.89	26.21	17.71%

The 3 best performing networks are CNN, 3-layered FFNN and 3-layered CNN. The 3-layered CNN outperforms all other networks with improvement of 17.71% over FFNN. CNN (14.98%) and 3-layered (15.99%) have also produced accurate results with fast convergence but the main advantage of 3-layered CNN over all the networks is its uniform error distribution throughout the day. Most of the networks have failed to suppress error during the first half of the day except 3-layered CNN. The performances of RNNL (7.19%), RNNG (6.15%), CRNNL (6.89%) and CRNNG (3.77%) look almost alike but the hybrid networks have reduced error during first half of the day to a larger extent than other two. Thus keeping all the aspects under consideration the best choice among all the networks is 3-layered CNN. Furthermore a comparison of forecasting error performance between 3-layered CNN and FFNN is given in Figure 64.

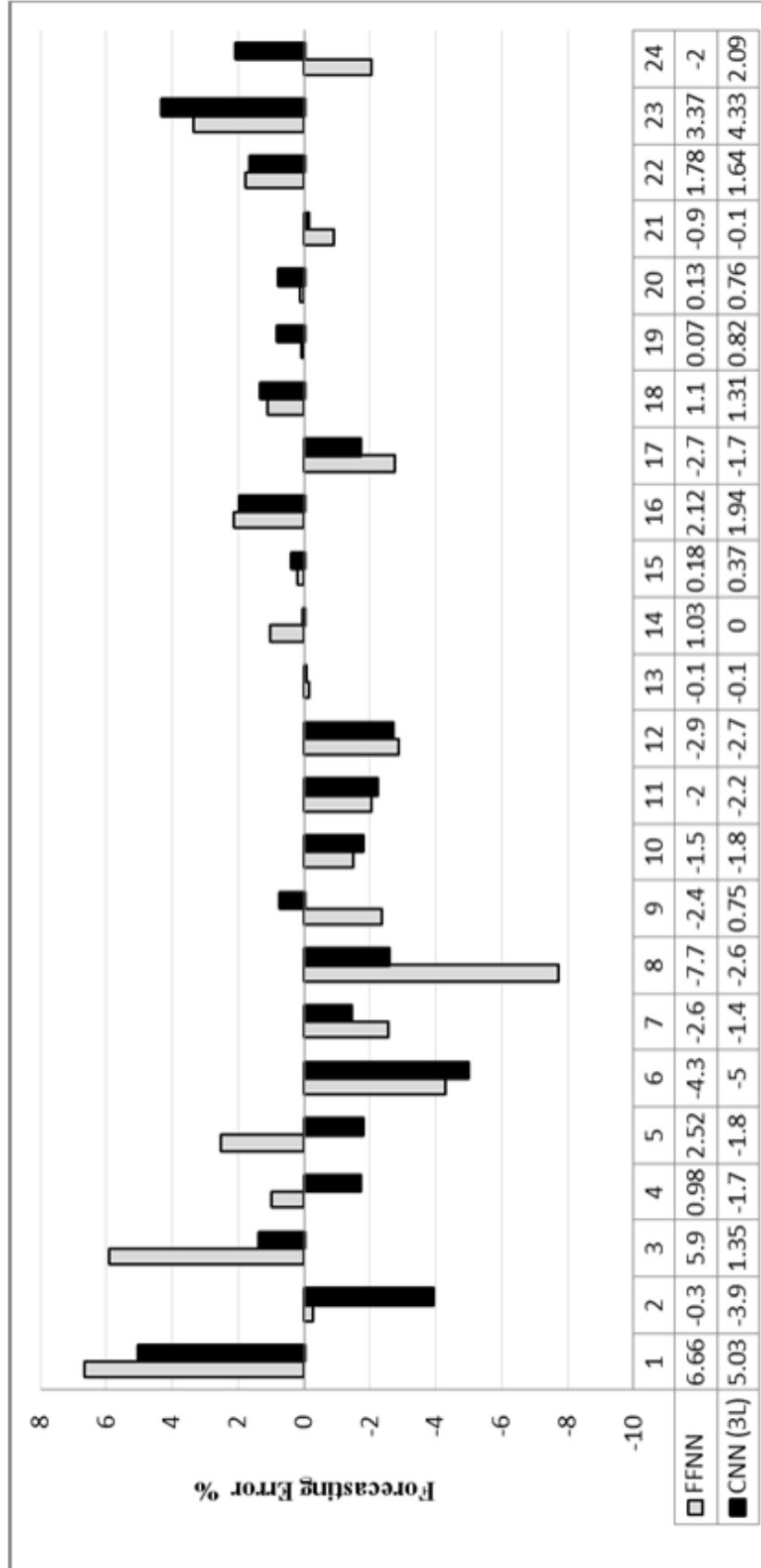
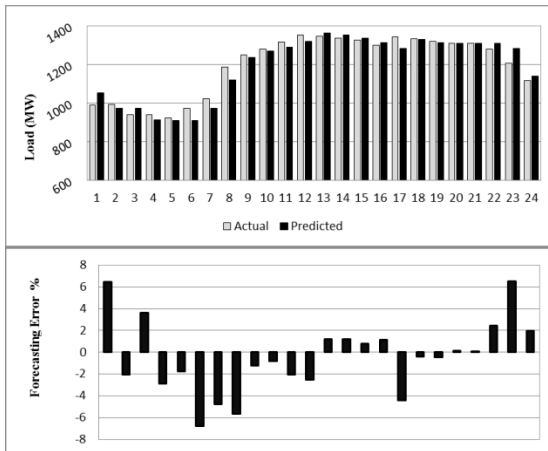


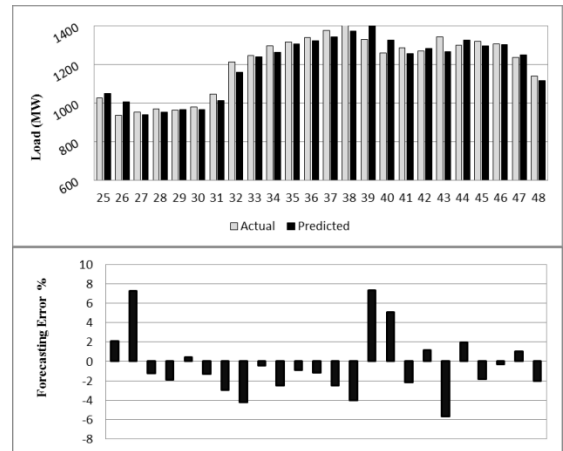
Figure 64: 24 hour ahead forecasting error performance comparison between FFNN and CNN (3L) for New Brunswick

5.2.3 Week ahead forecast for Weekdays

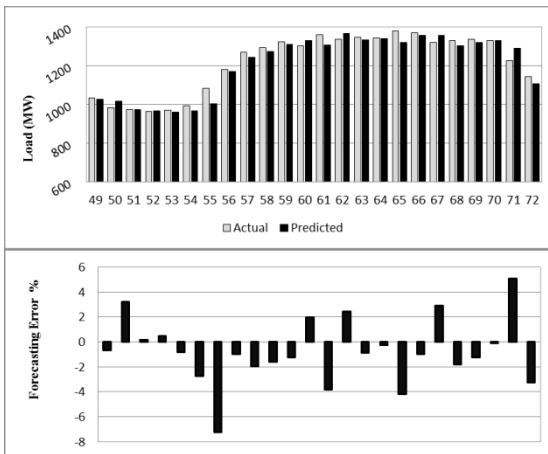
The week ahead NB load forecast for weekdays is discussed in this section. The same week is selected to forecast the load as was done for Nova Scotia. The eight networks treated in section 4.4 are tested and their results are elaborated as follows:



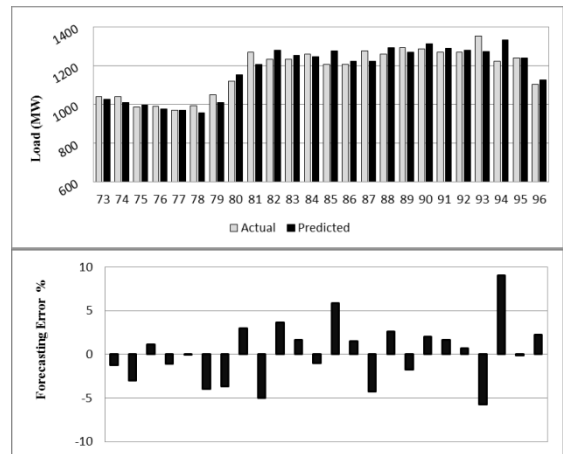
(a) Monday



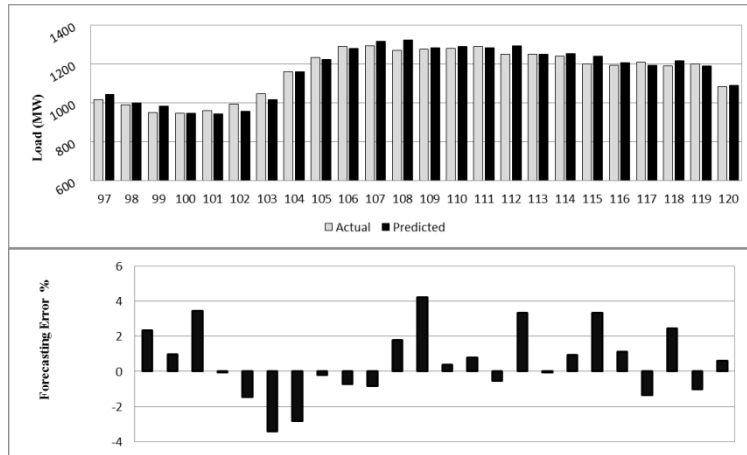
(b) Tuesday



(c) Wednesday



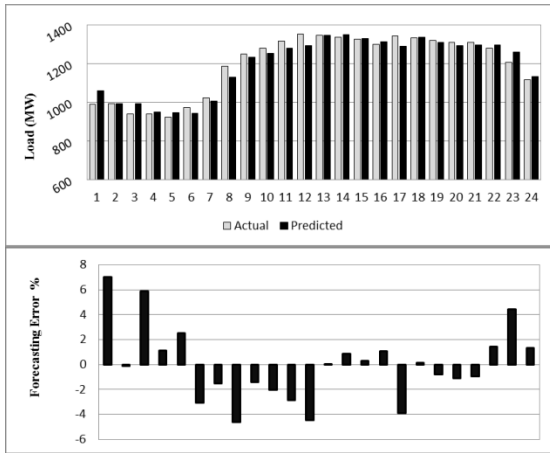
(d) Thursday



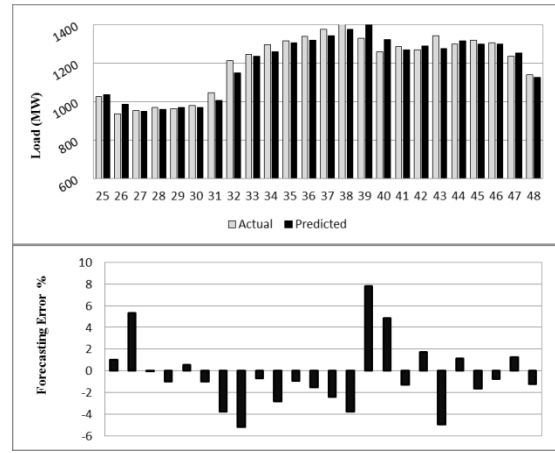
(e) Friday

Figure 63: Week ahead forecast results for FFNN

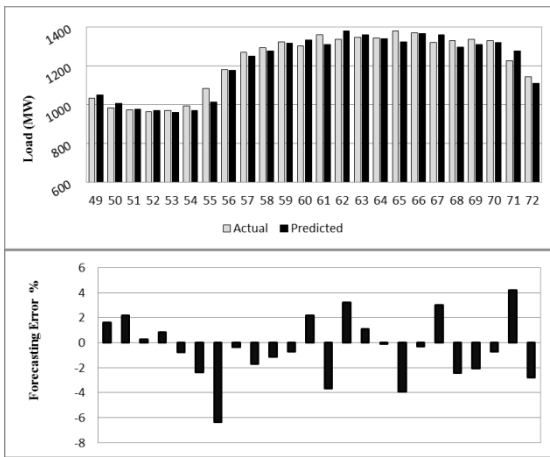
The forecasting results for FFNN are shown in Figure 65. The error ranges from -7.24% to 9%. A negative error is observed during the starting hours of each day. The forecast for the fourth day (Thursday) is the most accurate. It is seen that overall load demand start to drop during fifth day (Friday) but it do not affect the forecast. A noticeable amount of peaks also occurs during the forecast of the first day (Monday). The MAPE is 2.31%.



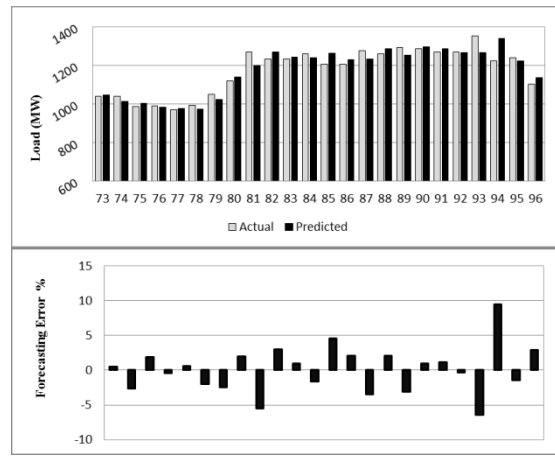
(a) Monday



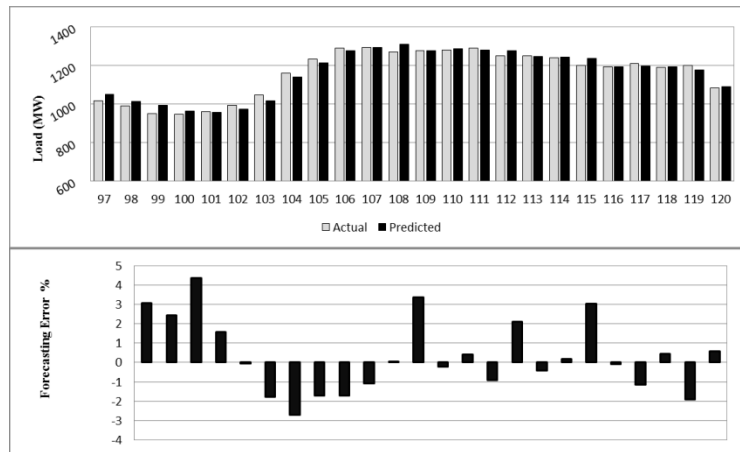
(b) Tuesday



(c) Wednesday



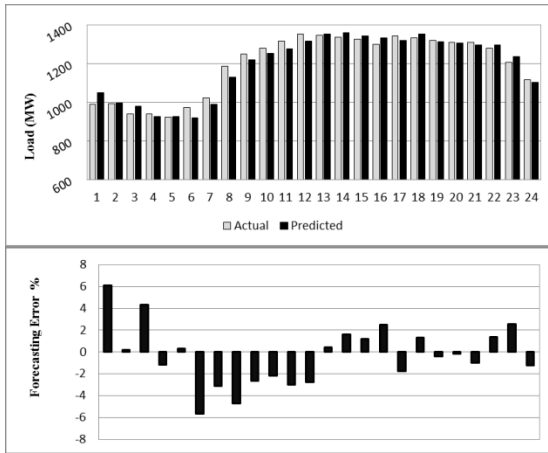
(d) Thursday



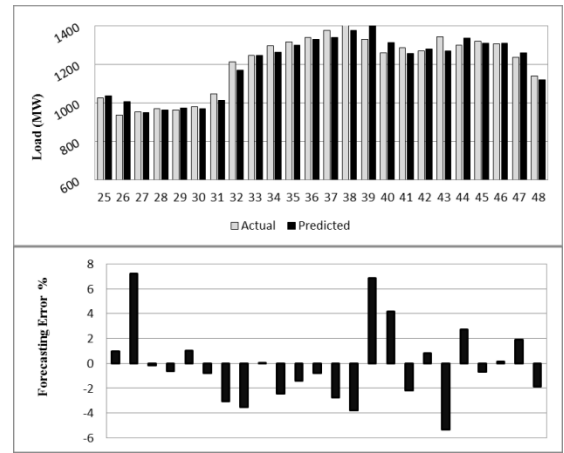
(e) Friday

Figure 64: Week ahead forecast results for CNN

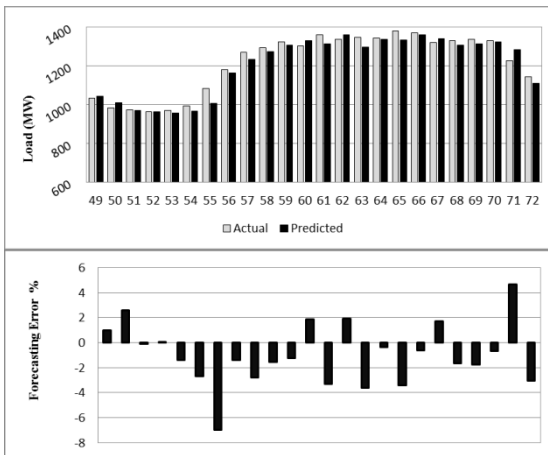
An increased accuracy is observed in the forecast generated by CNN. The error spikes that occurred during forecast of Monday are eliminated. Though for Thursday the forecasting error is increased but the overall performance is steady. The MAPE for this network is 2.12% and the error ranges from -6.4% to 9.47%. The error curve has passed above or below 5% only in 7 occasions out of a total of 120 hours. The negative error that occurred during starting hours of every day for FFNN has also been suppressed. As a result the forecast produced is uniform.



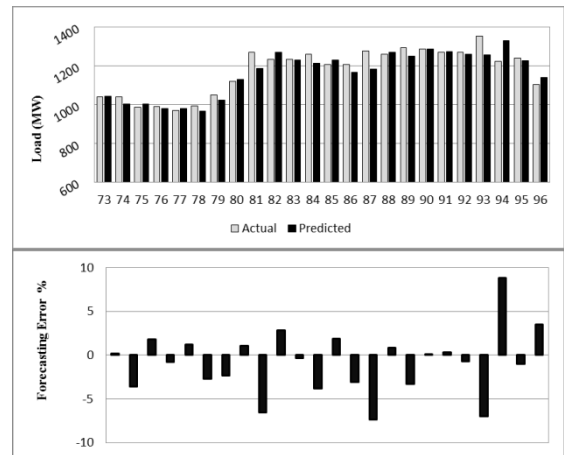
(a) Monday



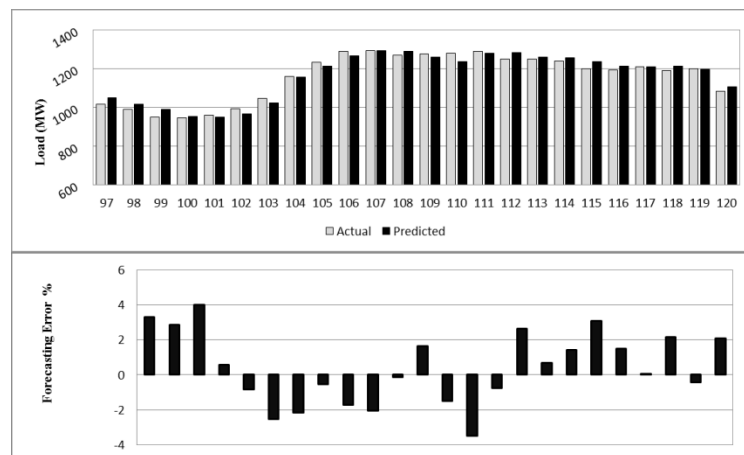
(b) Tuesday



(c) Wednesday



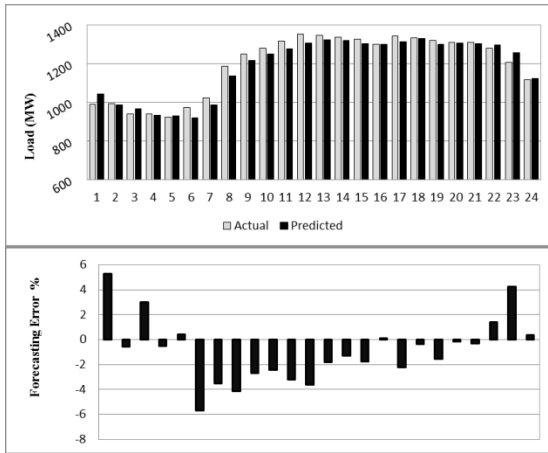
(d) Thursday



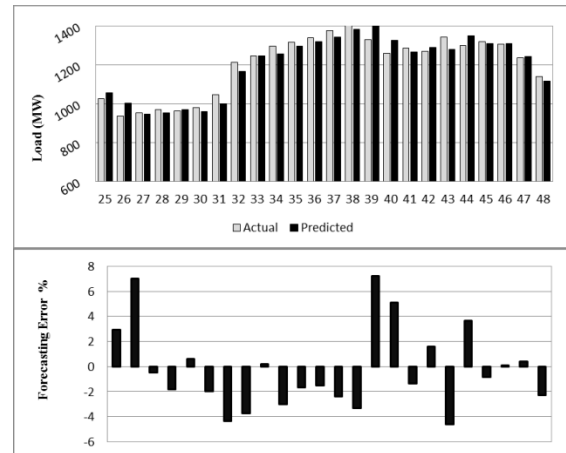
(e) Friday

Figure 65: Week ahead forecast results for RNNL

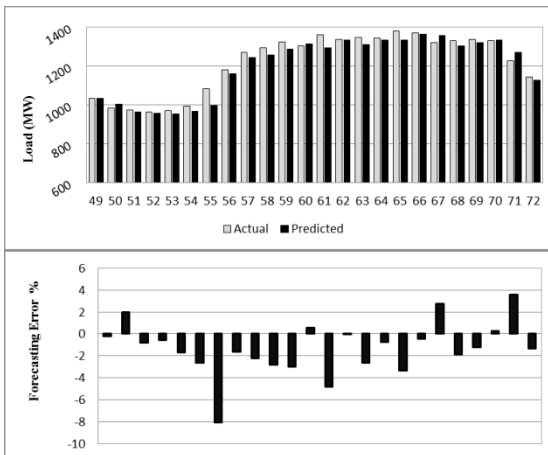
Figure 67 demonstrates the forecasting results for RNNL. The error peaks are - 7.32% and 8.82%. The MAPE is 2.20%. It's also a big improvement over FFNN. Performance during peak hours is not very satisfying. Most of the errors are towards negative side of the axis.



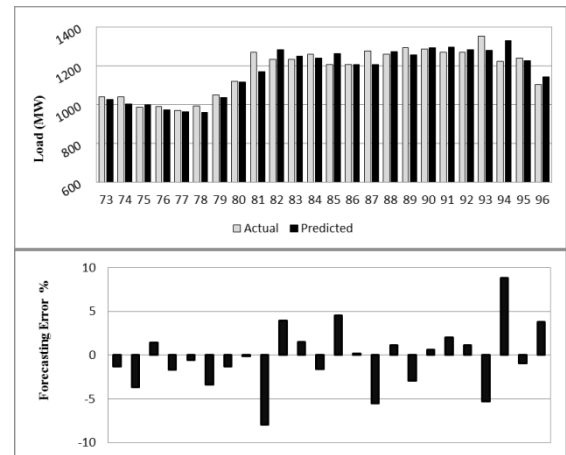
(a) Monday



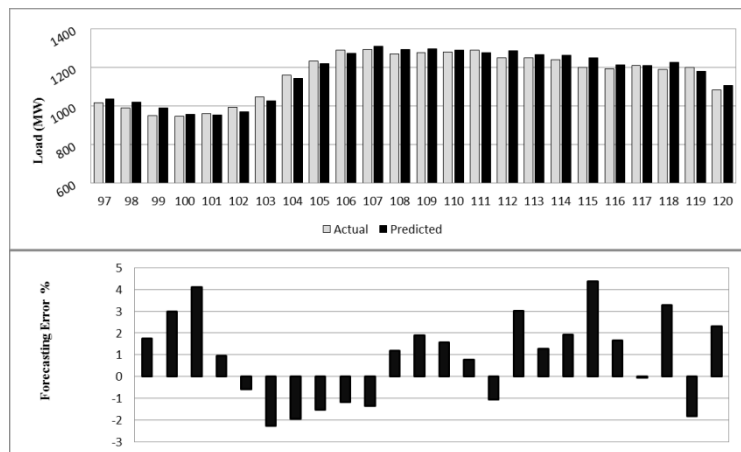
(b) Tuesday



(c) Wednesday



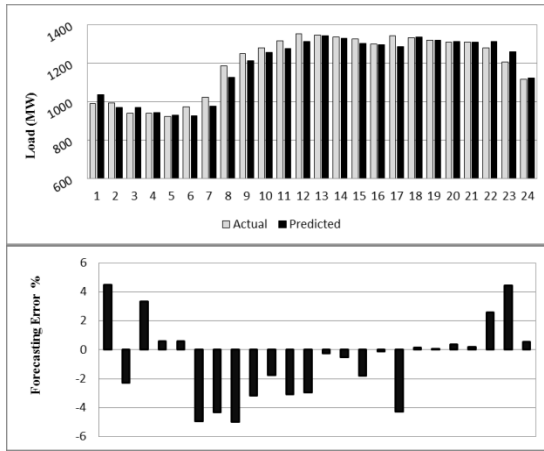
(d) Thursday



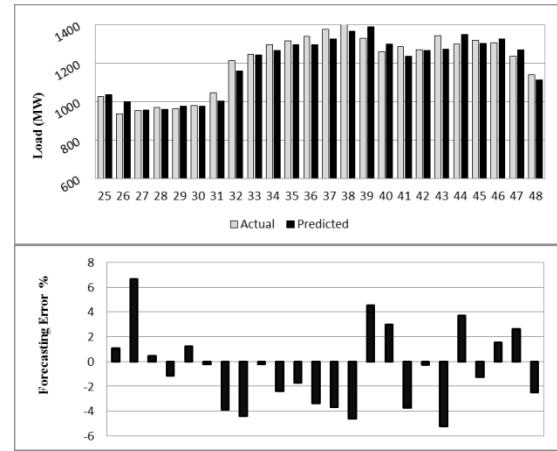
(e) Friday

Figure 66: Week ahead forecast results for RNNG

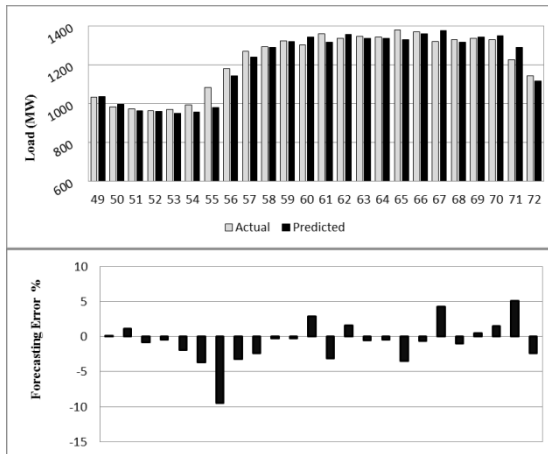
With a MAPE of 2.27% RNNG do not offers much improvement over FFNN. Of course the training time as a recurrent network is also a factor to consider. It has the same problem as was found in FFNN of a rise in error during starting hours of each day. The rest of the performance is much like FFNN. Its error ranges from -8.04% to 8.82%.



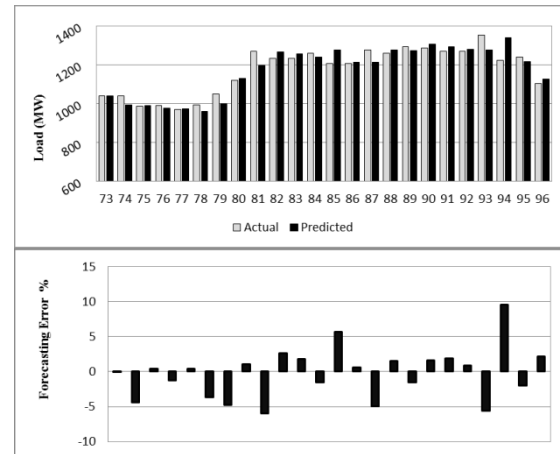
(a) Monday



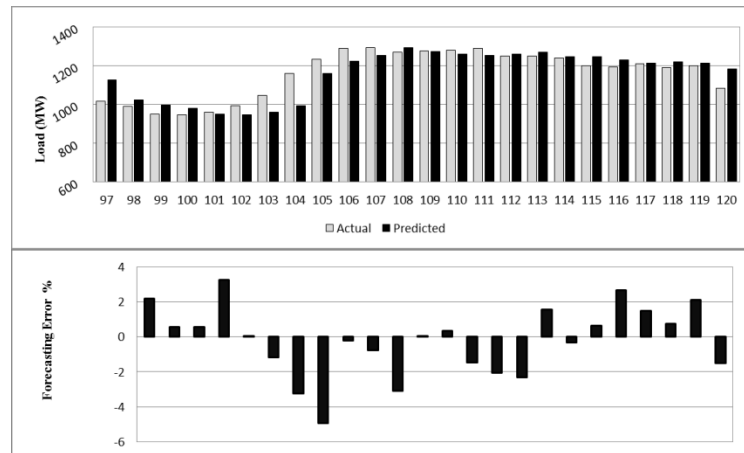
(b) Tuesday



(c) Wednesday



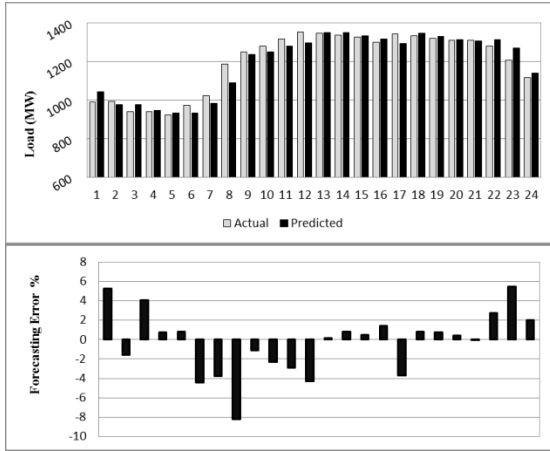
(d) Thursday



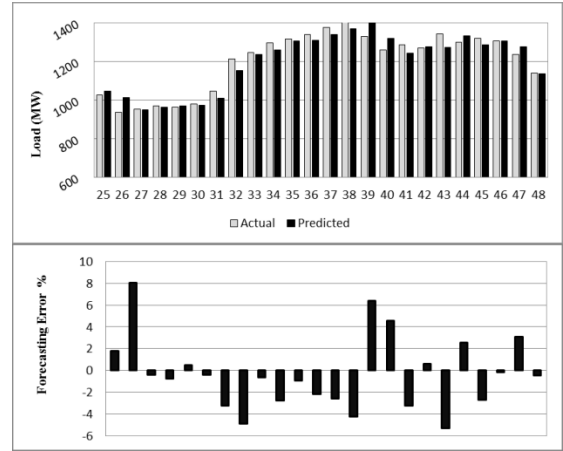
(e) Friday

Figure 67: Week ahead forecast results for CRNNL

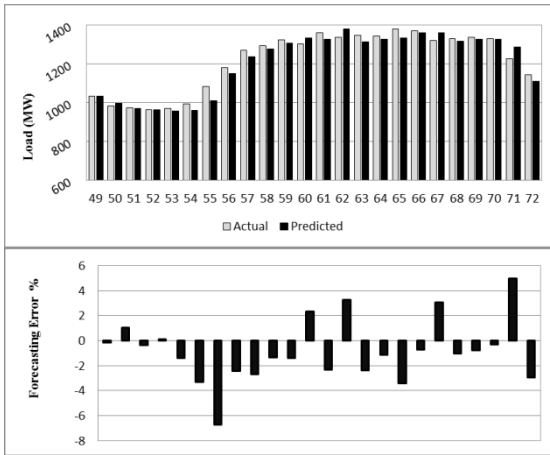
CRNNL produces a very stable forecast. MAPE is 2.24% which is due to contribution of high error peaks. This network offers good accuracy during peak hours and can be considered over FFNN. Except for Thursday the error spikes mostly occur during the non-peak hours. It error ranges from -9.47 to 9.52% which is quite high. The error distribution is uniform and throughout with few points where error rises abruptly.



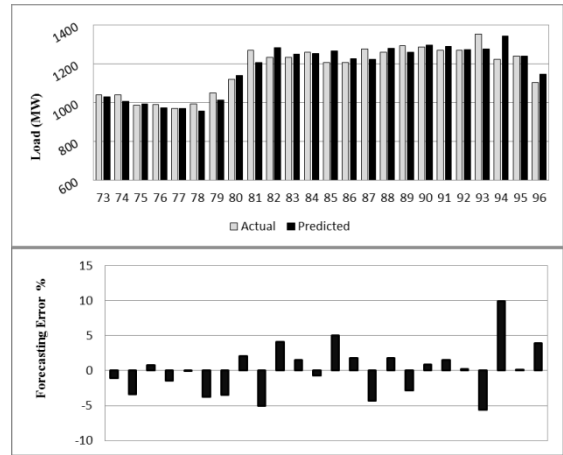
(a) Monday



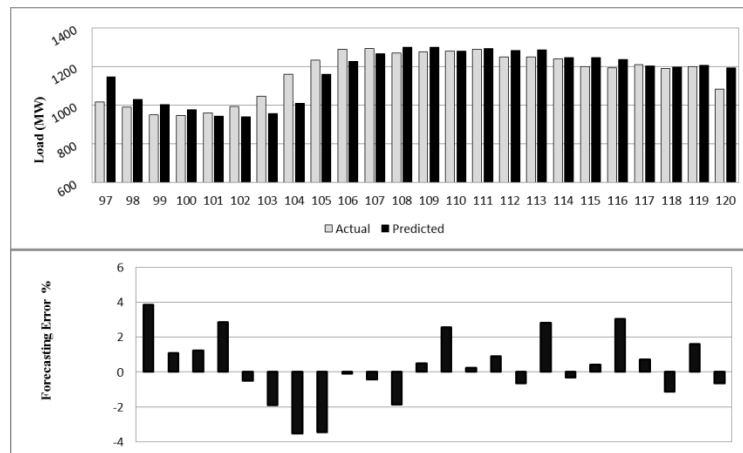
(b) Tuesday



(c) Wednesday



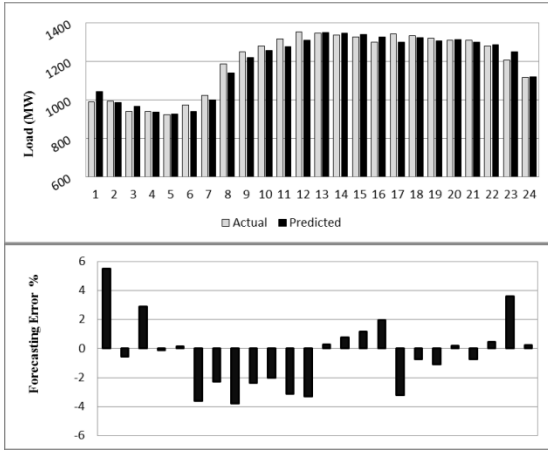
(d) Thursday



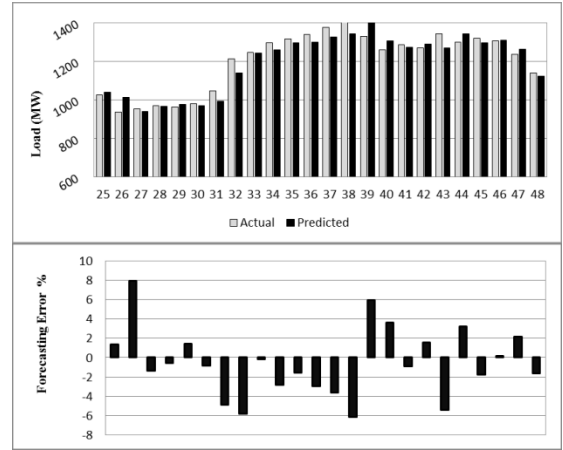
(e) Friday

Figure 68: Week ahead forecast results for CRNG

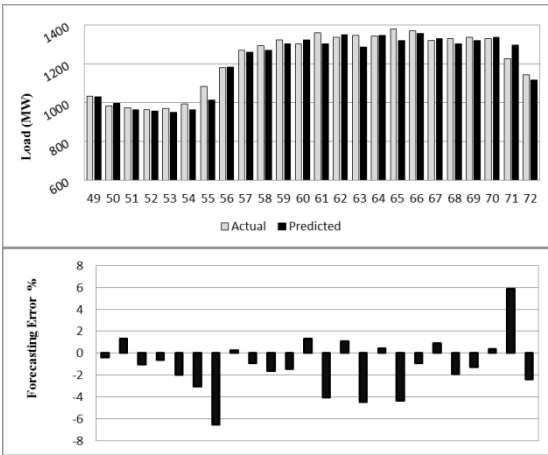
The performance of CRNNG is very similar to RNNG but its performance during peak hours is better. Error ranges from -8.22% to 9.93%. The error distribution is much better than RNNG along with the error performance during starting hours of the day. Since the error peaks are quite high the overall performance has suffered and resulted in a MAPE of 2.26%.



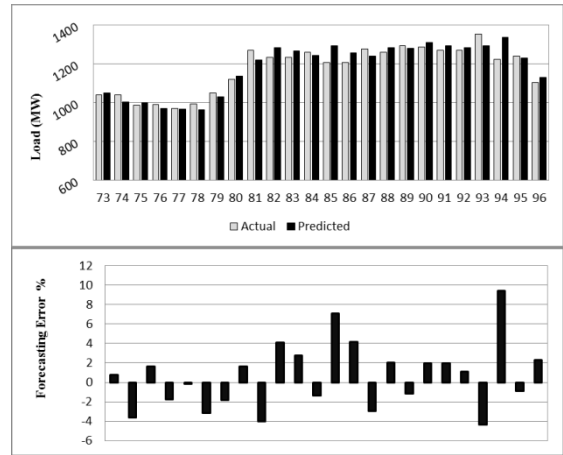
(a) Monday



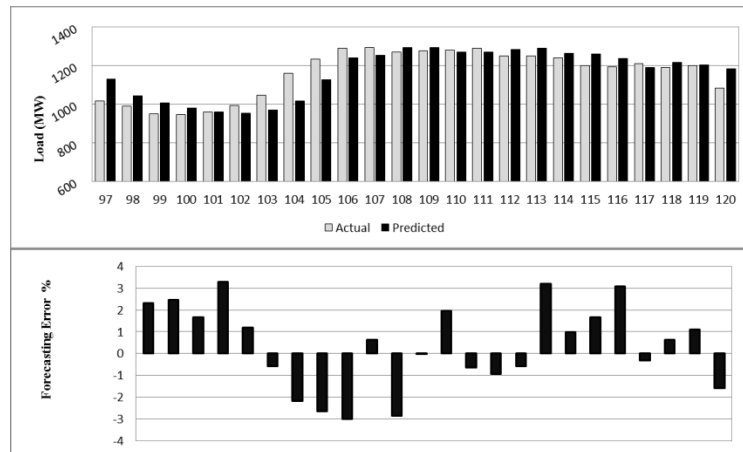
(b) Tuesday



(c) Wednesday



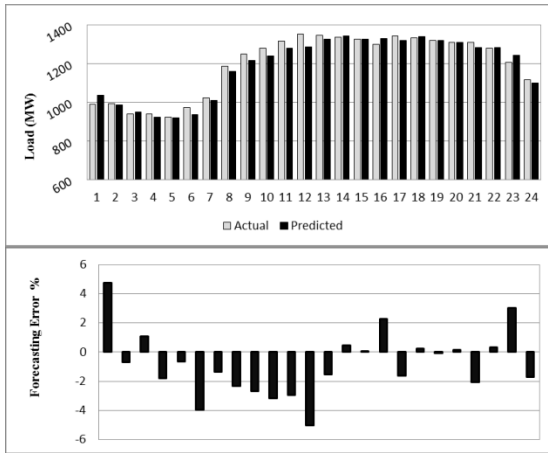
(d) Thursday



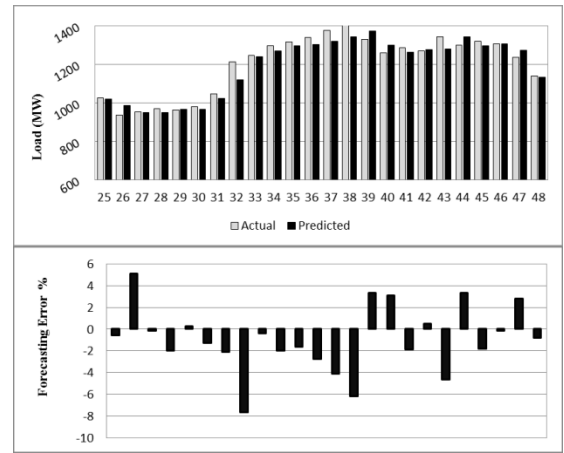
(e) Friday

Figure 69: Week ahead forecast results for 3-layered FFNN

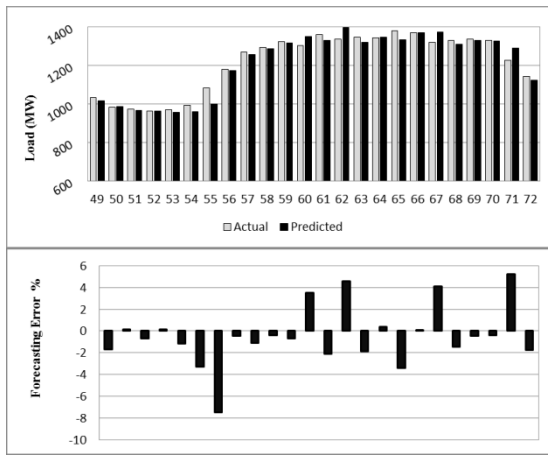
Though most of the errors are negative, performance throughout remains almost the same with very few abnormal error spikes in error distribution. With a MAPE of 2.21% and very fast training this network can be preferred over RNNL. Its errors falls in between -6.52% and 9.4% with error getting as low as a minimum of 0.028%.



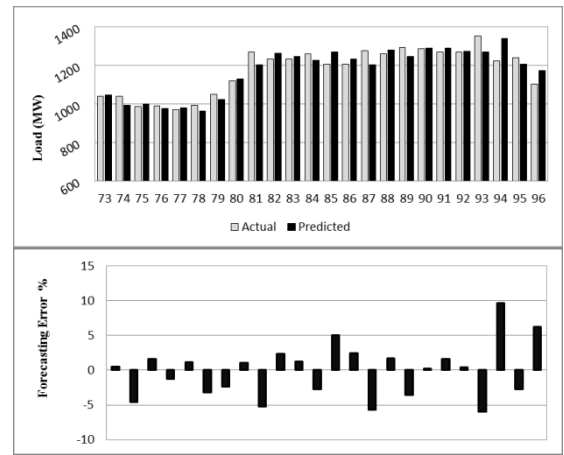
(a) Monday



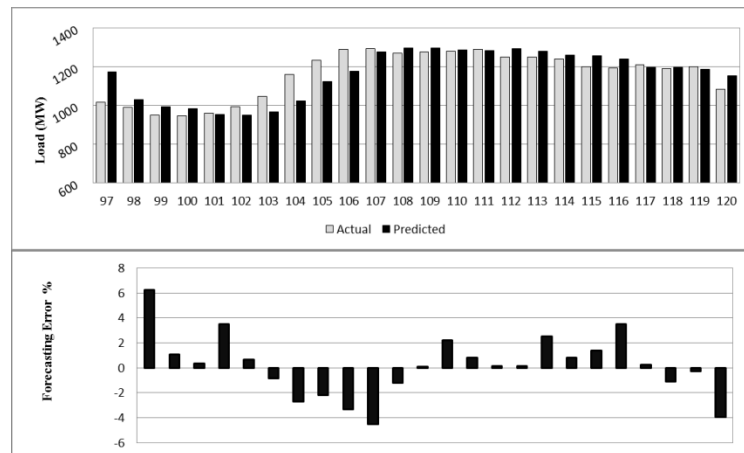
(b) Tuesday



(c) Wednesday



(d) Thursday



(e) Friday

Figure 70: Week ahead forecast results for 3-layered CNN

The output of the 3-layered CNN is presented above. The predicted and actual load curves match closely with a very stable error curve. MAPE is 2.18% which is a touch higher than CNN (2.12%). MAPE has gone high due to occurrence of high error peaks. Error ranges from -7.62% to 9.6% with minimum error of 0.02%. Due to higher accuracy the CNN may be preferred over its 3-layered version.

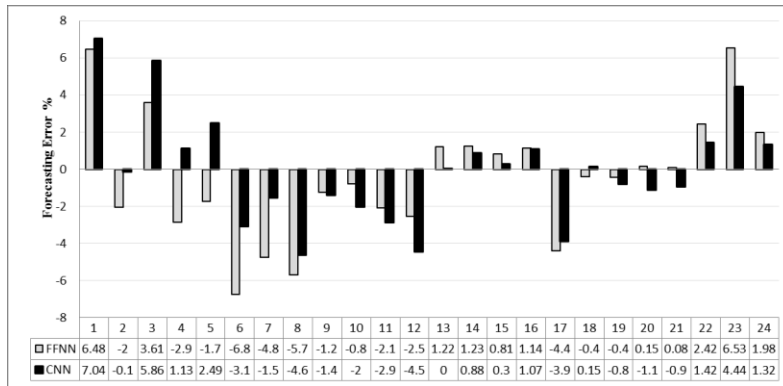
5.2.4 Comparative Analysis of Week ahead forecast for Weekdays

The overall performance of errors for the week ahead NB load forecast for weekdays is presented in Table 8. The performance of FFNN produces the largest MAPE so an improvement of all the other networks over it is also given.

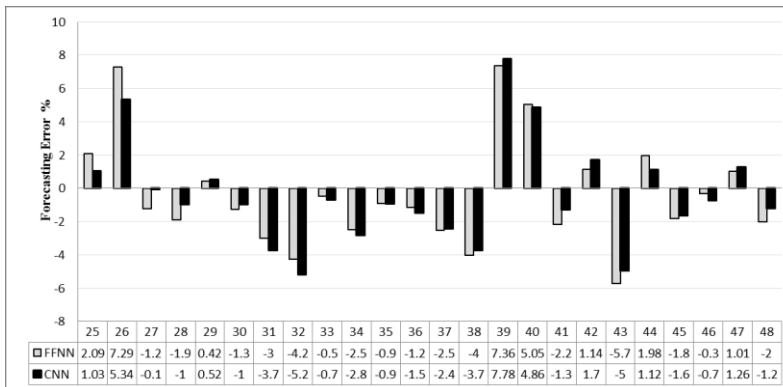
Table 8: Week ahead MAPE and RMSE for New Brunswick

Architecture	MAPE	RMSE	Improvement
FFNN	2.31	35.36	0.00%
CNN	2.12	33.43	8.18%
RNNL	2.20	34.23	4.48%
RNNG	2.27	34.76	1.76%
CRNNL	2.24	35.05	2.96%
CRNNG	2.26	35.33	2.12%
FFNN (3L)	2.21	34.28	4.29%
CNN (3L)	2.18	34.95	5.67%

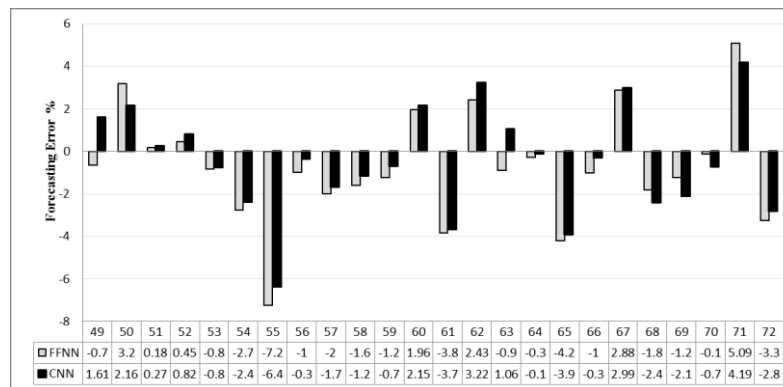
Among the eight networks the performance of the CNN stands out with an improvement of 8.18% over the FFNN. The 3-layered version of the CNN has also proved to be accurate with an improvement of 5.67%. The results of the RNNL (4.48%) and the 3-layered FFNN (4.29%) are very similar but preference should be given to the 3-layered FFNN among them due to its fast convergence. All remaining architectures do not offer significant enhancement in their forecasts and also due to slower convergence than FFNN cannot be preferred over it. Moreover a comparison of forecasting performance between the FFNN and the CNN is given in Figure 73 which can clearly show the error suppression done by CNN.



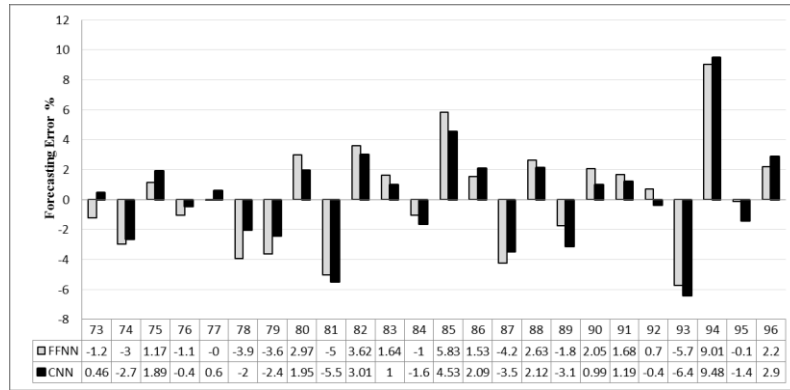
(a) Monday



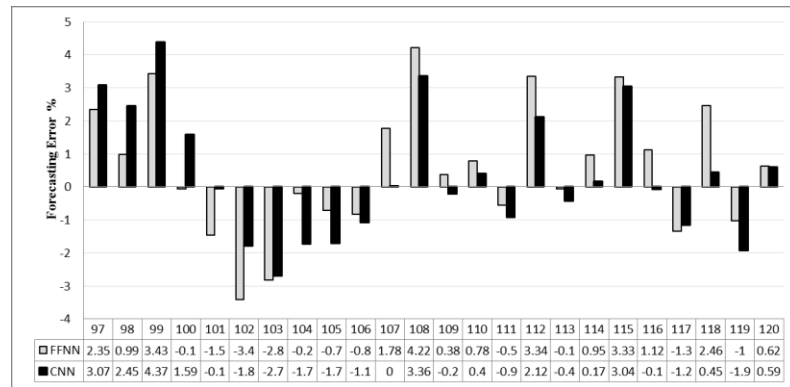
(b) Tuesday



(c) Wednesday



(d) Thursday



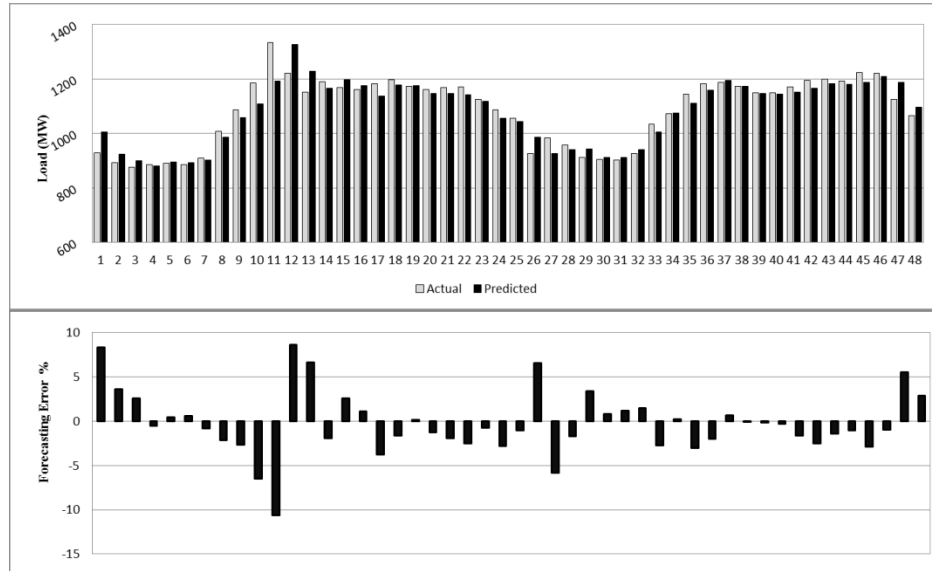
(e) Friday

Figure 71: Week ahead forecasting error performance comparison between FFNN and CNN for New Brunswick

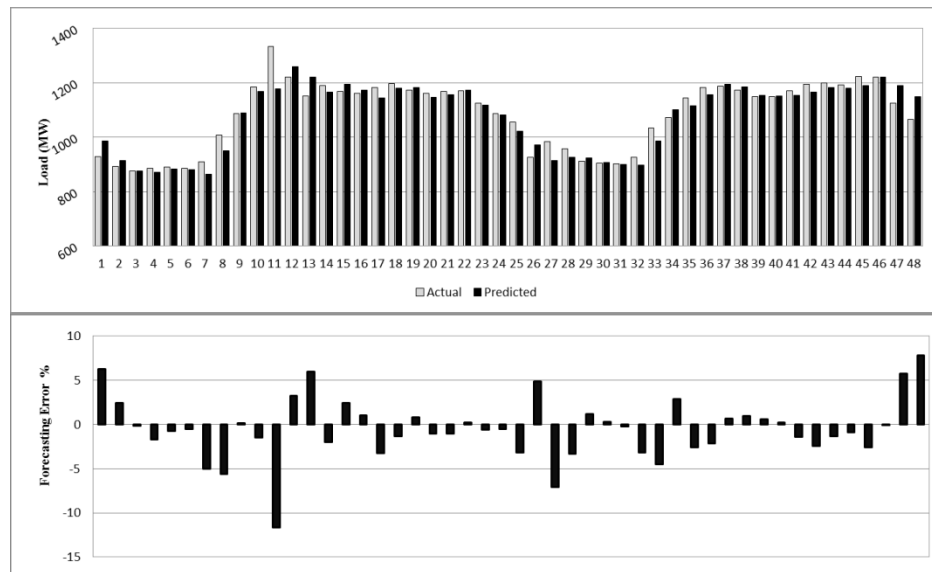
5.2.5 Load forecast for Weekend

The load forecast for the weekend for New Brunswick is discussed in this section.

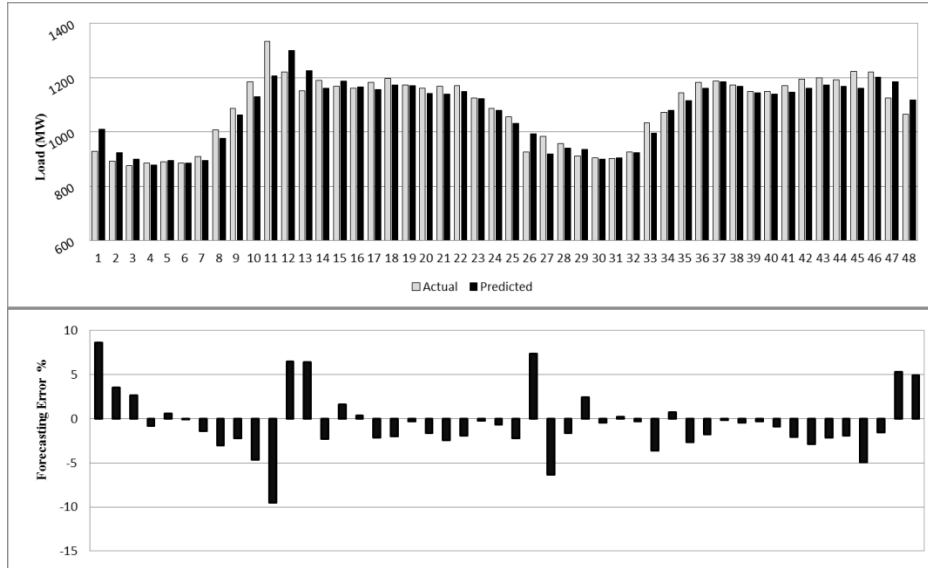
The same days as those selected for Nova Scotia are selected for the forecast. The performance graphs of each network tested is given and explained. The results of the first four architectures; FFNN, CNN, RNNL and RNNG are presented in Figure 74.



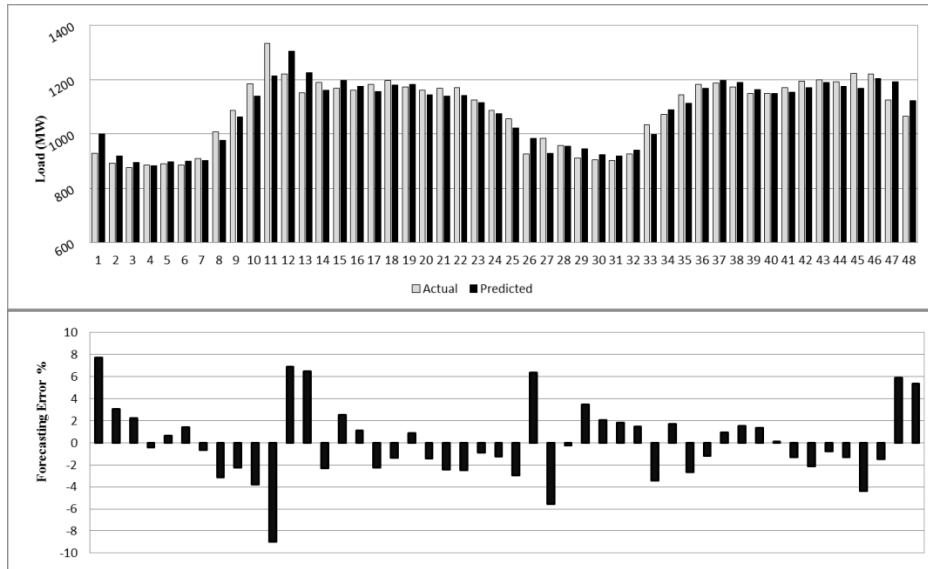
(a) FFNN



(b) CNN



(c) RNNL

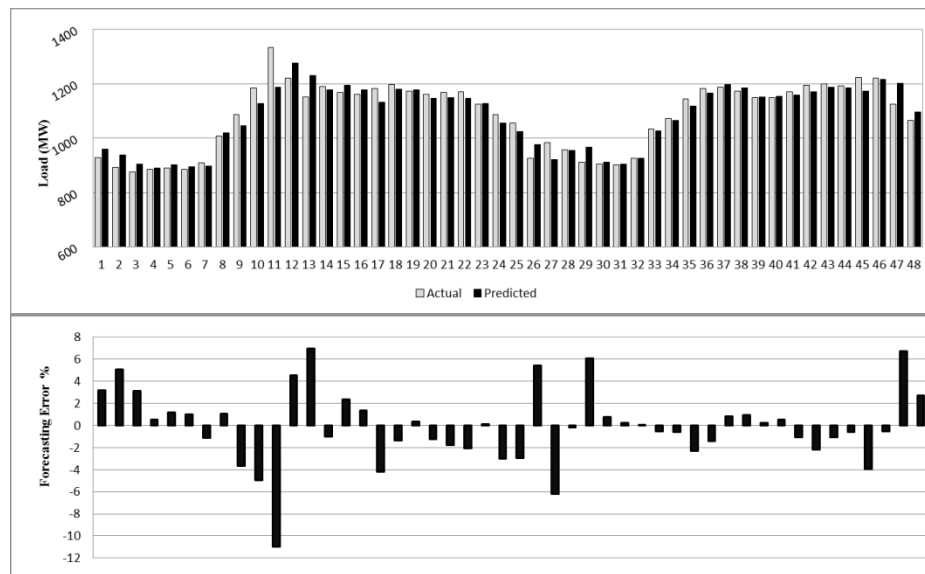


(d) RNNG

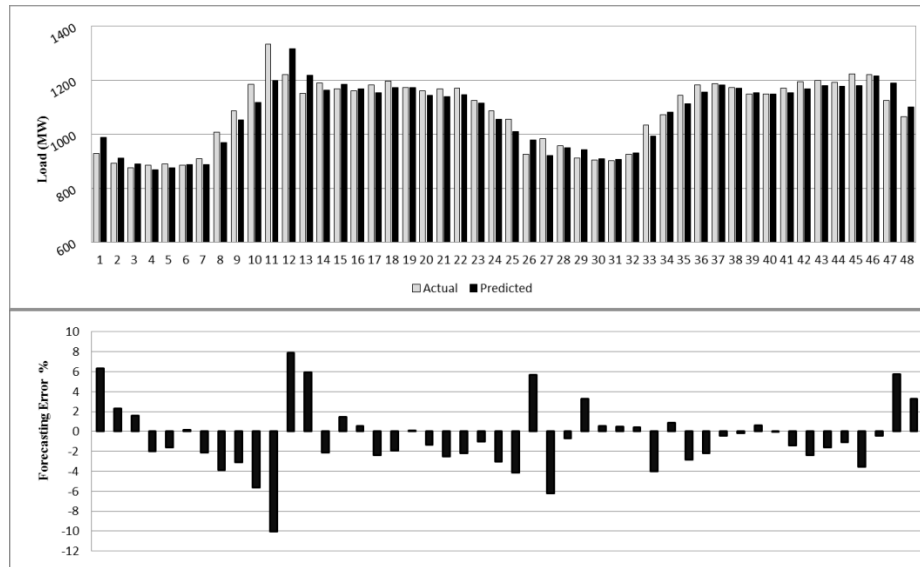
Figure 72: Weekend forecast results (FFNN, CNN, RNNL and RNNG)

Due to a drastic increase in the load demand during 10th hour on Saturday the forecast accuracy has suffered for all the networks. The FFNN in particular produces a MAPE of 2.59%. The error ranges from -10.65% to 8.65%. The error decreases after the occurrence of demand spike. The CNN on the other hand improves the overall performance with MAPE 2.48% but generates a big negative error of -11.69% at the 10th

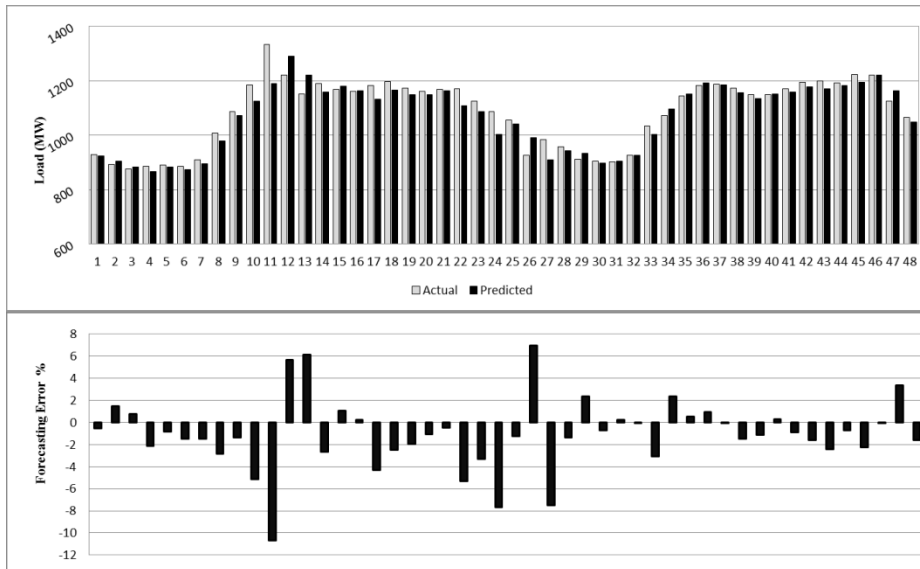
hour. The error response stabilizes and improves the forecast of rest of the hours. The maximum positive error of 7.83% occurs for last hour of Sunday. The performance of RNNL and RNNG are very similar to FFNN with MAPEs of 2.55% and 2.62% and the error ranges of -9.55% to 8.63% and -8.97% to 7.74%. The error curve of the RNNG does not come near to zero for the most of the time due to which MAPE has suffered. The RNNL on the other hand produces high error peaks almost at the same hours as the RNNG but its error curve remains near to zero for most part which have resulted in a reduced MAPE. The graphs for forecast produced by CRNNL, CRNNG, 3-layered FFNN and 3-layered CNN are discussed next.



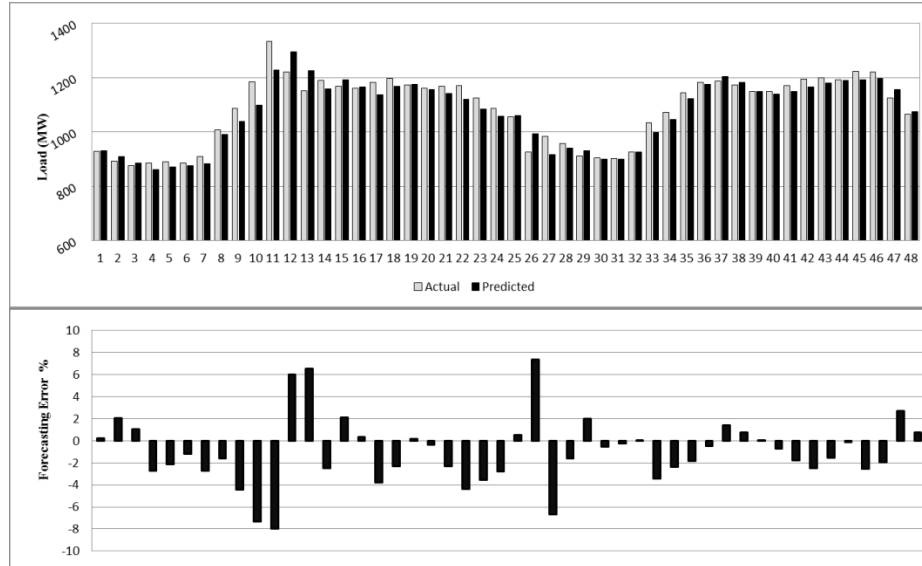
(a) CRNNL



(b) CRNNG



(c) FFNN (3L)



(d) CNN (3L)

Figure 73: Weekend forecast results: (CRNNL, CRNNG, FFNN (3L) and CNN (3L))

The error curves of both hybrid networks CRNNL and CRNNG look almost the same with similar sort of peaks occurring at same hours. CRNNL produces MAPE of 2.37% and CRNNG 2.56%. For the most part of the forecast the error curve of the CRNNL remain close to zero. Whereas during the starting hours the error curve of CRNNG fall towards negative side resulting in decreased performance. The error for the CRNNL varies from -10.97% to 6.93% and that of the CRNNG from -10.01% to 7.91%. Furthermore the relationship between the error curves of both the 3-layered networks is much alike the hybrid ones. Both error curves closely resemble each other at almost all the points. The MAPE of the 3-layered FFNN is 2.38 and 3-layered CNN is 2.39%. The error curve of the former regularly stays near zero with presence of few error peaks. Its error range is from -10.7% to 6.96%. The error distribution of 3-layered CNN looks very unstable but remains in a close range of -7.95% to 7.37%.

5.2.6 Comparative Analysis of Load forecast for Weekend

For New Brunswick weekend forecast RNNG generates the maximum error. Hence performance of each network is portrayed as an improvement over it. Table 9 summarizes the load forecasting performances of all the networks.

Table 9: Weekend forecast MAPE and RMSE for New Brunswick

Architecture	MAPE	RMSE	Improvement
FFNN	2.59	40.32	1.30%
CNN	2.48	38.89	5.38%
RNNL	2.55	38.42	2.72%
RNNG	2.62	37.64	0.00%
CRNNL	2.37	37.60	9.52%
CRNNG	2.56	38.48	2.52%
FFNN (3L)	2.38	38.23	9.42%
CNN (3L)	2.39	35.74	9.16%

Due to the rapid increase in demand during the first half of Saturday the forecast produced by all the networks is affected. Overall, no network performs better than their performance for the weekdays. CRNNL improves the results 9.52% over the RNNG. Other networks with noticeable performance are the 3-layered networks with the FFNN version improving by 9.42% and the CNN version improving 9.16%. CNN, RNNL, CRNNG and FFNN also produces better results with improvements of 5.38%, 2.72%, 2.52% and 1.3%. A comparison of forecasting error between the RNNG and the CRNNL is given below which highlights the improvement of forecasting results of the CRNNL over the RNNG.

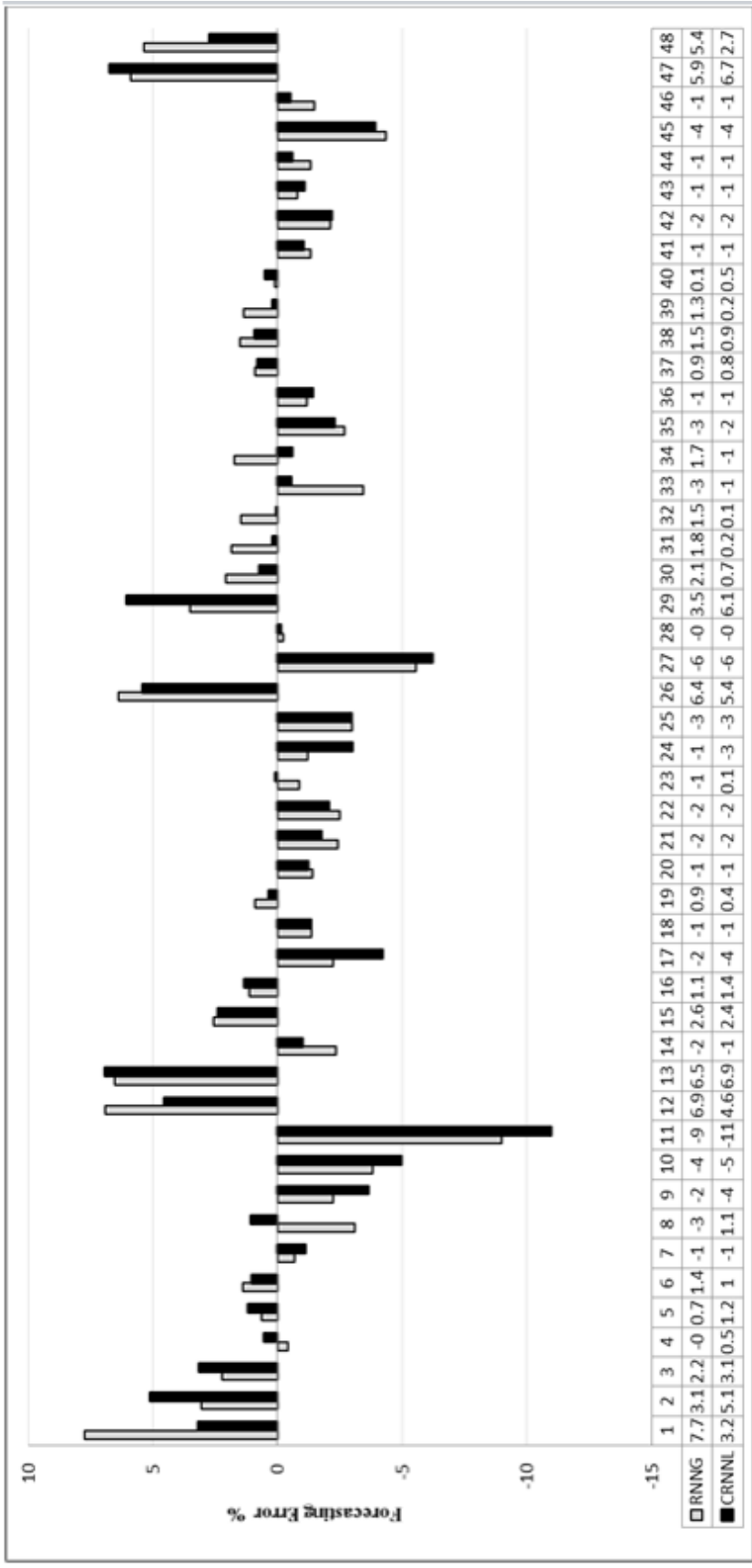


Figure 76: Weekend forecasting error performance comparison between RNNG and CRNNL for New Brunswick

5.2.7 Summary of Load forecast results for New Brunswick

Table 10 summarizes all the forecasting results achieved for New Brunswick followed by a graphical presentation.

Table 10: Summary of Load forecast results for New Brunswick

Architecture	Day ahead	Week ahead	Weekend
FFNN	2.30	2.31	2.59
CNN	1.96	2.12	2.49
RNNL	2.14	2.20	2.56
RNNG	2.16	2.27	2.63
CRNNL	2.14	2.24	2.38
CRNNG	2.21	2.26	2.56
FFNN (3L)	1.93	2.21	2.38
CNN (3L)	1.89	2.18	2.39

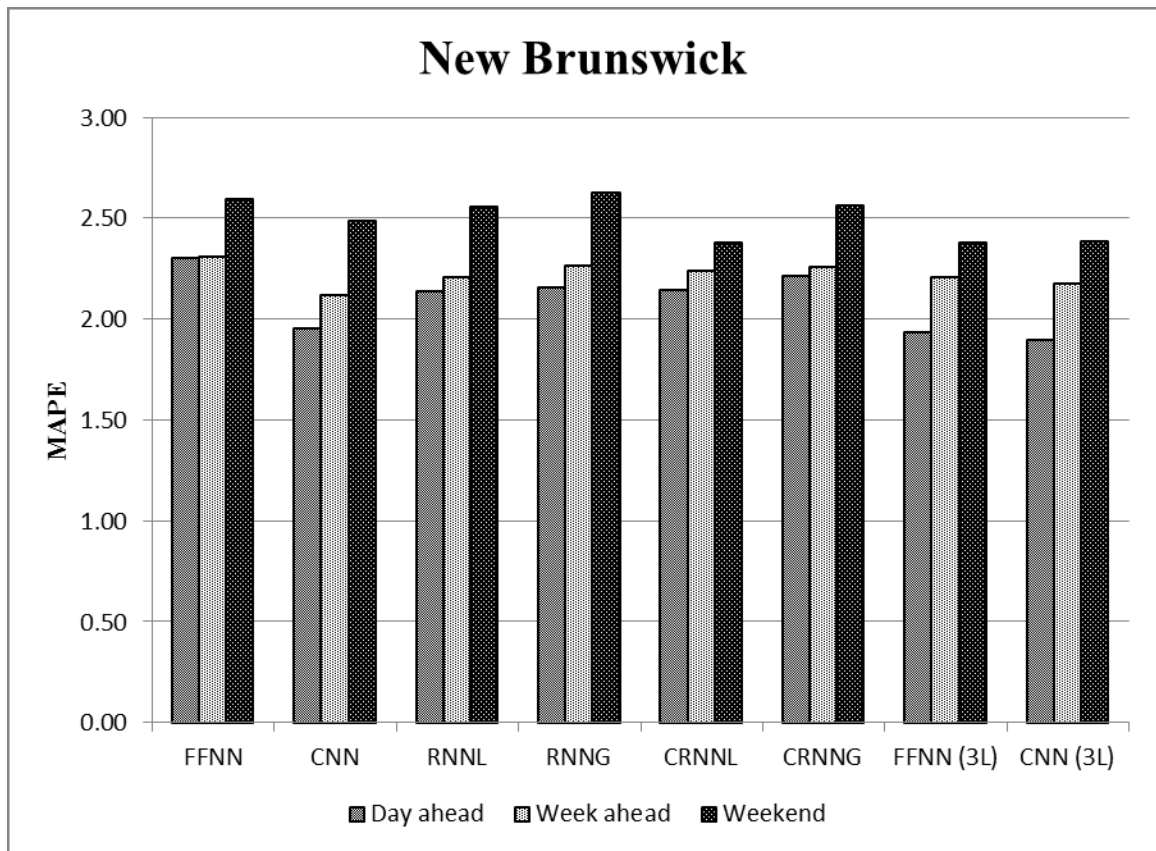


Figure 74: Graphical presentation of load forecast results for New Brunswick

5.3 Ontario

The electrical load demand of Ontario varies from 10816 MW to 24927 MW during June to August 2013. Ontario is an industrialized province so the load demand drops down a significant amount during the weekend. The load and temperature data from June 1 to August 25, 2013 are used for training all the networks just like the other two jurisdictions. This data was gathered from online resources broadcasted by Government of Canada, climate database [64] and Independent Electricity System Operator (IESO), Power Data [67].

5.3.1 24 hours ahead forecast for Weekdays

The 24 hours ahead ON load forecast using eight network architectures of section 4.4 are given for a weekday. The day selected for forecast is August 26, 2013. Since the load demand for Ontario is very high a percent in error will weigh more than a percent of error for Nova Scotia or New Brunswick. Following figures present forecasts for all the networks.

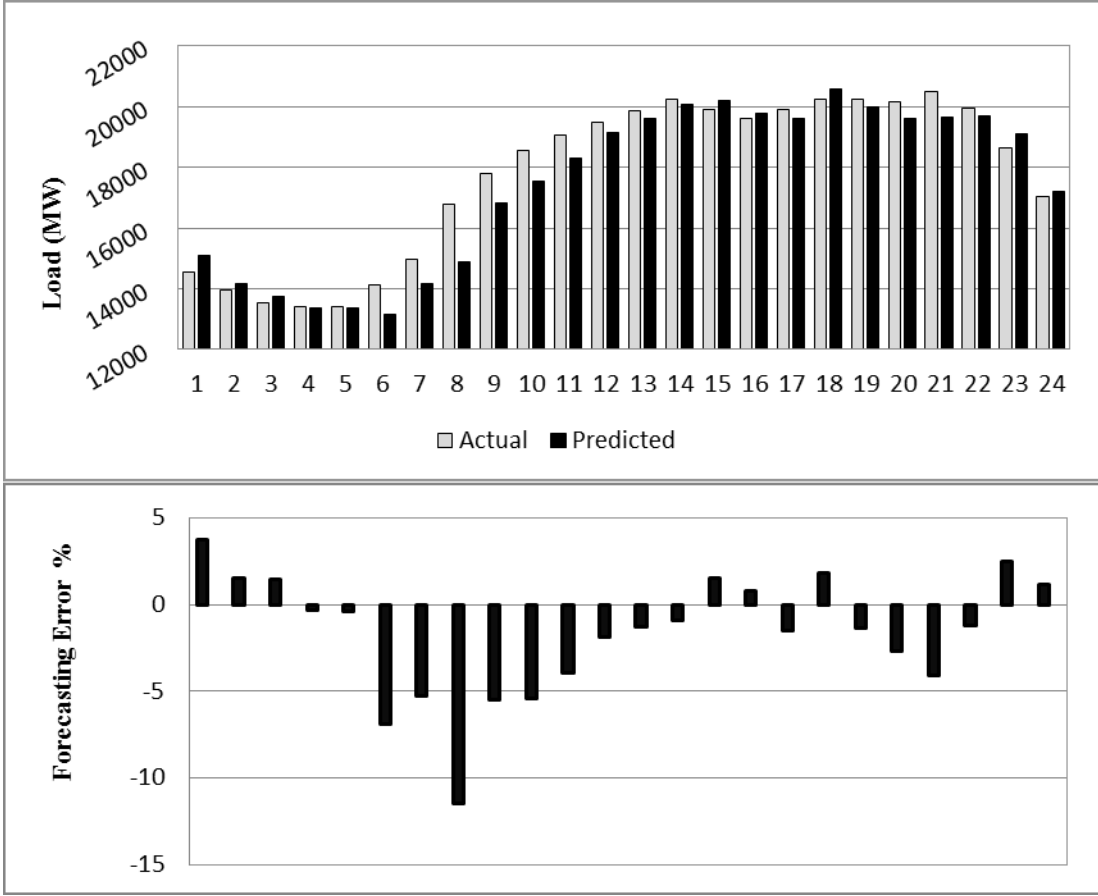


Figure 75: Day ahead forecast results for FFNN

The load forecast for Ontario using FFNN is shown above. It can be seen that prediction falls short of actual load during most part of the forecast. The error decreases during peak hours which is a big advantage considering high total demand. The error ranges from -11.43% to 3.76% and it falls to a minimum of 0.3% during peaks hours. The overall MAPE is 2.86% mainly due to high negative error.

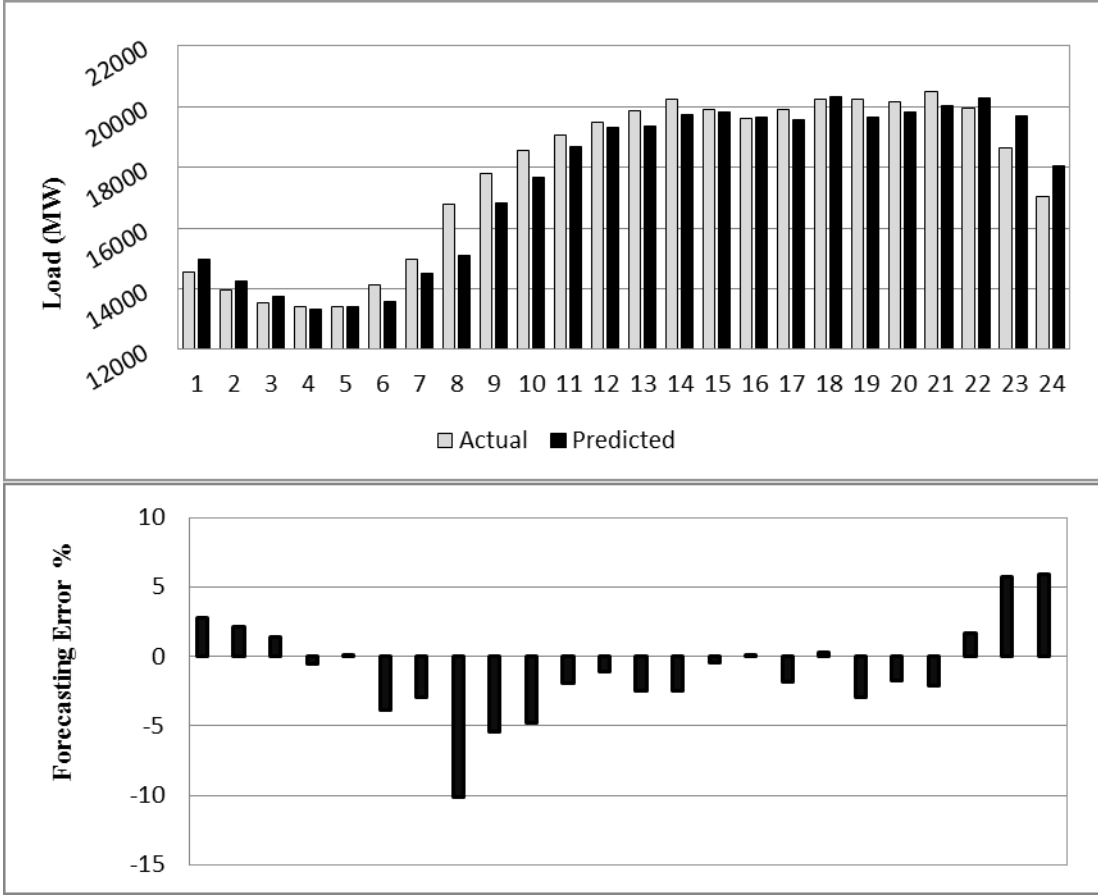


Figure 76: Day ahead forecast results for CNN

The load forecast generated by CNN is given above. The negative error is greatly suppressed. Throughout the day, predicted curve has remained below actual load which is not acceptable for a big industrialized province like Ontario. The error ranges from -10.1% to 5.91%. Even during the peak hours error curve remains in the negative axis. The overall MAPE is 2.7%.

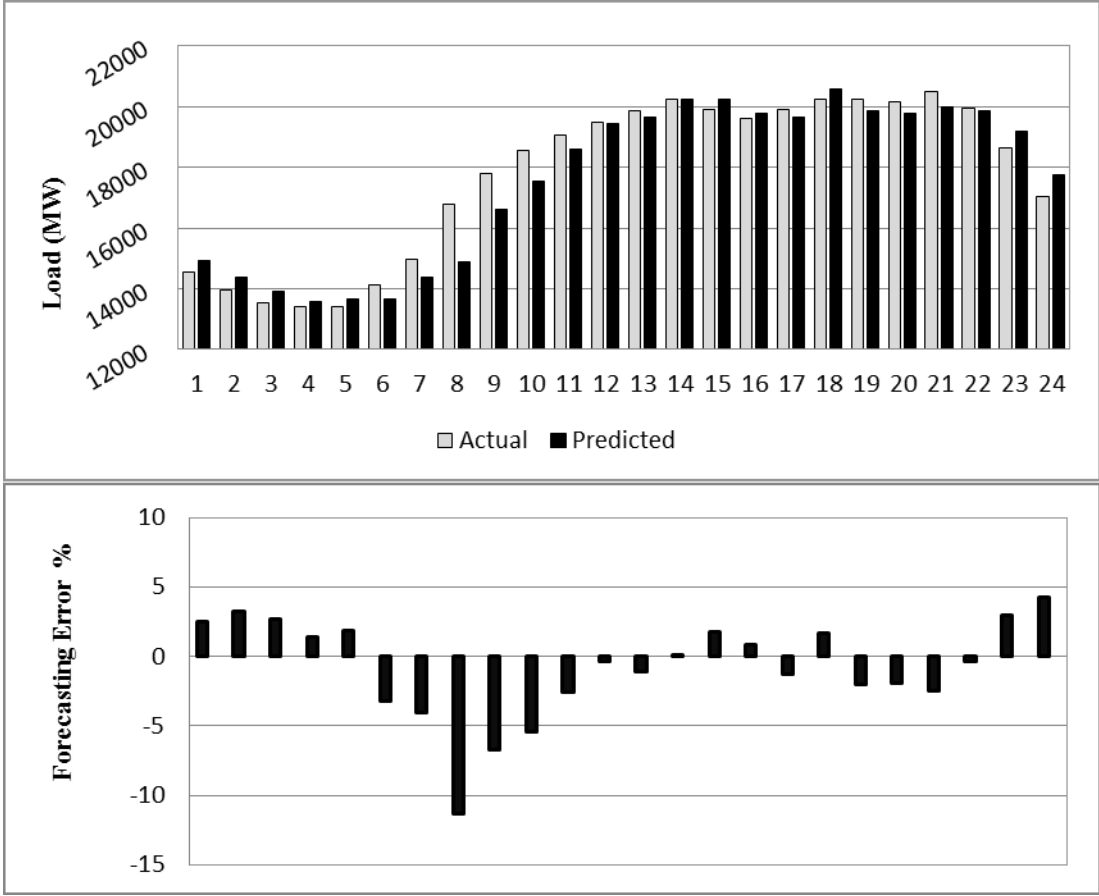


Figure 77: Day ahead forecast results for RNNL

RNNL produces a MAPE of 2.75% in its forecast. Though negative error in first half of the day resembles from FFNN forecast but performance during peak hours is greatly improved. MAPE has suffered due to high negative peak of -11.29%. The positive error peak is 4.31%. During peak hours the error curve becomes steady and the error falls as low as 0.06%.

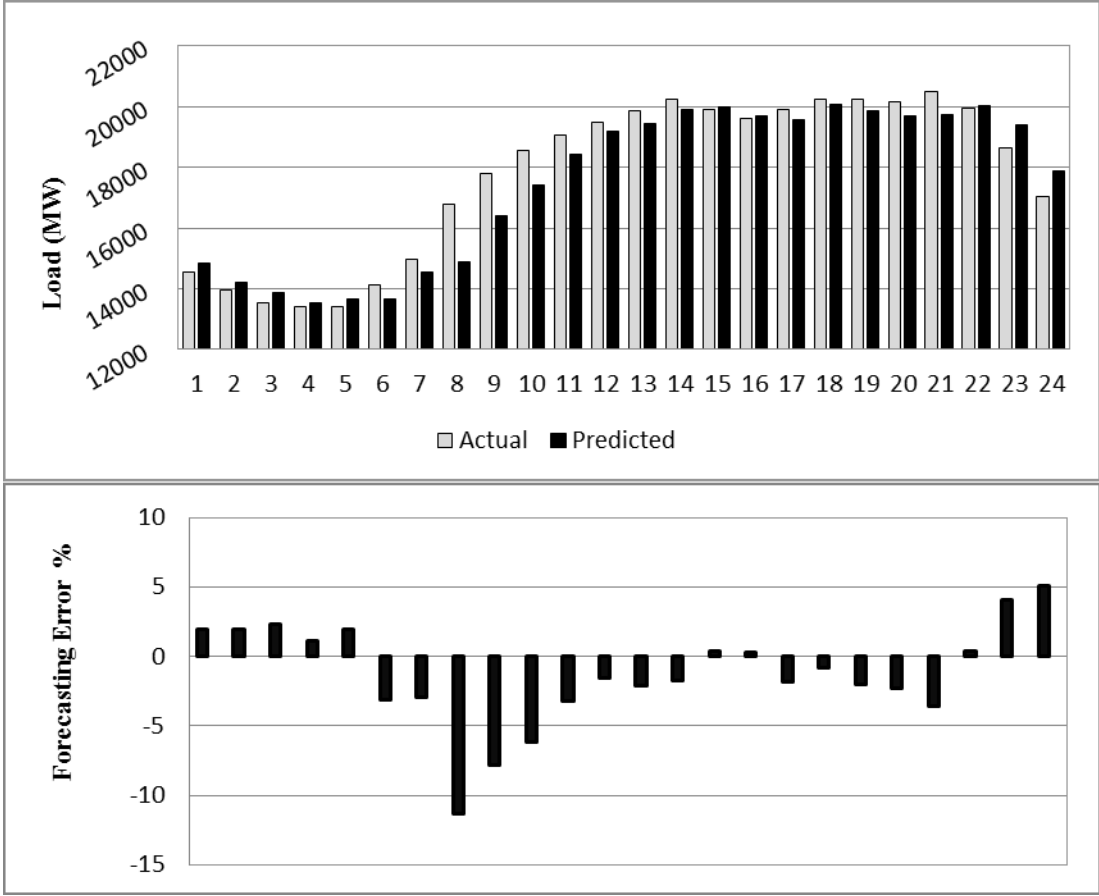


Figure 78: Day ahead forecast results for RNNG

RNNG gives inferior forecasting results than FFNN with the MAPE of 2.92%. The predicted load remains below actual demand. The error range is from -11.32% to 5.12%. Considering training time this network is not preferred for application.

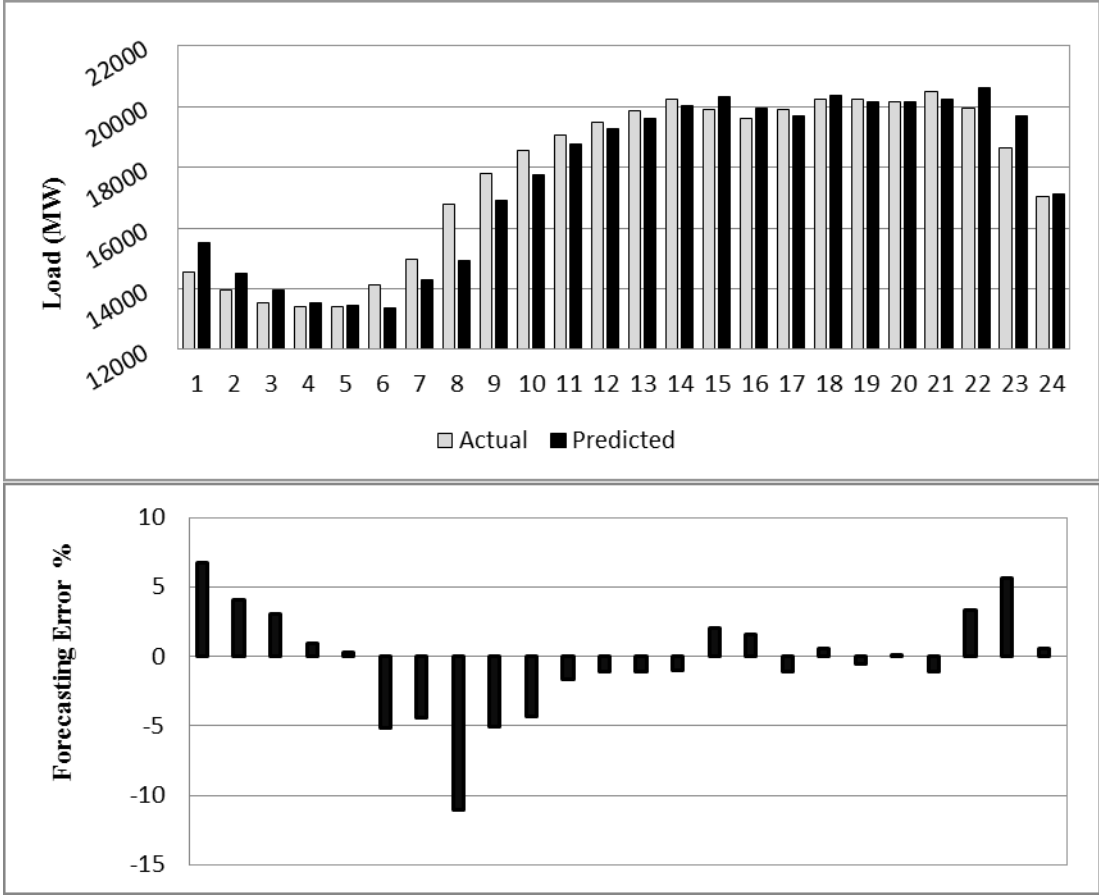


Figure 79: Day ahead forecast results for CRNNL

The CRNNL offers good improvement over FFNN. With a MAPE of 2.77% it gives accurate results specially during peak hours. Though the negative error during first half is not greatly minimized and the negative error peak at -11%, due to accurate forecast during peak hours the overall performance is improved. The minimum error goes as low as 0.01% and the positive error peaks at 6.72% which occurs for the first hour.

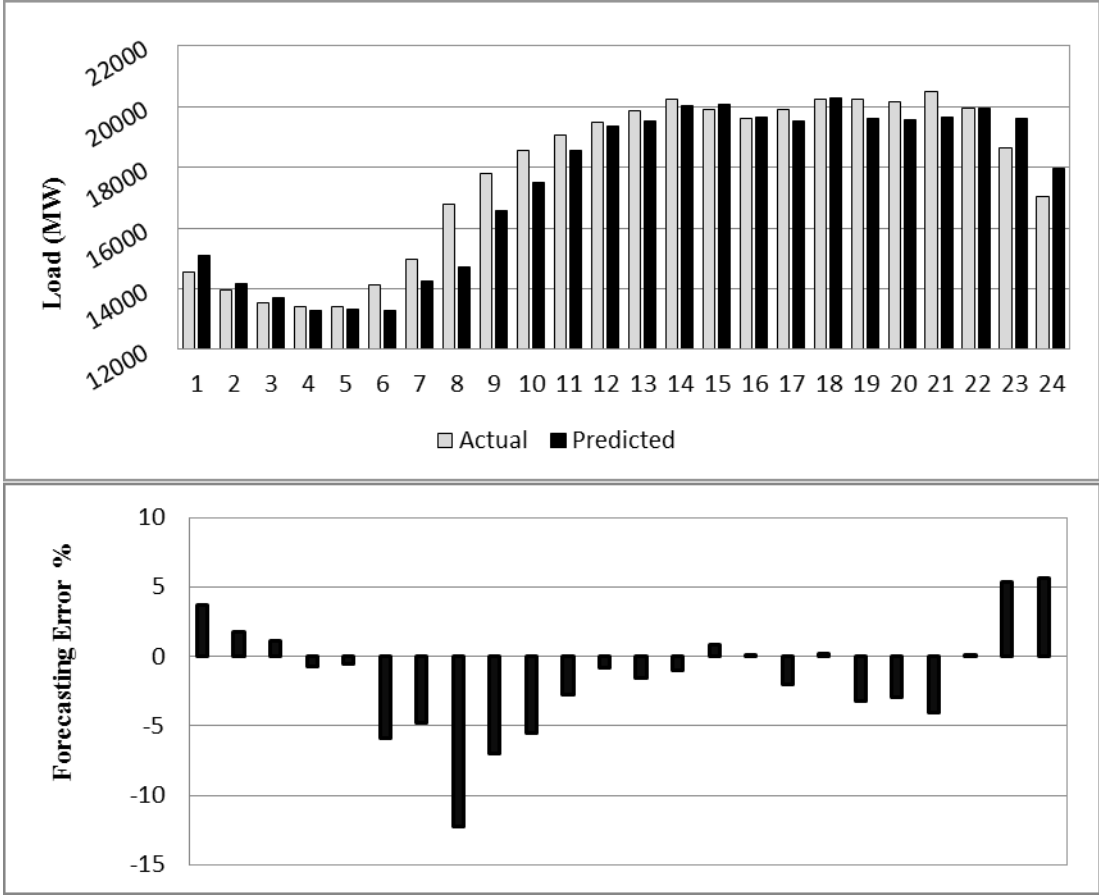


Figure 80: Day ahead forecast results for CRNNG

The CRNNG produces poorest performance of all the networks. Its negative peak goes to -12.26% and positive error peak is 5.64%. Like RNNG the predicted curve remain under demand curve for the most of the forecast period.

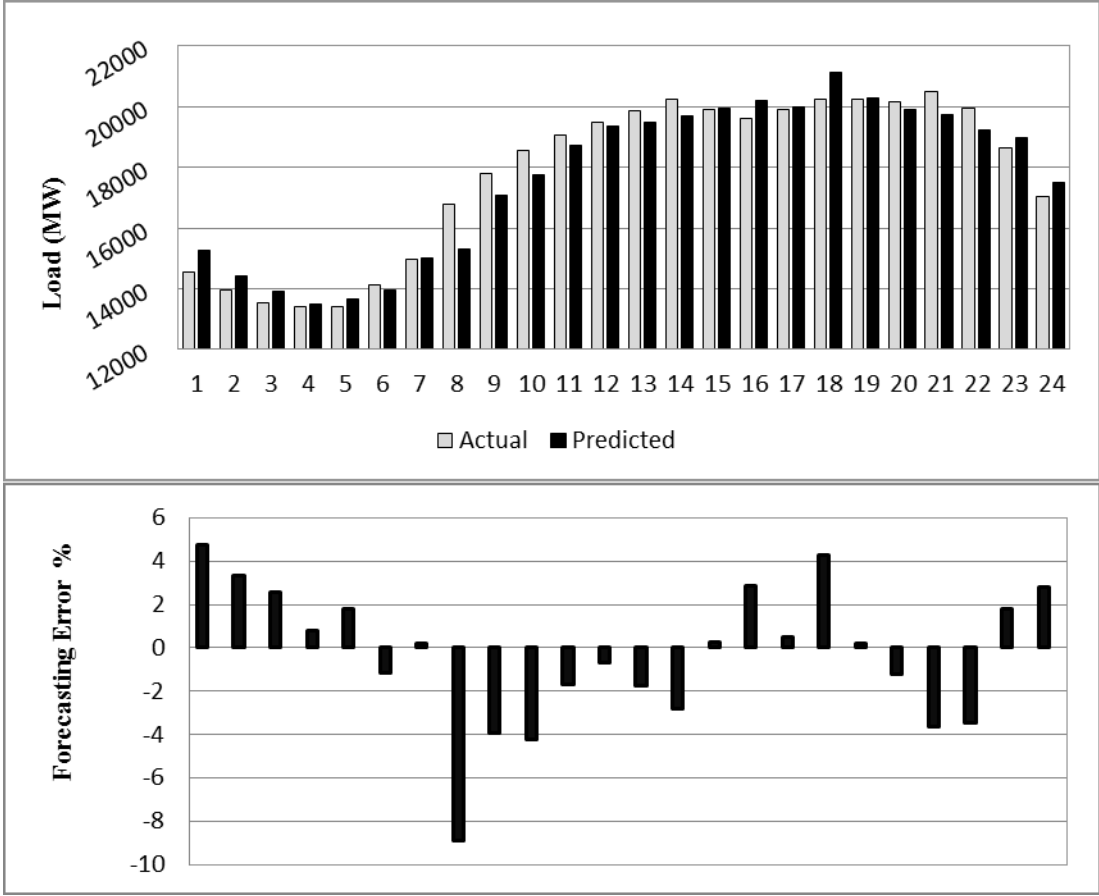


Figure 81: Day ahead forecast results for 3-layered FFNN

The 3-layered FFNN greatly reduces negative error during first half of the day. During peak hours few error spikes occur but the error varies from -8.86% to 4.73%. It means that predicted curve closely follows the load curve and produces a MAPE of 2.48%.

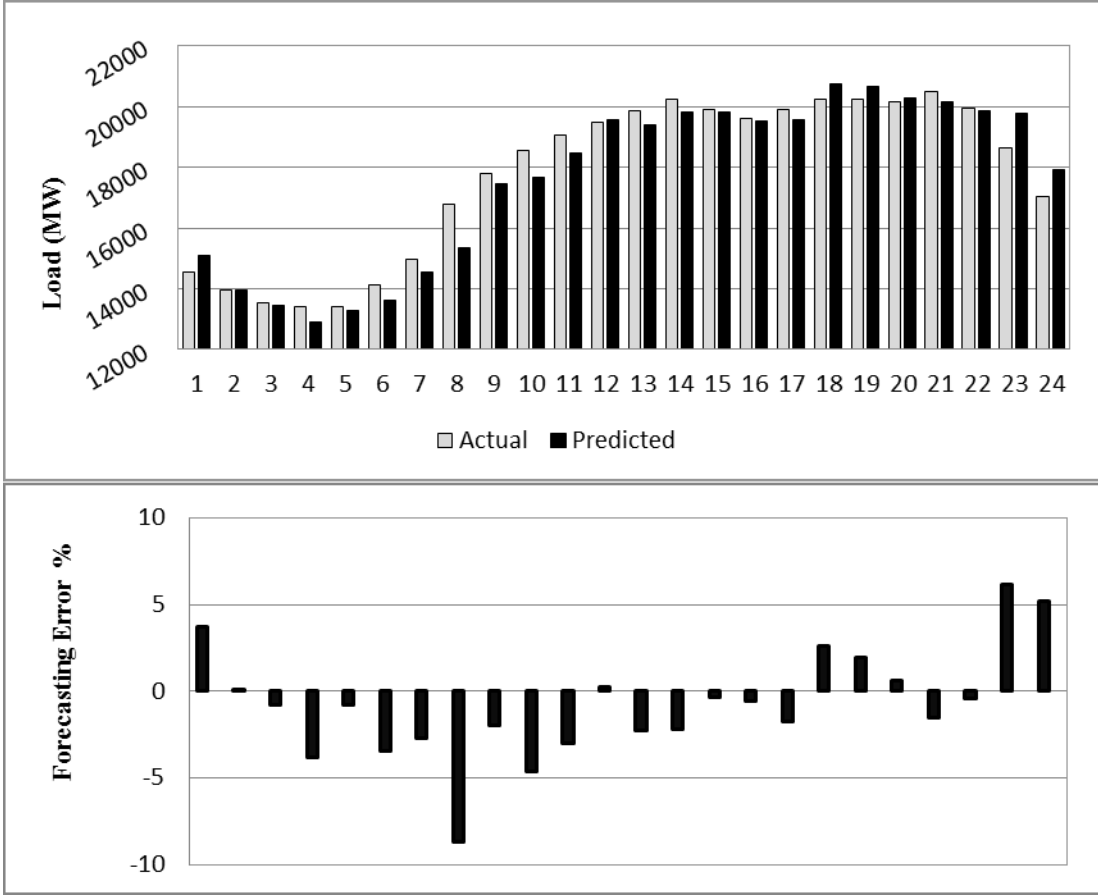


Figure 82: Day ahead forecast results for 3-layered CNN

3-layered CNN produces best error distribution among all the networks tested. Its MAPE is 2.48% which is same as 3-layered FFNN. The negative error during first half is not that greatly suppressed but forecasts made during peak hours is more accurate than 3-layered FFNN. Its error ranges from -8.65% to 6.18% and error falls as low as 0.05% during peak hours. With fast convergence this network can be preferred over all the other networks.

5.3.2 Comparative Analysis of 24 hours ahead forecast for Weekdays

For Ontario the worst performing network is proved to be CRNNG. The performance of all the other networks is presented in the table below with an improvement of each over CRNNG.

Table 11: 24 hours ahead MAPE and RMSE for Ontario

Architecture	MAPE	RMSE	Improvement
FFNN	2.86	649.48	7.31%
CNN	2.70	618.90	12.33%
RNNL	2.75	619.19	10.75%
RNNG	2.92	668.77	5.31%
CRNNL	2.77	628.70	10.09%
CRNNG	3.08	719.32	0.00%
FFNN (3L)	2.48	551.70	19.40%
CNN (3L)	2.48	558.22	19.52%

The recurrent networks having global feedbacks CRNNG and RNNG perform poorly for Ontario. Though RNNG offers 5.31% enhancement over CRNNG but due to the slow convergence might not be a good choice. FFNN (7.31%) has proven to be better than the recurrent networks with global feedbacks. The other two recurrent networks that have a local feedback have generated superior results than FFNN. RNNL improves accuracy by 10.75% and CRNNL by 10.09% over CRNNG. Both cascaded architectures and the 3-layered FFNN have once more proved to be the best of the lot with CNN improving 12.33%, 3-layered FFNN 19.40% and 3-layered CNN 19.52%. The most recommended architecture is 3-layered CNN whose accuracy is almost equivalent to 3-layered FFNN but performs better during peak hours. Also, a comparison of forecasting error between CRNNG and 3-layered CNN is given as follows:

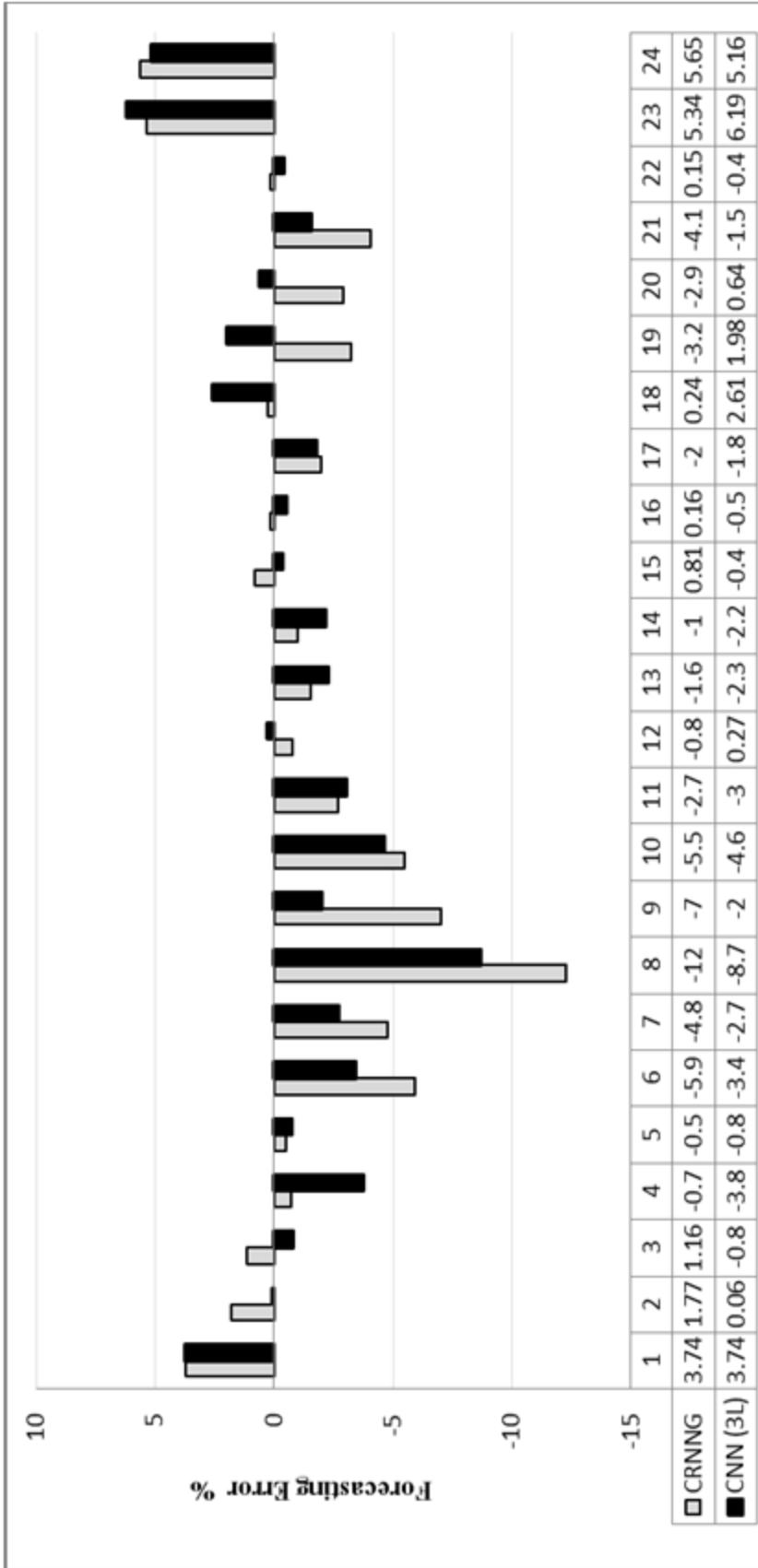
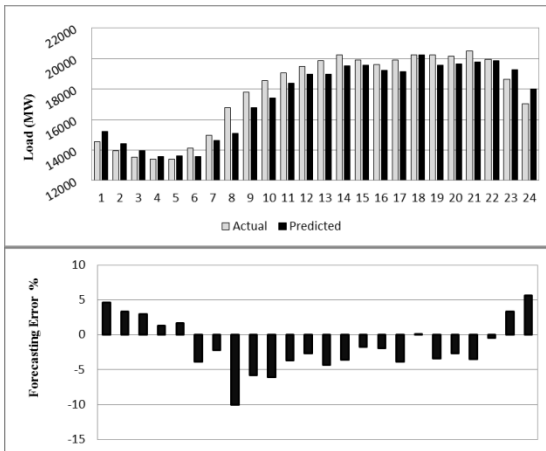


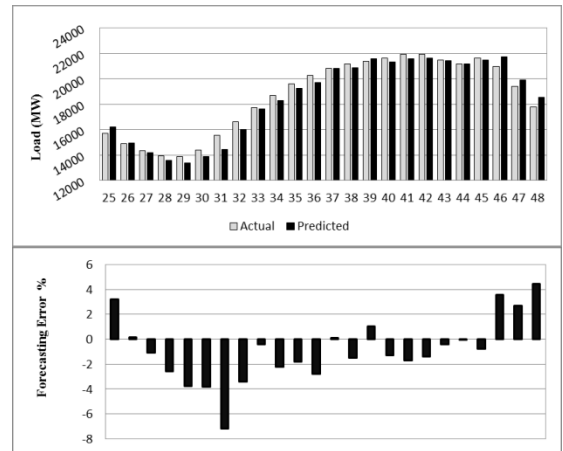
Figure 86: 24 hours ahead forecasting error performance comparison between CRNNG and CNN (3L) for Ontario

5.3.3 Week ahead forecast for Weekdays

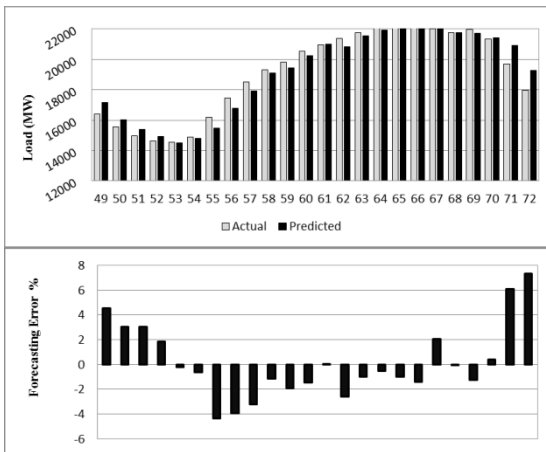
The week ahead ON load forecast for weekdays is explained in this section. Same week for the forecast as selected Nova Scotia and New Brunswick is selected. The performance of eight networks is elaborated by the graphs below:



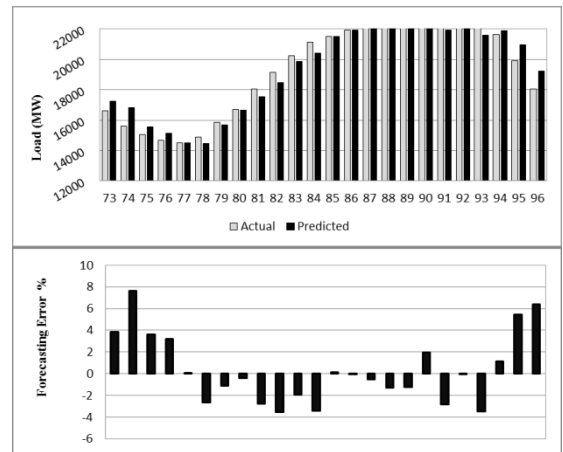
(a) Monday



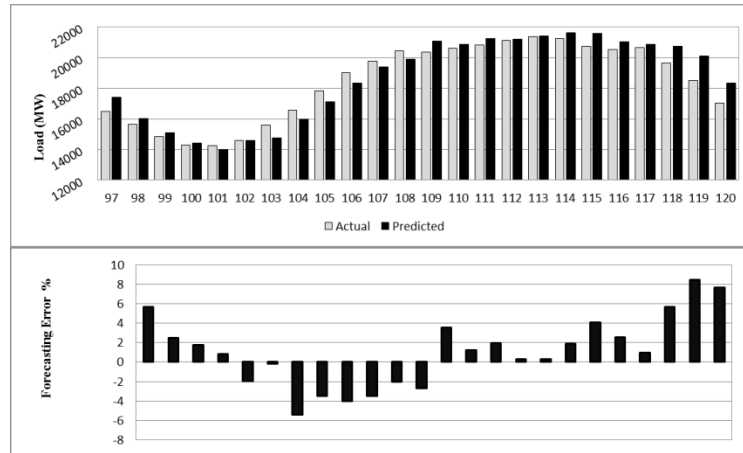
(b) Tuesday



(c) Wednesday



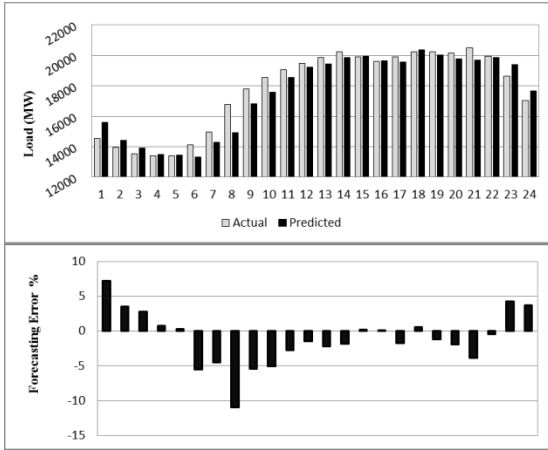
(d) Thursday



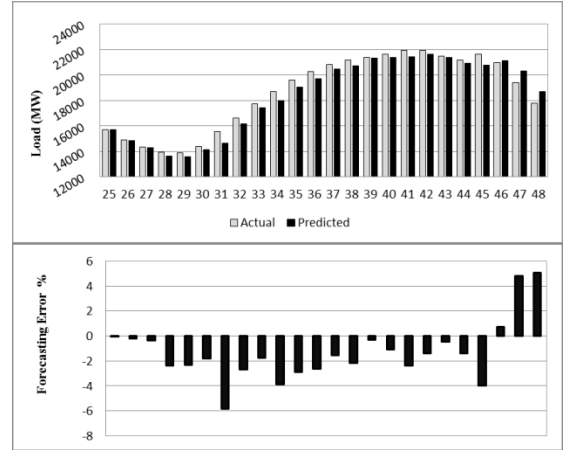
(e) Friday

Figure 83: Week ahead forecast results for FFNN

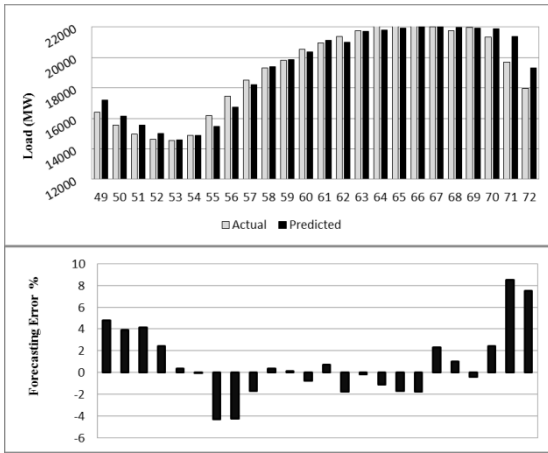
The forecast produced by FFNN is shown above. The forecast for Monday and Friday is inaccurate. For three days in middle, the predicted and demand curves are quite close to each other. The MAPE for this network is 2.65% with error ranging from -10.04% to 8.49%. The minimum error also falls as low as 0.009%.



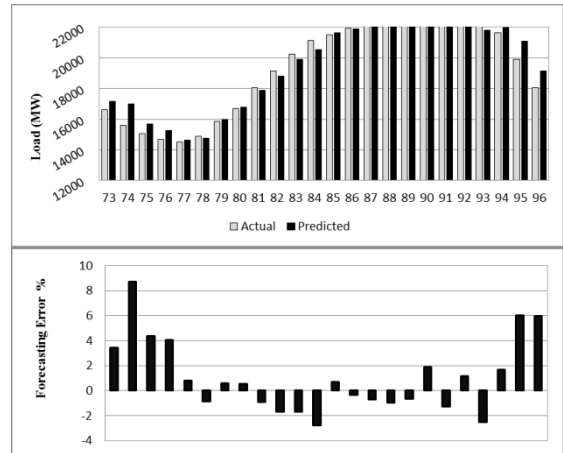
(a) Monday



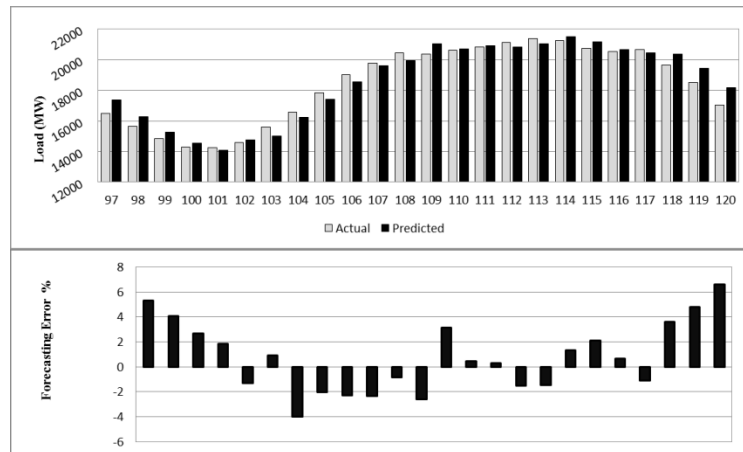
(b) Tuesday



(c) Wednesday



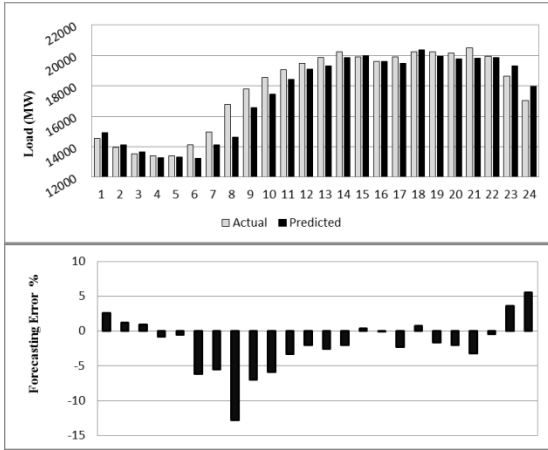
(d) Thursday



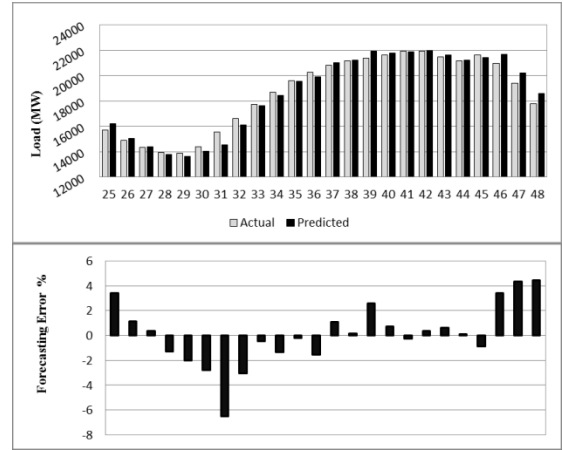
(e) Friday

Figure 84: Week ahead forecast results for CNN

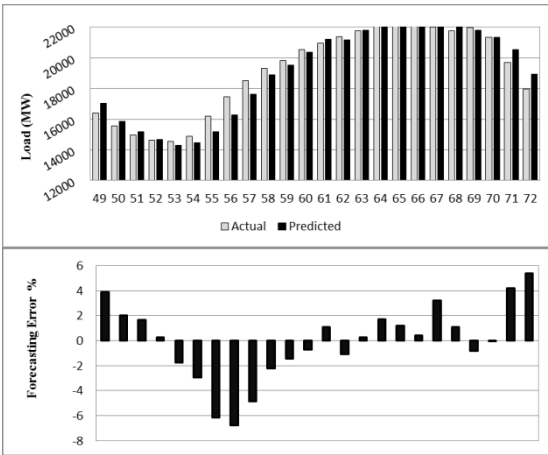
The inaccuracies observed for Monday and Friday are suppressed and an accurate predicted curve is produced by CNN. The MAPE is improved to 2.43% though the error range is -10.97% to 8.7% which is similar to FFNN. Better performance is due to lesser number of error spikes during the forecast. Only 12 times out of 120 the error goes rises more than 5%.



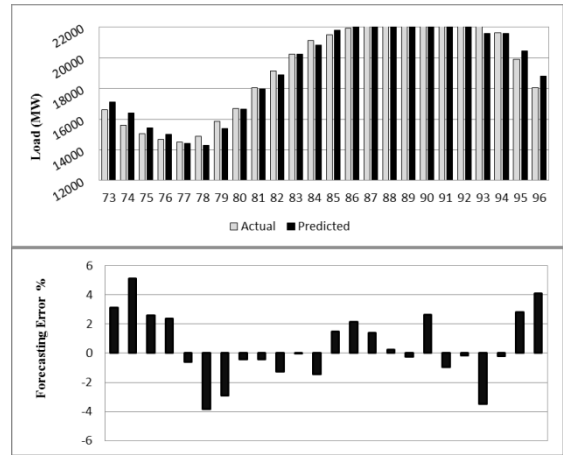
(a) Monday



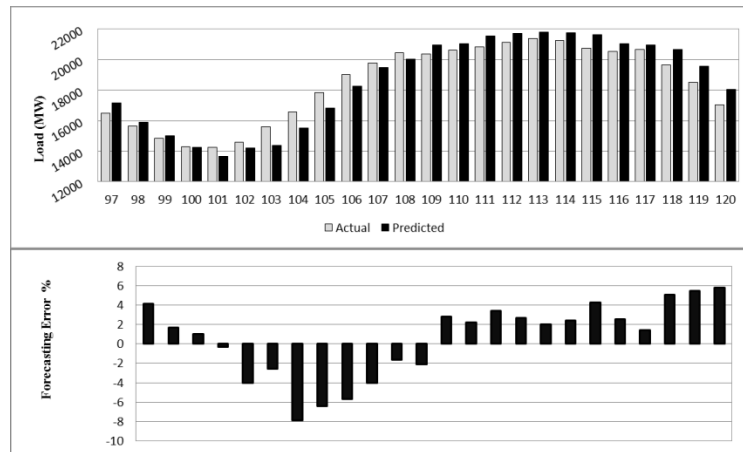
(b) Tuesday



(c) Wednesday



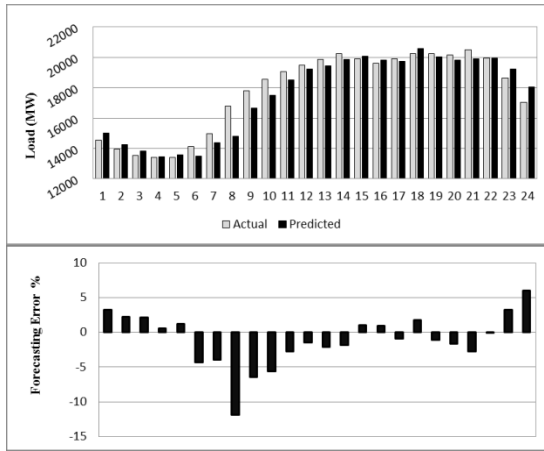
(d) Thursday



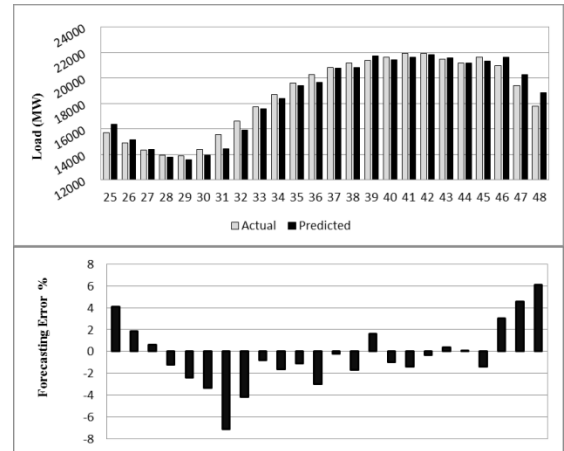
(e) Friday

Figure 85: Week ahead forecast results for RNNL

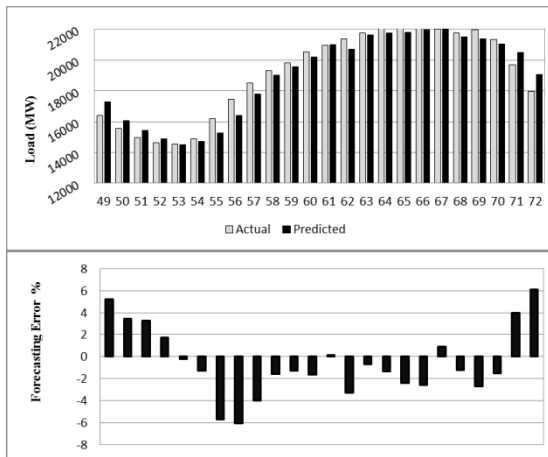
With a MAPE of 2.47% RNNL generates a good forecasting result. The first half of all the days experience negative error but during peak hours the performance improves. Error ranges from -12.81% to 5.79%. The performance can be considered good because of better results during peak hours.



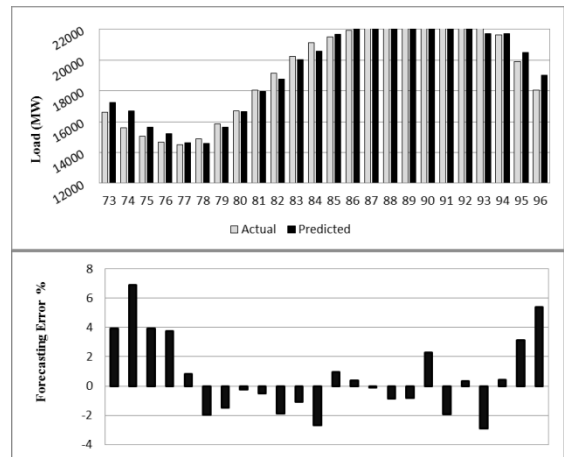
(a) Monday



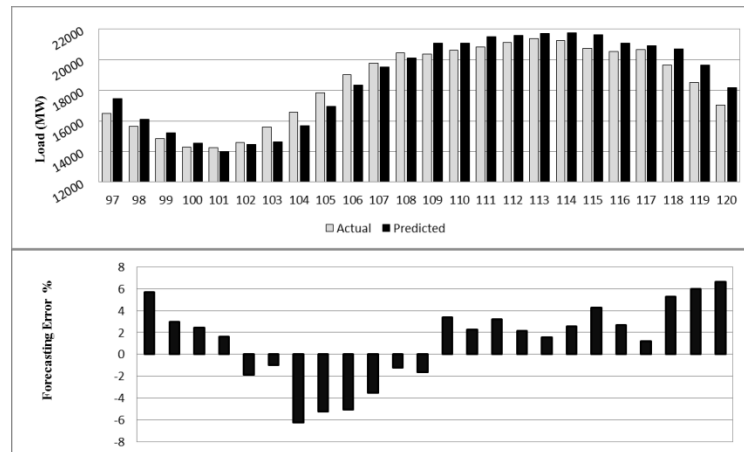
(b) Tuesday



(c) Wednesday



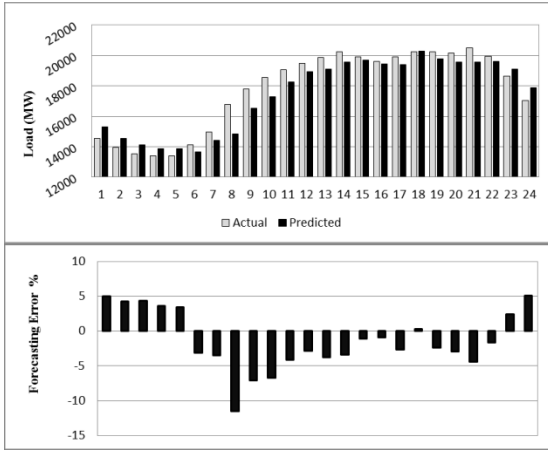
(d) Thursday



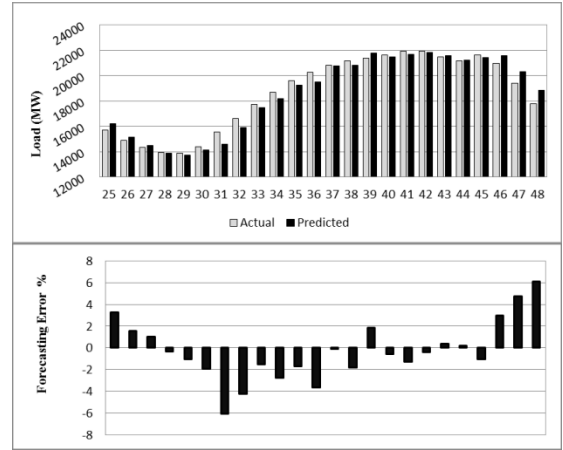
(e) Friday

Figure 86: Week ahead forecast results for RNNG

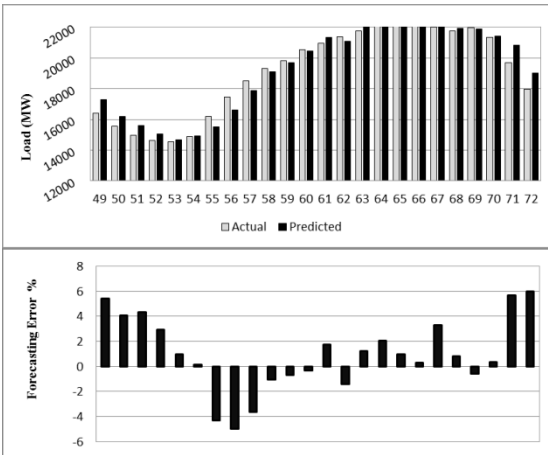
With error spikes occurring regularly RNNG produces a forecast much alike FFNN. Its MAPE is 2.61% and except Tuesday, the performance during peak hours is erroneous. Error varies from -11.92% to 6.9%. Though RNNG forecasts the load slightly better than FFNN but cannot be preferred over it.



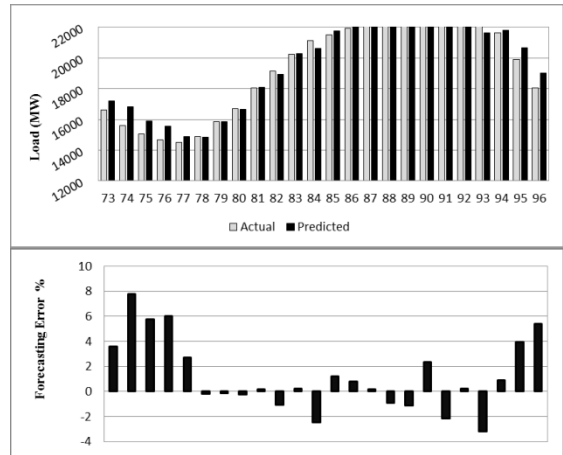
(a) Monday



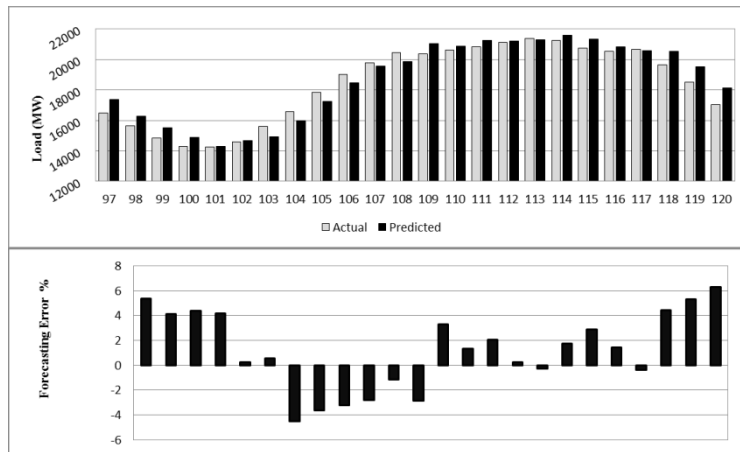
(b) Tuesday



(c) Wednesday



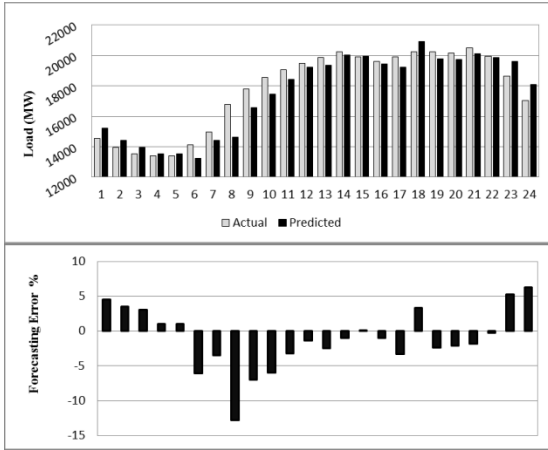
(d) Thursday



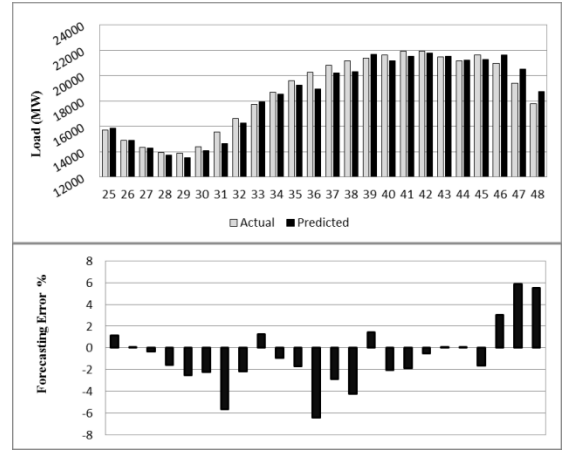
(e) Friday

Figure 87: Week ahead forecast results for CRNNL

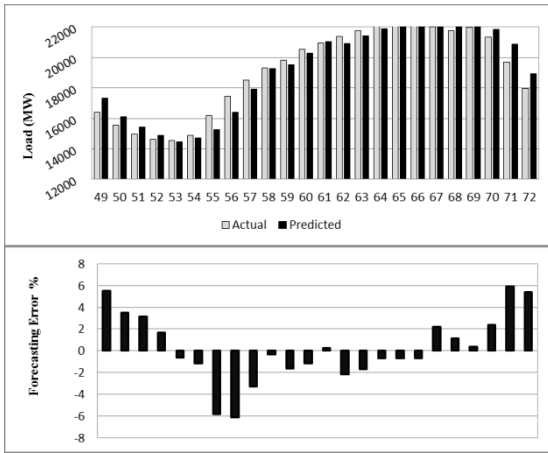
The forecast generated by CRNNL is equivalent to FFNN. With MAPE of 2.64% it does not offer much improvement. Error ranges from -11.5% to 7.78%. Except first day the forecasting results during peak hours are quite accurate. The performance suffers due to rise in error during starting and ending hours of each day.



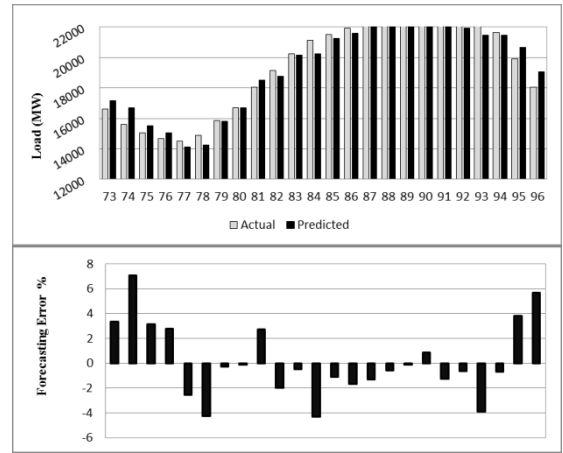
(a) Monday



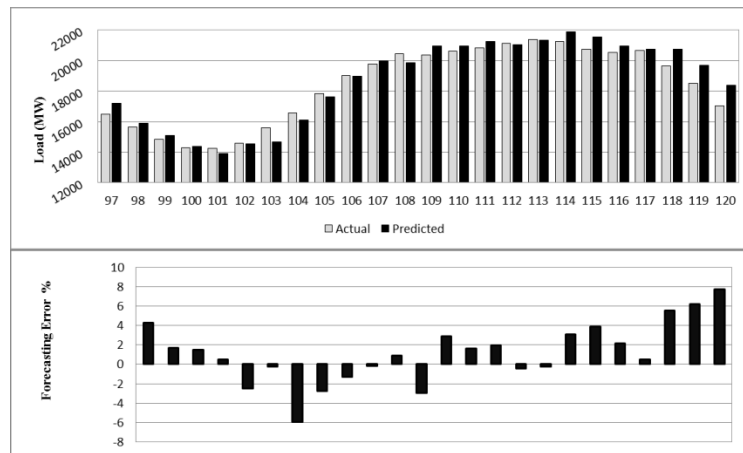
(b) Tuesday



(c) Wednesday



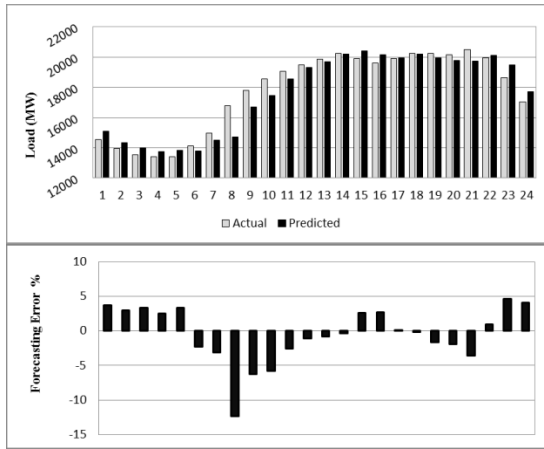
(d) Thursday



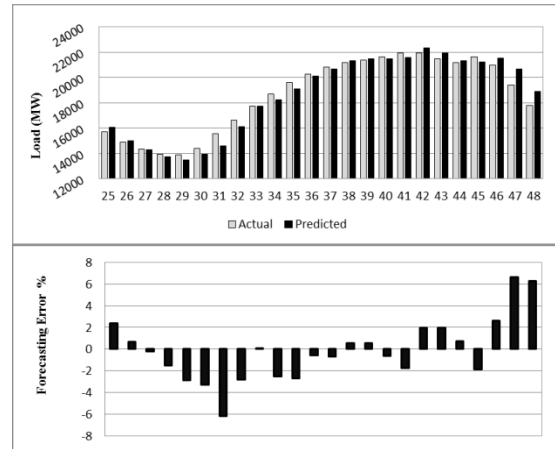
(e) Friday

Figure 88: Week ahead forecast results for CRNG

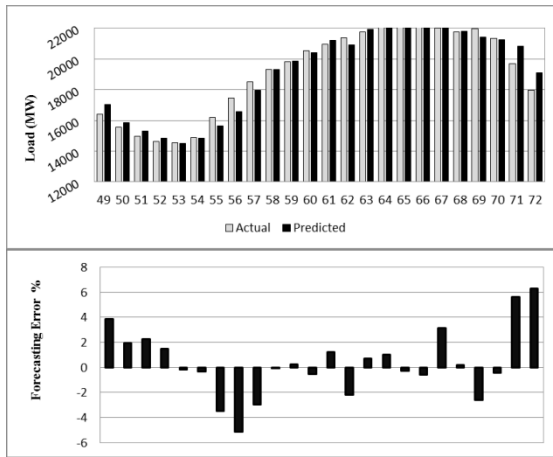
The load forecast for CRNNG is presented above. The error curve is an improvement over RNNG and CRNNL as it suppresses the error at starting and ending hours of days to a small extent to produce MAPE of 2.59% and error peaks -12.75% and 7.75% on the negative and positive axes of the error distribution.



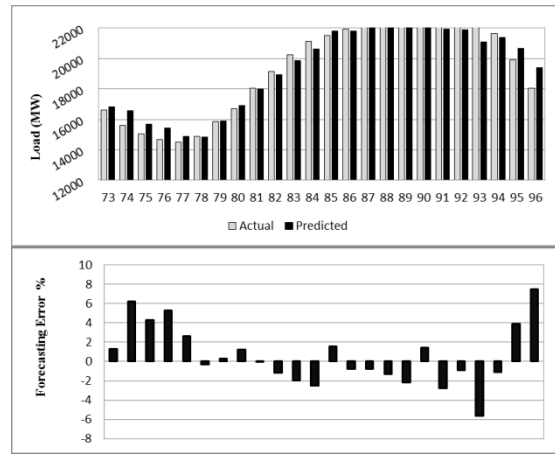
(a) Monday



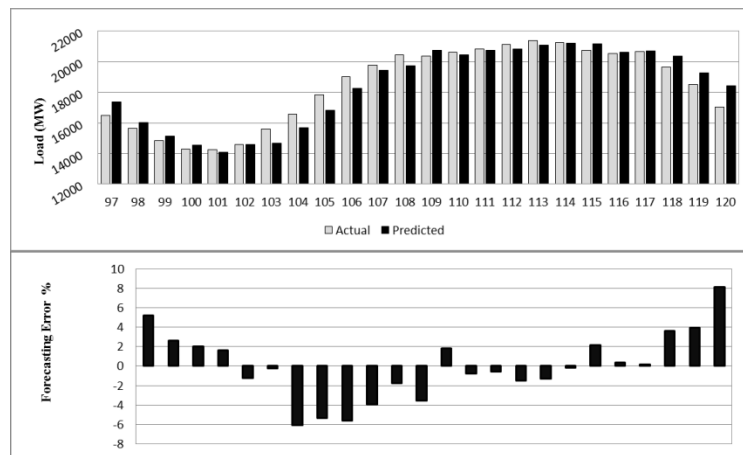
(b) Tuesday



(c) Wednesday



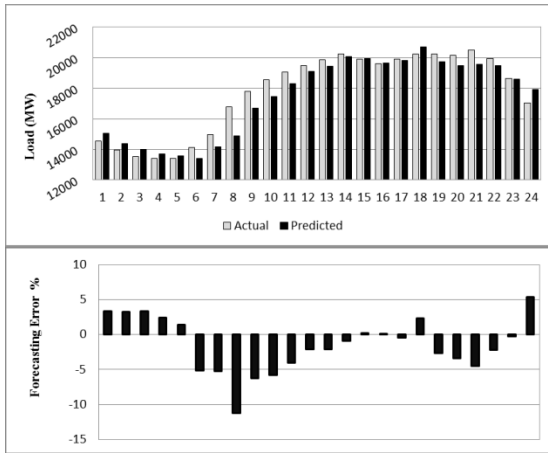
(d) Thursday



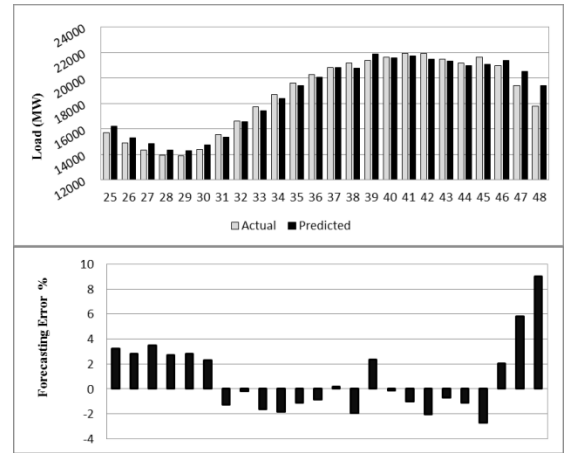
(e) Friday

Figure 89: Week ahead forecast results for 3-layered FFNN

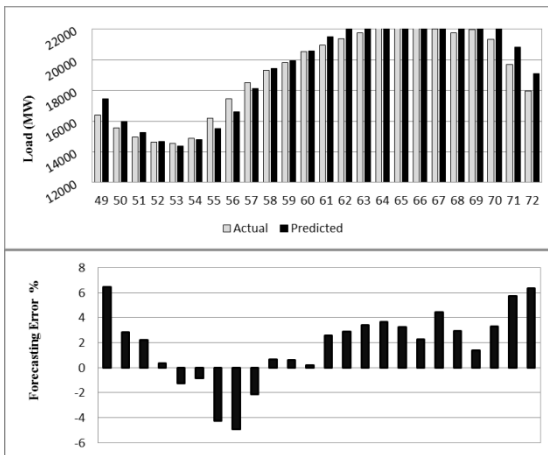
The error range of 3-layered FFNN is -12.33% to 8.14% which is not a reduced one but overall performance is improved over most of the networks discussed. Error performance throughout is uniform with minimal occurrence of abnormal error spikes. This has helped to improve the MAPE to 2.43%.



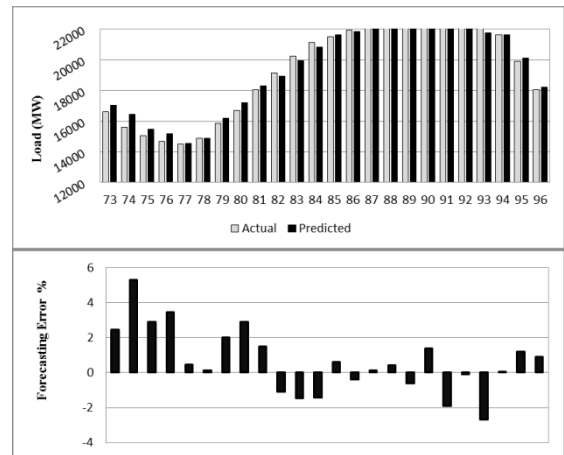
(a) Monday



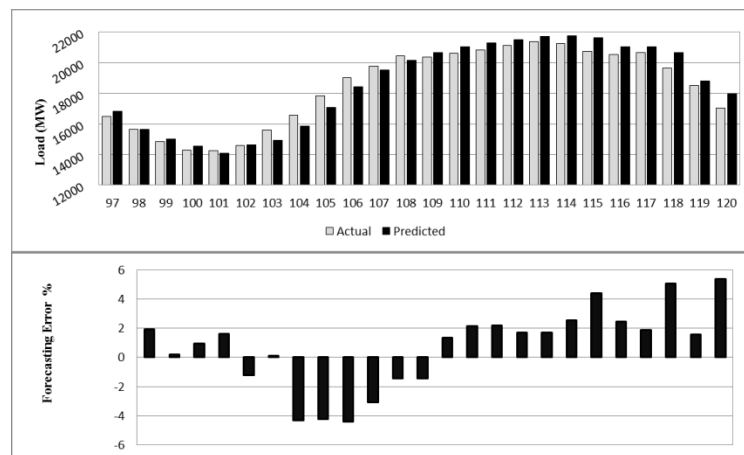
(b) Tuesday



(c) Wednesday



(d) Thursday



(e) Friday

Figure 90: Week ahead forecast results for 3-layered CNN

With a similar error range as 3-layered FFNN, the 3-layered CNN's forecasting results are demonstrated in the figure above. Error varies between -11.22% and 9.01% but the overall negative error is reduced as compared to the other networks. The error curve starts negative but remains towards the positive side especially during peak hours.

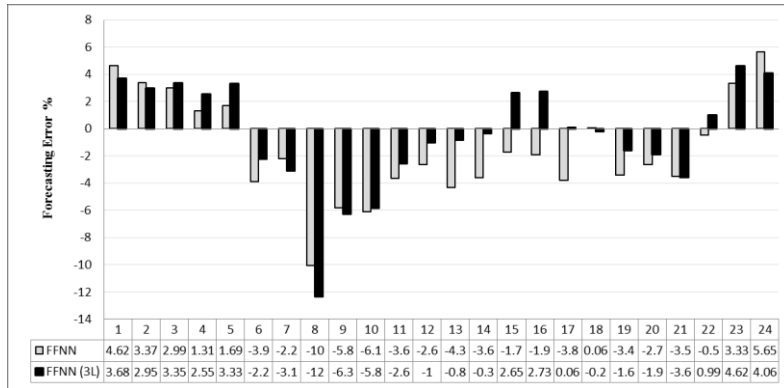
5.3.4 Week ahead forecast for Weekdays

The week ahead ON load forecast for weekdays using all the networks is summarized in the table below. The performance of FFNN has proved least accurate so an improvement of all the other networks over it is given.

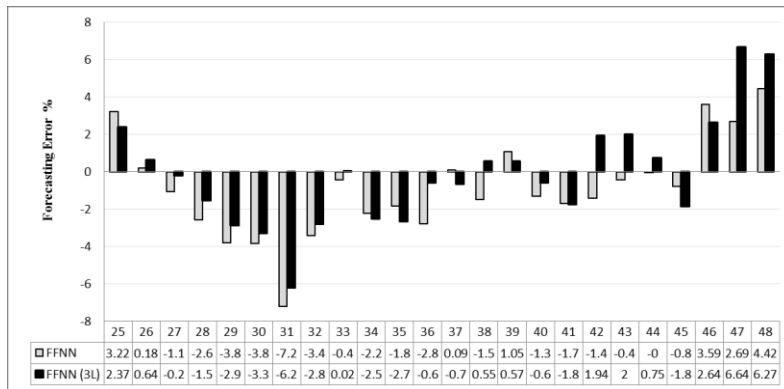
Table 12: Week ahead MAPE and RMSE for Ontario

Architecture	MAPE	RMSE	Improvement
<u>FFNN</u>	<u>2.65</u>	<u>595.59</u>	<u>0.00%</u>
CNN	2.43	561.93	8.20%
RNNL	2.47	571.28	6.68%
RNNG	2.61	579.74	1.72%
CRNNL	2.64	585.84	0.34%
CRNNG	2.59	600.44	2.40%
<u>FFNN (3L)</u>	<u>2.43</u>	<u>571.25</u>	<u>8.25%</u>
CNN (3L)	2.44	558.29	8.09%

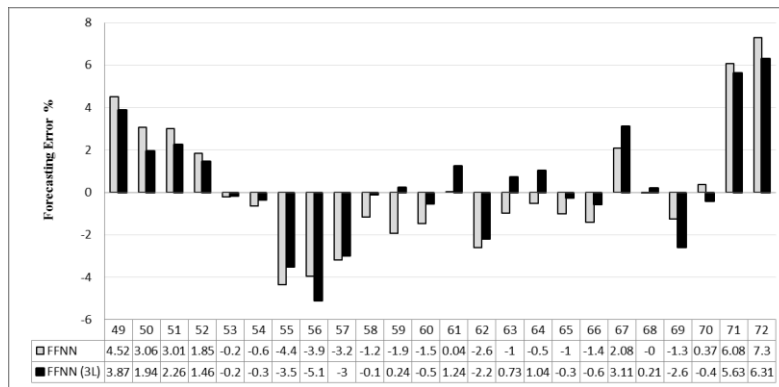
The network RNNG, CRNNL and CRNNG gives minimal improvements of 1.72%, 0.34% and 2.40% over FFNN. RNNL with 6.68% increased accuracy can be preferred over FFNN as the forecast it produces performs well for the peak hours. The CNN, 3-layered FFNN and 3-layered CNN enhance the forecast with 3-layered FFNN performing slightly better than the two. All the three networks are fast in convergence. The 3-layered CNN gives better results during peak hours than the other two but 3-layered FFNN offers suppression of error throughout the week which makes it an ideal choice for a utility company to implement. Following a graph comparing the forecasting error performances of FFNN and 3-layered FFNN is given to highlight the improvement.



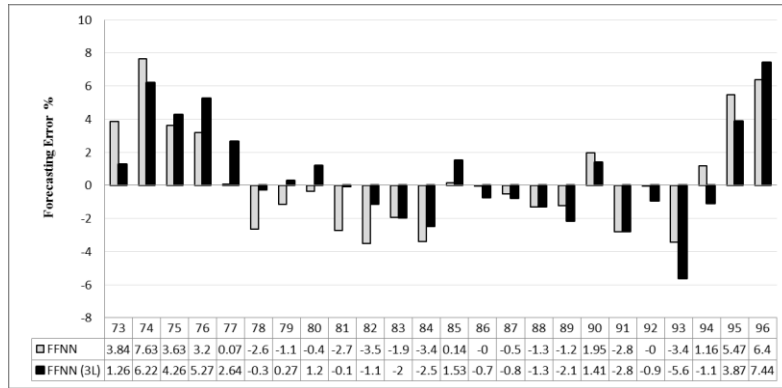
(a) Monday



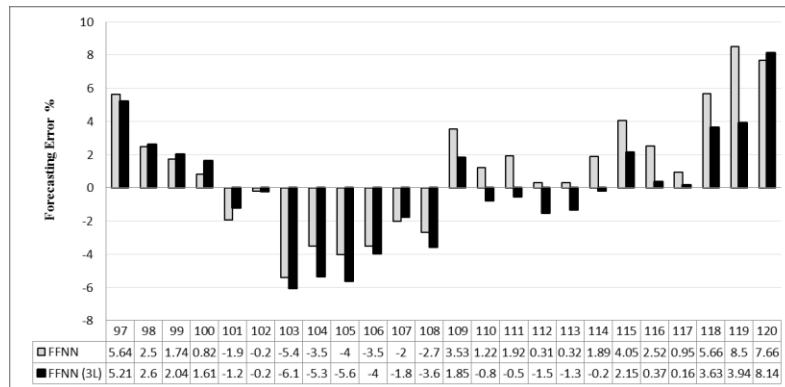
(b) Tuesday



(c) Wednesday



(d) Thursday



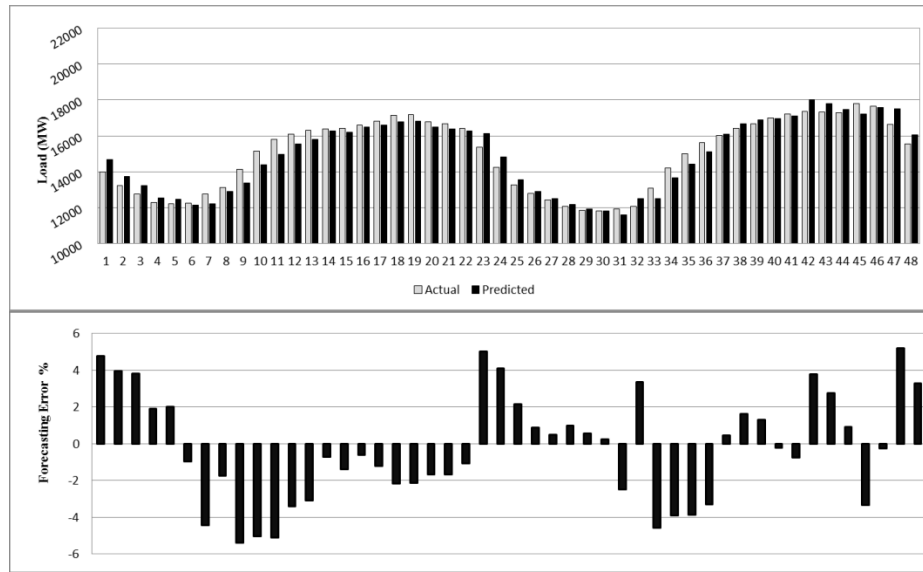
(e) Friday

Figure 91: Week ahead forecasting error performance comparison between FFNN and FFNN (3L) for Ontario

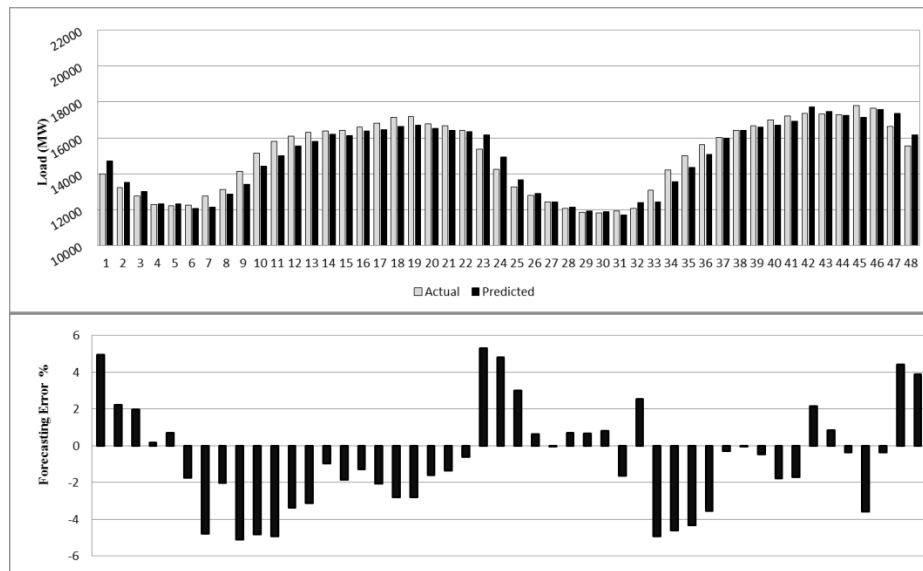
5.3.5 Load forecast for Weekdays

The weekend forecast for Ontario is given. Same weekend is selected to forecast as for other two provinces. The weekend load becomes more predictable for Ontario.

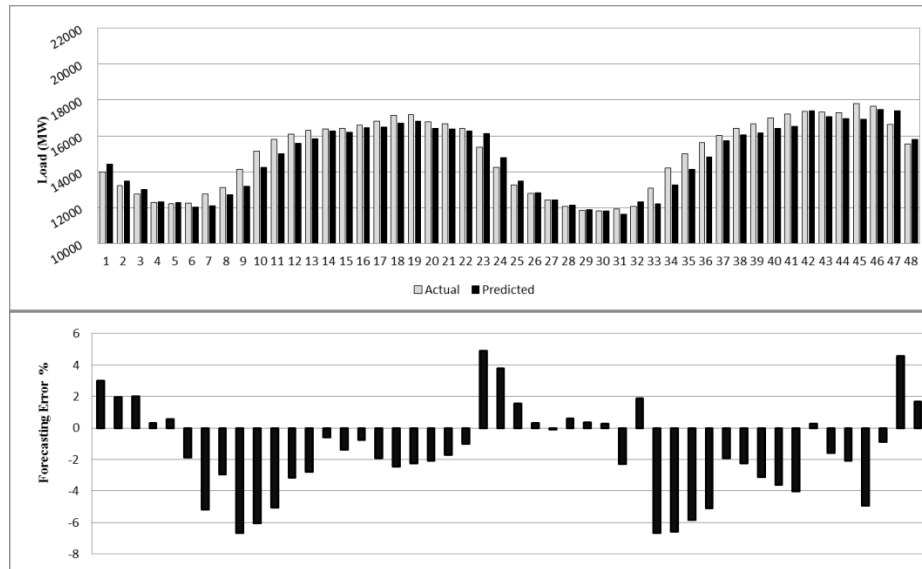
Following figure gives the forecasts for FFNN, CNN, RNNL and RNNG networks.



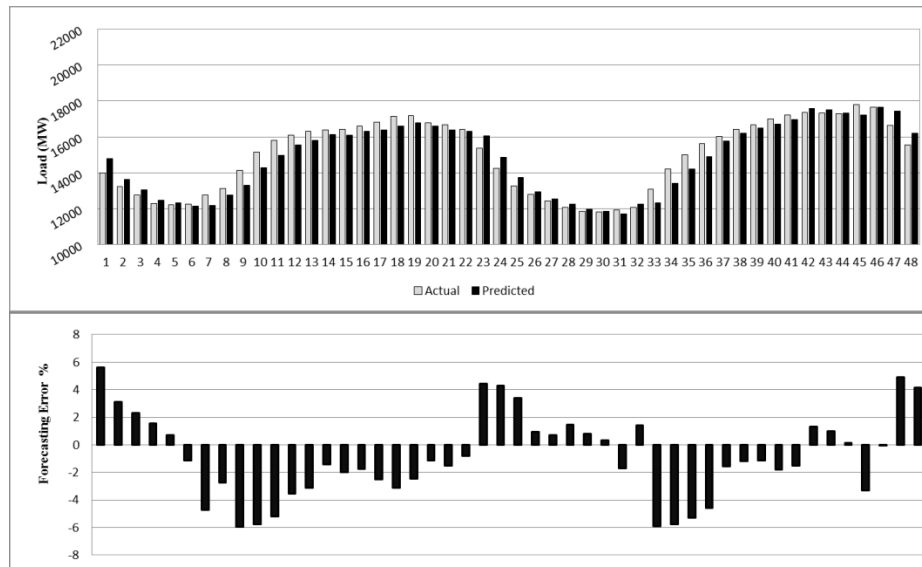
(a) FFNN



(b) CNN



(c) RNNL

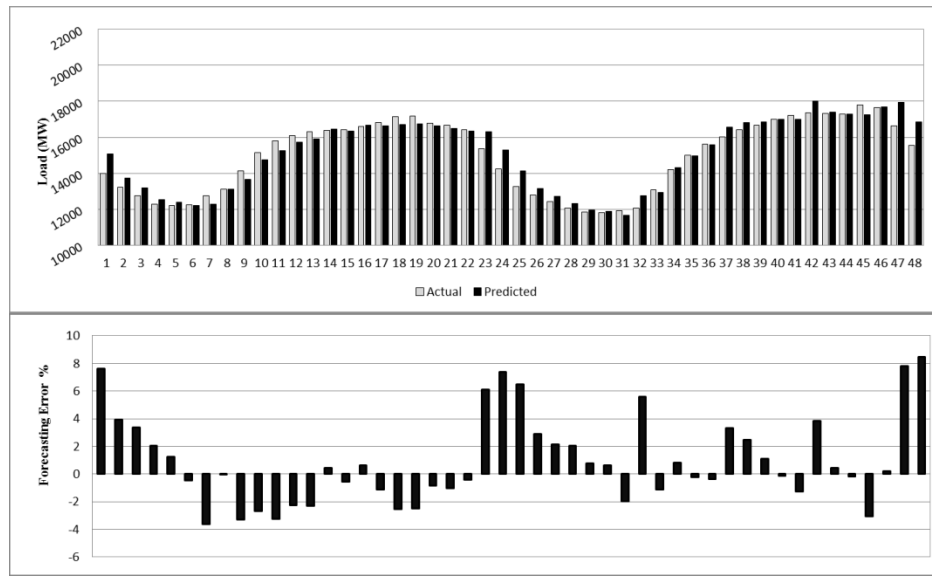


(d) RNNG

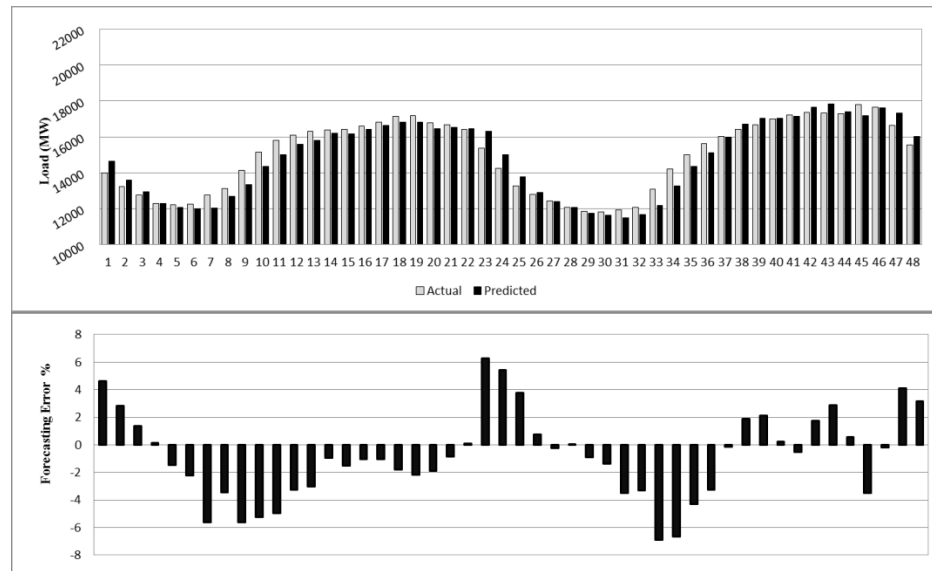
Figure 92: Weekend forecast results (FFNN, CNN, RNNL and RNNG)

FFNN and CNN give similar performance. Most of the times the prediction curve remain below the demand curve. The MAPE for FFNN is 2.46% and for CNN is 2.34%. CNN produces an improved performance because it contains more points with error near to zero than FFNN. The error range for FFNN is -5.39% to 5.21% and that of CNN is -5.1% to 5.31%. The recurrent networks RNNL and RNNG also produce similar

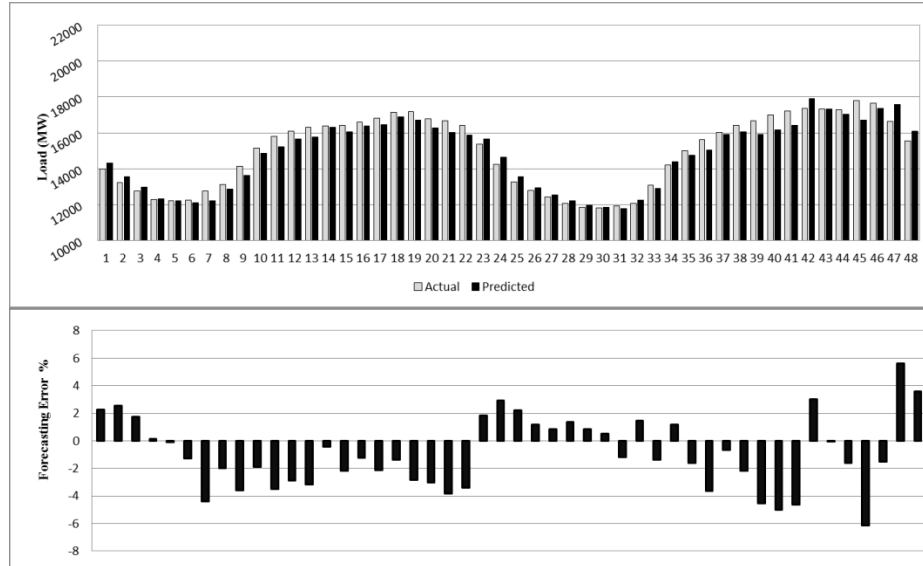
performance to each other. Both have high MAPEs of 2.64% (RNNL) and 2.6% (RNNG). Due to the presence of many error spikes in the error curve both have inferior performance. The error range for RNNL is -6.67% to 4.92% and RNNG is -5.96% to 5.59%.



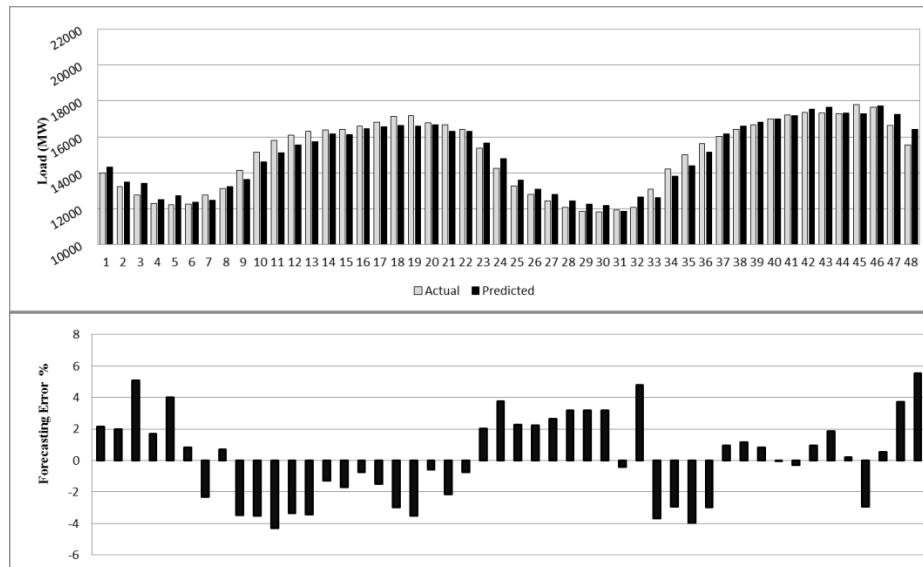
(a) CRNNL



(b) CRNNG



(c) FFNN (3L)



(d) CNN (3L)

Figure 93: Weekend forecast results (CRNNL, CRNNG, FFNN (3L) and CNN (3L))

CRNNL produces a MAPE of 2.43% and mostly produces positive error. This network has greatly reduced the negative error. Its error ranges from -3.64% to 8.45%. CRNNG on the other hand performs similar to RNNG with a MAPE of 2.56%. During first half of both days it generates negative error which reduces during peak hours. Its error ranges from -6.88% to 6.25%. The error curve produced by 3-layered FFNN is very

steady and error remains near to zero on many occasions. The forecast of Sunday is more erroneous than Saturday. Its MAPE is 2.3% with error variation from -6.13% to 5.62%. Finally 3-layered CNN produces a MAPE of 2.33 and performs better on Sunday than 3-layered FFNN. Its error ranges from -4.27% to 5.52%. Because of the occurrence of consistent error peaks during the last half of Saturday its MAPE has gone above 3-layered FFNN.

5.3.6 Comparative Analysis of Load forecast for Weekdays

For weekend forecast of Ontario RNNL gives the poorest performance. The table summarizing and comparing all the results is as follows.

Table 13: Weekend forecast MAPE and RMSE for Ontario

Architecture	MAPE	RMSE	Improvement
FFNN	2.46	437.33	6.94%
CNN	2.34	430.73	11.29%
<u>RNNL</u>	<u>2.64</u>	<u>491.73</u>	<u>0.00%</u>
RNNG	2.60	469.11	1.31%
CRNNL	2.43	489.99	7.97%
CRNNG	2.56	468.53	3.09%
<u>FFNN (3L)</u>	<u>2.30</u>	<u>437.50</u>	<u>12.67%</u>
CNN (3L)	2.33	396.62	11.66%

All the networks perform better to forecast weekend load as compared to weekdays. The 3-layered FFNN stands out of all the other networks with improvement of 12.67%. Other significant improvements are 3-layered CNN (11.66%) and CNN (11.29%). CRNNL (7.97%), FFNN (6.94%), CRNNG (3.09%) and RNNG (1.31%) also produce enhanced results. The forecasting error performance between RNNL and 3-layered FFNN is compared in the following graph:

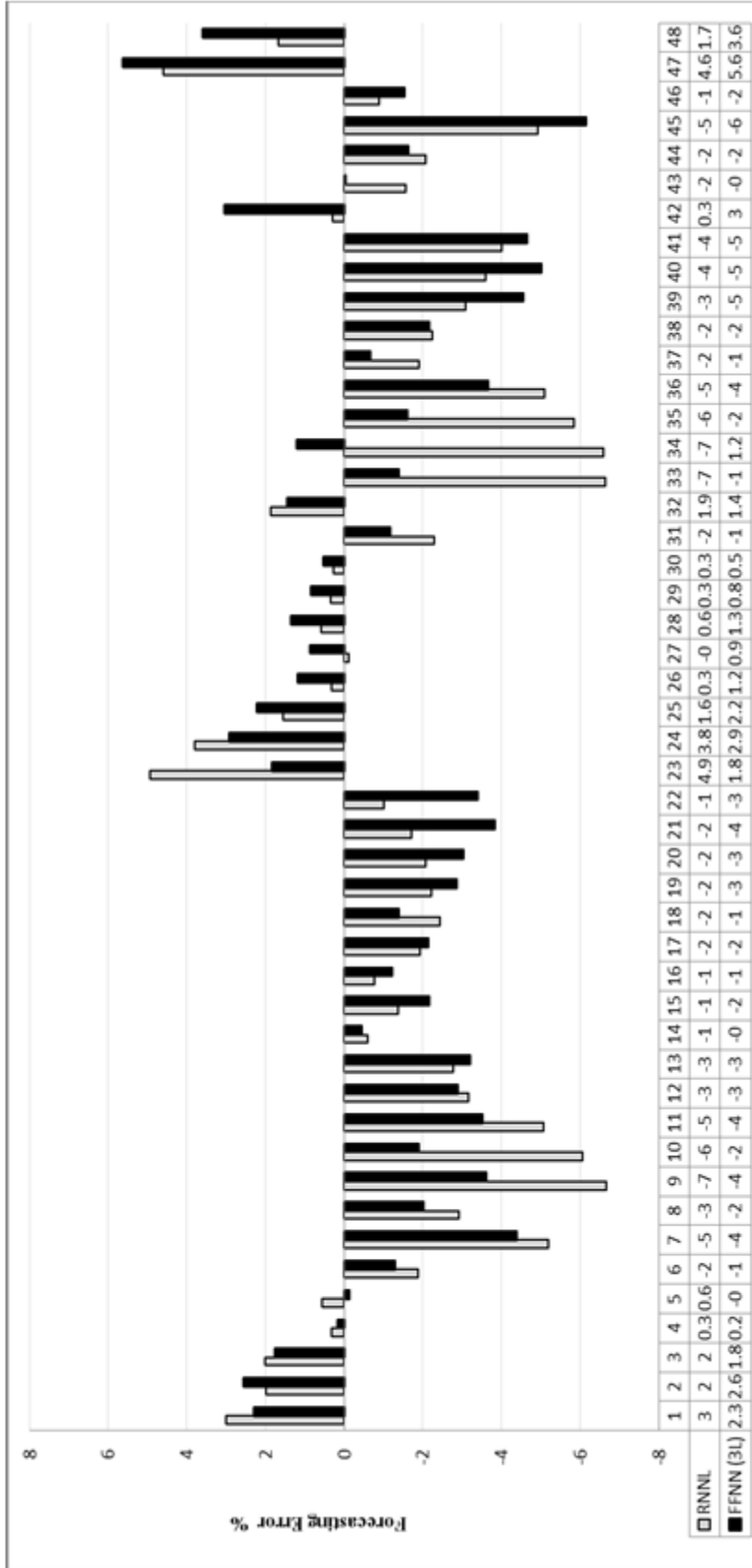


Figure 98: Weekend forecasting error performance comparison between RRNL and FFNN (3L) for Ontario

5.3.7 Summary of Load forecast results for Ontario

A table summarizing all the forecasting results achieved for Ontario is given as below followed by a graphical presentation.

Table 14: Summary of Load forecast results for Ontario

Architecture	Day ahead	Week ahead	Weekend
FFNN	2.86	2.65	2.46
CNN	2.70	2.43	2.35
RNNL	2.75	2.47	2.64
RNNG	2.92	2.61	2.61
CRNNL	2.77	2.64	2.43
CRNNG	3.08	2.59	2.56
FFNN (3L)	2.48	2.43	2.31
CNN (3L)	2.48	2.44	2.34

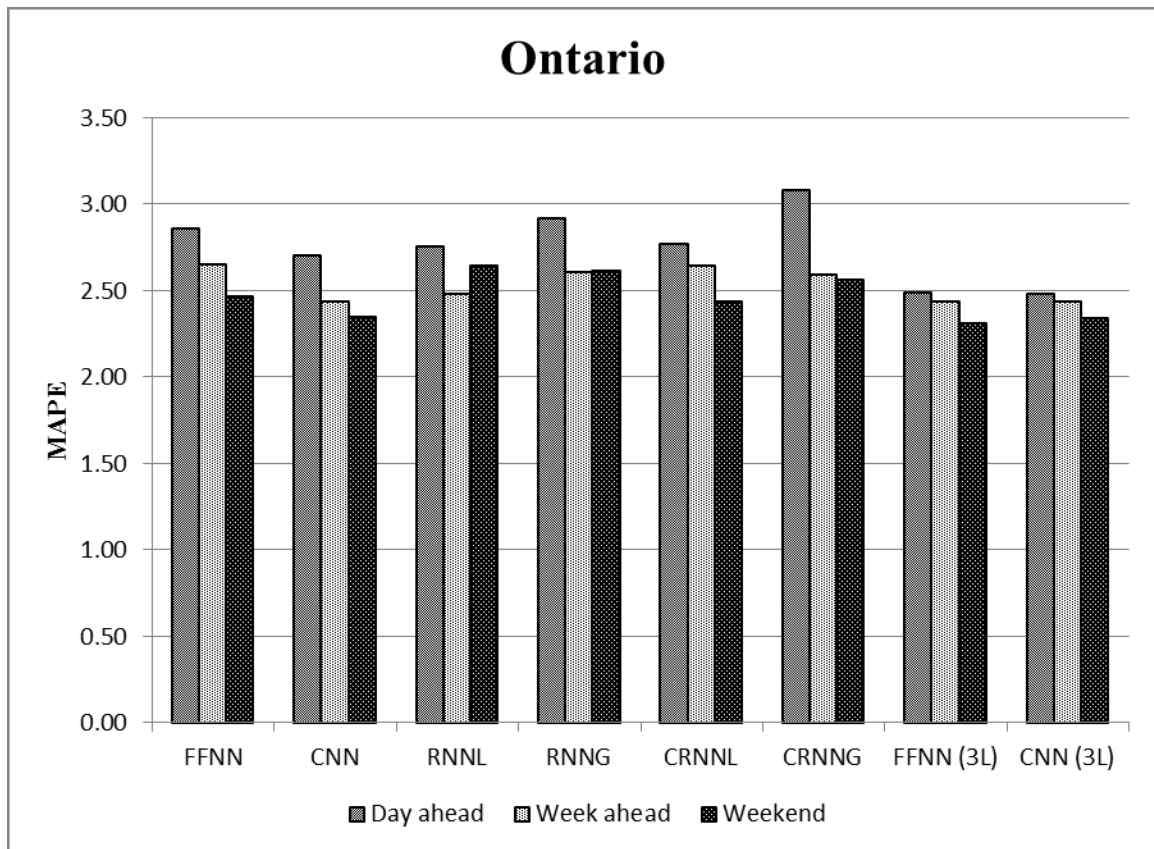


Figure 94: Graphical presentation of load forecast results for Ontario

5.4 Summary of Results

The yearly peak load of Maritime Provinces is approximately 2000 MW. For the time period June 1 to August 30, 2014 the maximum load for Nova Scotia was 1373 MW and that of New Brunswick was 1450 MW. On the other hand Ontario peaks at 24927 MW which occurs during the mentioned period. The big difference between load demand of Ontario and the Maritime Provinces is evident. Previous sections elaborate the performance and comparison between each individual network; this section summarizes all the forecasting results achieved and presents a comparative analysis of forecasting error performance for all three provinces.

5.4.1 24 hours ahead forecast for Weekdays

Following table gives a summary of error performances of 24 hours ahead forecast for the three understudy provinces. The best and worst performing networks for each jurisdiction are highlighted and an improvement of all networks over worst performing one is also demonstrated:

Table 15: Summary of 24 hours ahead forecast for Weekdays

Architecture	Nova Scotia		New Brunswick		Ontario	
	MAPE	% Imp	MAPE	% Imp	MAPE	% Imp
FFNN	2.32	0.00%	2.30	0.00%	2.86	7.31%
CNN	1.93	16.68%	1.96	14.98%	2.70	12.33%
RNNL	2.26	2.80%	2.14	7.19%	2.75	10.75%
RNNG	2.18	5.86%	2.16	6.15%	2.92	5.31%
CRNNL	2.21	4.76%	2.14	6.89%	2.77	10.09%
CRNNG	2.18	5.86%	2.21	3.77%	3.08	0.00%
FFNN (3L)	1.99	14.42%	1.93	15.99%	2.48	19.40%
CNN (3L)	1.81	21.88%	1.89	17.71%	2.48	19.52%

From the above table it can be observed the 3-layered CNN outperforms other networks for each jurisdiction. The performance of 3-layered FFNN and CNN are also

noticeable. It is seen that all the networks produce accurate forecasts for Maritime Provinces but for a big industrialized zone like Ontario the forecasting error increases. Also for a big utility, the hybrid and recurrent networks have not proved to be effective as CRNNG gave the most inaccurate results. For Nova Scotia and New Brunswick all the networks offered improvement over FFNN. It was observed that during peak hours when the demand curve stabilizes, the forecasts produced by both hybrid and recurrent networks were accurate and error curve went near to zero. But overall 3-layered CNN was found to produce the most precise forecasts throughout the whole period. Following graph demonstrates the comparison of all the networks for the three selected jurisdictions for day ahead forecasting performances.

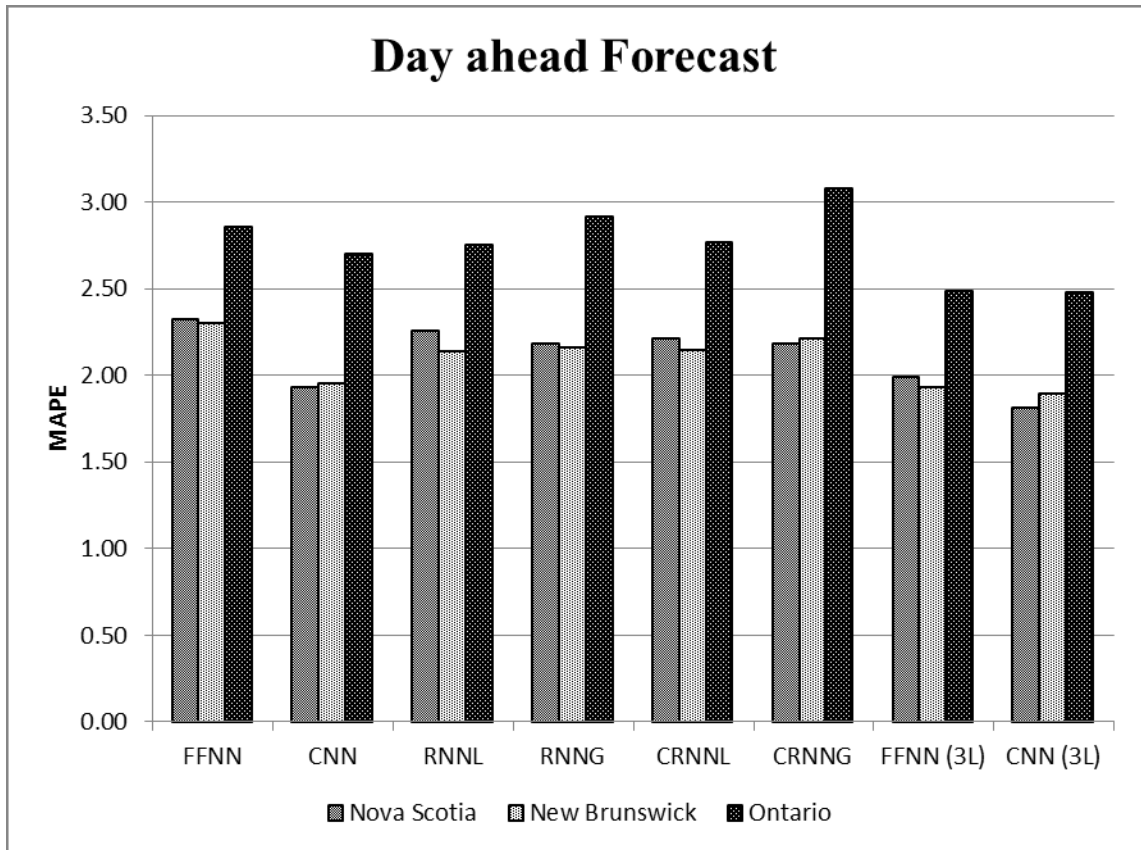


Figure 95: Graphical summary of Day ahead forecast results

5.4.2 Week ahead forecast for Weekdays

Following table gives a summary of error performances of week ahead forecasts for three selected provinces. The best and worst performing networks for each jurisdiction are highlighted and an improvement of all networks over worst performing one is also demonstrated in the table given as follows:

Table 16: Summary of Week ahead forecast for Weekdays

Architecture	Nova Scotia		New Brunswick		Ontario	
	MAPE	% Imp	MAPE	% Imp	MAPE	% Imp
FFNN	2.21	0.00%	2.31	0.00%	2.65	0.00%
CNN	2.03	8.26%	2.12	8.18%	2.43	8.20%
RNNL	2.16	2.32%	2.20	4.48%	2.47	6.68%
RNNG	2.08	5.83%	2.27	1.76%	2.61	1.72%
CRNNL	2.10	4.81%	2.24	2.96%	2.64	0.34%
CRNNG	2.12	4.15%	2.26	2.12%	2.59	2.40%
FFNN (3L)	2.17	1.76%	2.21	4.29%	2.43	8.25%
CNN (3L)	1.99	10.04%	2.18	5.67%	2.44	8.09%

Just like day ahead load forecast, the 3-layered CNN has proven to be most accurate among all the networks for week ahead forecast for NS and ON, whereas for NB CNN has proven to be the best.. The decrease in performance for each individual network is illustrated when a bigger power system like Ontario is forecasted. Hence it can be concurred that a small percent of variation in Ontario’s load demand weighs more and affects the output to a greater extent as compared to a similar percentage of variation in load demands of NS or NB. 3-layered CNN has given better results than other networks after CNN for NB and for NS and ON, CNN has given improved performances as compared to the other networks which means that a cascaded connection for all three provinces has contributed to increment in overall accuracy of forecast. Furthermore, the hybrid networks offer noticeable improvements for Maritimes provinces but for a large sized demand horizon they have failed to impress. Furthermore, the overall week ahead

forecast performances for Ontario have improved as compared to 24 hour ahead forecasts. For Nova Scotia all the recurrent networks; RNNL, RNNG, CRNNL and CRNNG have performed better than day ahead as well. This is not the case for New Brunswick where all the networks have given a bit inferior results than its day ahead forecast. Following graph shows a comparison between the error performances of all the networks for Nova Scotia, New Brunswick and Ontario.

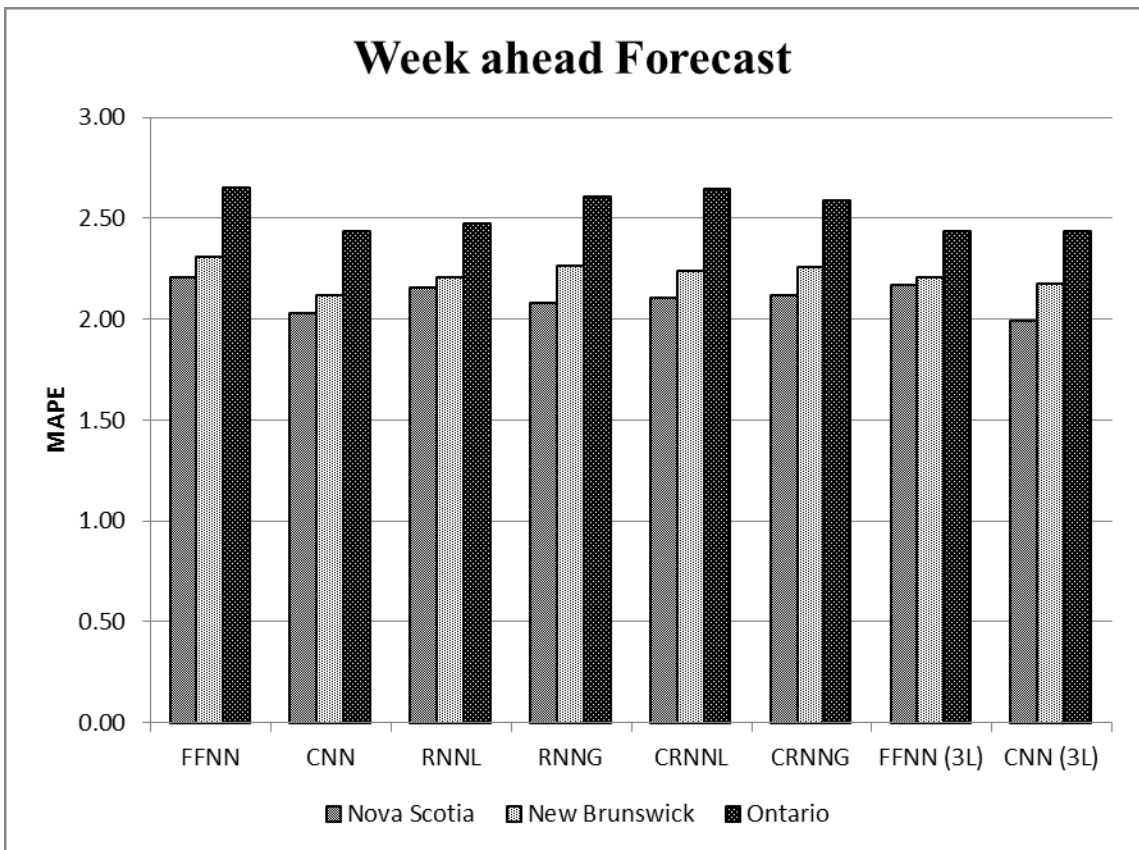


Figure 96: Graphical summary of Week ahead forecast results

5.4.3 Weekend forecast

The error performances of each network for three provinces are summarized in the following table followed by a discussion. Table underlines the best and worst performing networks and an improvement of all networks over worst performing one is also demonstrated:

Table 17: Summary of Weekend forecast

Architecture	Nova Scotia		New Brunswick		Ontario	
	MAPE	% Imp	MAPE	% Imp	MAPE	% Imp
FFNN	2.17	3.46%	2.59	1.30%	2.46	6.94%
CNN	1.85	17.65%	2.49	5.38%	2.35	11.29%
RNNL	2.10	6.54%	2.56	2.72%	<u>2.64</u>	<u>0.00%</u>
RNNG	<u>2.24</u>	<u>0.00%</u>	<u>2.63</u>	<u>0.00%</u>	2.61	1.31%
CRNNL	2.09	7.07%	2.38	9.52%	2.43	7.97%
CRNNG	1.76	21.53%	2.56	2.52%	2.56	3.09%
FFNN (3L)	<u>1.71</u>	<u>23.93%</u>	<u>2.38</u>	<u>9.42%</u>	<u>2.31</u>	<u>12.67%</u>
CNN (3L)	1.94	13.68%	2.39	9.16%	2.34	11.66%

Weekends have different load patterns than weekdays and are generally more predictable. The best performing network for all three jurisdictions for this case have turned out to be 3-layered FFNN. For Nova Scotia and Ontario weekend forecasts all networks except RNNL and CRNNL have produced better results than New Brunswick forecasts. The main reason for this is an unexpected peak in load demand of New Brunswick that occurred during Saturday. If we compare the performances of all the networks from weekday performances, then except for New Brunswick the forecasting error has been greatly reduced. For Nova Scotia only RNNG and for Ontario RNNL have performed below weekdays forecasts of the same networks. For New Brunswick all networks have performed below par with RNNG to give the worst results. A graphical

presentation of weekend forecast results produced by all the networks for Nova Scotia, New Brunswick and Ontario is shown below.

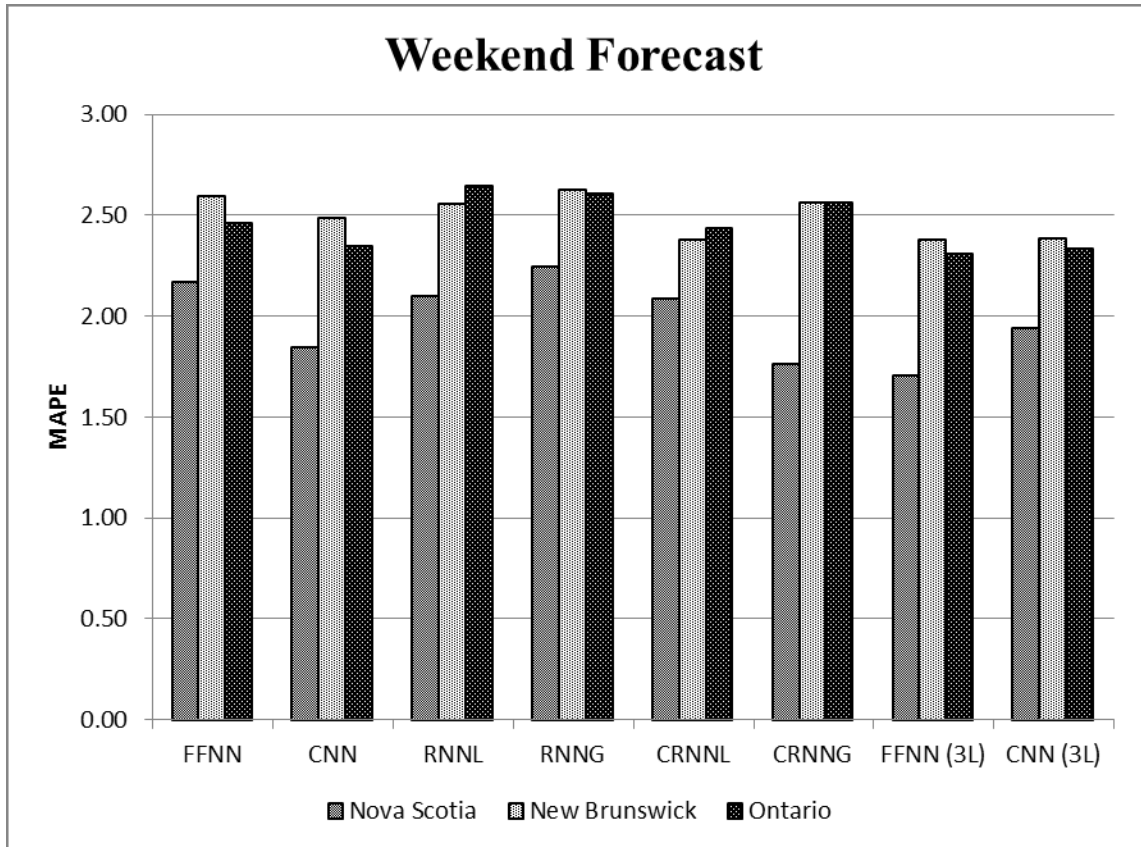


Figure 97: Graphical summary of Weekend forecast results

Chapter 6: Conclusion

6.1 Summary and Contributions

Various techniques are used to perform STLF. With the advent of deregulated market environment classical forecasting methods are being replaced by intelligent methods to understand variations in load. After going through the literature available on ANN to perform STLF, it was realized that some advanced neural network architectures have not been studied comprehensively and there is room for utilizing the properties of these modern architectures and introduce new ones. This thesis evaluates the performances of eight ANN architectures. In order to explore the applicability of these designs real data from three Canadian provinces; Nova Scotia (NS), New Brunswick (NB) and Ontario (ON) was used to experiment each individual network.

The ANN designs examined in this work start from the most commonly used feed forward neural network (FFNN). FFNN served as a base to compare the performances of remaining ANNs. Three advanced architectures tested were cascaded neural network (CNN), recurrent neural network with local feedback (RNNL) and recurrent neural network with global feedback (RNNG). Also, four new architectures for the application of STLF were also introduced. These include two hybrid and two modified architectures. The hybrid models proposed were the combination of CNN and the two recurrent networks first was the cascaded – recurrent neural network with local feedback (CRNNL) and the other one was cascaded – recurrent neural network with global feedback (CRNNG). Modified architectures consisted of 3-layered FFNN and 3-layered CNN.

A statistical analysis of load demand of each province indicated that weekends possessed different load response than weekdays, hence weekdays and weekends were

forecasted separately. The performances were evaluated on the basis of mean absolute percentage error (MAPE). Each network produced three forecasts for each province. Two out of three forecasts were day or 24 hours ahead forecast and week ahead forecast for weekdays and the third one forecasted weekend load. After achieving forecasting results for all the cases a comparative analysis was made.

Comparisons were primarily based on three different forecasts for each province. For day ahead weekdays forecasts 3-layered CNN produced the most accurate results as compared to other networks with 1.81%, 1.89% and 2.48% MAPE for NS, NB and ON respectively. For week ahead weekdays forecasts 3-layered CNN presented accurate forecasts for NS and ON with MAPE of 1.99% (NS) and 2.44% (ON), whereas CNN proved best for NB with MAPE 2.12%. Finally for weekend forecast 3-layered FFNN with MAPE 1.71%, 2.38% and 2.31% for NS, NB and ON outperforms all networks.

Hybrid and recurrent networks performed better in peak hours but didn't produce satisfactory results. RNNL and CRNNL for day ahead forecast of Ontario enhanced results by improvements of 10.75% and 10.09% over worst performing CRNNG network. Other noticeable results include week ahead forecast by RNNG (5.83% improvement) for NS and CRNNL (9.52% improvement) for NB and weekend forecast by CRNNG (21.53% improvement) for NS and RNNL (4.48% improvement) for NB.

Furthermore, CNN and 3-layered architectures offered big improvements. With fast convergence they produced superior results than any of recurrent networks for most cases stated. Finally, the economic worth of forecasting error depends upon the operation zone of the system (Unit commitment).

6.2 Suggestions for Future Work

Perhaps this thesis presents a detailed performance evaluation of advanced and new neural network architectures and it was seen that architectures especially with cascaded connections were found to perform better than hybrid or recurrent networks. Suggestion for future work involves replacement of Levenberg-Marquardt Back propagation learning algorithm with Kalman filter or Support Vector machine. Both these techniques may prove to improve both accuracy and speed. Moreover, wavelet technique to transform inputs to filter any abnormality or redundancy might also aid in accuracy.

References

- [1] E. Kyriakides and M. Polycarpou, "Short Term Electric Load Forecasting: A Tutorial", *Studies in Computational Intelligence*, vol. 35, 391–418, Springer-Verlag Berlin Heidelberg 2007.
- [2] "Appendix B: Electric Power Systems Basics", *MIT study on The Future of the Electric Grid*, 243-260, Massachusetts Institute of Technology 2011. Available: <http://mitei.mit.edu/publications/reports-studies/future-electric-grid> [Accessed: Dec. 25 2013]
- [3] Steven W. Blume, "Electric Power Systems Basics". *IEEE Press Series on Power Engineering*, Series editor: Mohamed E. El-Hawary, A John Wiley and Sons, Inc., Publications 2007.
- [4] Hodge, B.-M.; Lew, D.; Milligan, M., "Short-Term Load Forecast Error Distributions and Implications for Renewable Integration Studies," *Green Technologies Conference, 2013 IEEE* , vol., no., pp.435,442, 4-5 April 2013.
- [5] Turkay, B.E., and Demren, D., "Electrical load forecasting using support vector machines," *Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on*, pp. I-49 – I-53, 1-4 Dec. 2011
- [6] Hobbs, B.F., Jitprapaikulsarn, S., Konda, S.; Chankong, V., Loparo, K.A., and Maratukulam, D.J., "Analysis of the value for unit commitment of improved load forecasts," *Power Systems, IEEE Transactions on*, vol.14, no.4, pp.1342-1348, Nov 1999
- [7] Hippert, H.S.; Pedreira, C.E.; Souza, R.C., "Neural networks for short-term load forecasting: a review and evaluation," *Power Systems, IEEE Transactions on* , vol.16, no.1, pp.44,55, Feb 2001
- [8] Feinberg, Eugene A., and Dora Genethliou. "Load forecasting" *Applied mathematics for restructured electric power systems*, pp. 269-285, Springer US, 2005.
- [9] Tongxin Zheng, Adly A Girgis, and Elham B Makram, "A hybrid wavelet-Kalman filter method for load forecasting", *Electric Power Systems Research*, vol. 54, Issue 1, pp. 11-17, April 5, 2000.
- [10] Alfares, Hesham K. and Nazeeruddin, Mohammad. "Electric load forecasting: Literature survey and classification of methods" *Int. J. Systems Science* 33 , no. 1, pp. 23-24, 2002
- [11] Moghram, I.S.; Rahman, S., "Analysis and evaluation of five short-term load forecasting techniques," *Power Systems, IEEE Transactions on* , vol.4, no.4, pp.1484,1491, Nov 1989

- [12] Peng, T. M.; Hubele, N.F.; Karady, G.G., "Advancement in the application of neural networks for short-term load forecasting," *Power Systems, IEEE Transactions on* , vol.7, no.1, pp.250,257, Feb 1992
- [13] Jain, A.K.; Jianchang Mao; Mohiuddin, K. M., "Artificial neural networks: a tutorial," *Computer* , vol.29, no.3, pp.31,44, Mar 1996
- [14] Khotanzad, A.; Afkhami-Rohani, R.; Maratukulam, D., "ANNSTLF-Artificial Neural Network Short-Term Load Forecaster generation three," *Power Systems, IEEE Transactions on* , vol.13, no.4, pp.1413,1422, Nov 1998
- [15] Bakirtzis, A.G.; Petridis, V.; Kiartzis, S.J.; Alexiadis, M.C.; Maissis, A.H., "A neural network short term load forecasting model for the Greek power system," *Power Systems, IEEE Transactions on* , vol.11, no.2, pp.858,863, May 1996
- [16] Ortiz-Arroyo, D.; Skov, M.K.; Huynh, Q., "Accurate Electricity Load Forecasting with Artificial Neural Networks," *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on* , vol.1, no., pp.94,99, 28-30 Nov. 2005
- [17] Wenjin Dai; Ping Wang, "Application of Pattern Recognition and Artificial Neural Network to Load Forecasting in Electric Power System," *Natural Computation, 2007. ICNC 2007. Third International Conference on* , vol.1, no., pp.381,385, 24-27 Aug. 2007
- [18] Yong Wang; Dawu Gu; Jianping Xu; Jing Li, "Back Propagation Neural Network for Short-term Electricity Load Forecasting with Weather Features," *Computational Intelligence and Natural Computing, 2009. CINC '09. International Conference on* , vol.1, no., pp.58,61, 6-7 June 2009
- [19] Dongxiao Niu; Hui Shi; Jianqing Li; Yanan Wei, "Research on short-term power load time series forecasting model based on BP neural network," *Advanced Computer Control (ICACC), 2010 2nd International Conference on* , vol.4, no., pp.509,512, 27-29 March 2010
- [20] Tasre, M.B.; Ghate, V.N.; Bedekar, P.P., "Hourly load forecasting using Artificial Neural Network for a small area," *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on* , vol., no., pp.379,385, 30-31 March 2012
- [21] Yongli Wang, Dongxiao Niu, Li Ji, "Short-term power load forecasting based on IVL-BP neural network technology", *Systems Engineering Procedia*, vol. 4, pp. 168-174, 2012, ISSN 2211-3819.
- [22] AlFuhaid, A.S.; El-Sayed, M.A.; Mahmoud, M. S., "Cascaded artificial neural networks for short-term load forecasting," *Power Systems, IEEE Transactions on* , vol.12, no.4, pp.1524,1529, Nov 1997

- [23] Dong-Chul Park; Dong-Min Woo; Seung-Soo Han, "Electric Load Forecasting Using Adaptive Multiresolution-Based Bilinear Recurrent Neural Network," *Image and Signal Processing, 2008. CISP '08. Congress on* , vol.4, no., pp.393,397, 27-30 May 2008.
- [24] Lee, K.Y.; Choi, T.I.; Ku, C.C.; Park, J.H., "Short-term load forecasting using diagonal recurrent neural network," *Neural Networks to Power Systems, 1993. ANNPS '93., Proceedings of the Second International Forum on Applications of* , vol., no., pp.227,232, 1993
- [25] S. H. Shin and D. C. Park, "Short-term load forecasting using bilinear recurrent neural network," in *Proc. 4th Int. Symp. Neural Netw.*, vol. 3. pp. 111–116, 2007.
- [26] Changhao Xia; Zhonghua Yang; Hongjie Li, "Electric load forecasting using virtual instrument based on dynamic recurrent Elman neural network," *Power Engineering and Automation Conference (PEAM), 2012 IEEE* , vol., no., pp.1,4, 18-20 Sept. 2012
- [27] Feng Zhao; Hongsheng Su, "Short-Term Load Forecasting Using Kalman Filter and Elman Neural Network," *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on* , vol., no., pp.1043,1047, 23-25 May 2007
- [28] Soliman, S.A; and Ahmad Al-Kandari, M.; "3 - Load Modeling for Short-Term Forecasting, In Electrical Load Forecasting", edited by Soliman Abdel-Hady Soliman and Ahmad M. Al-Kandari, Butterworth-Heinemann, Boston, pp. 79-97, 2010, ISBN 9780123815439
- [29] Lopez, M.; Valero, S.; Senabre, C.; Aparicio, J.; Gabaldon, A., "Standardization of short-term load forecasting models," *European Energy Market (EEM), 2012 9th International Conference on the* , vol., no., pp.1,7, 2012.
- [30] Abu-El-Magd, Mohamed A.; Sinha, Naresh K., "Short-Term Load Demand Modeling and Forecasting: A Review," *Systems, Man and Cybernetics, IEEE Transactions on* , vol.12, no.3, pp.370,382, May 1982
- [31] Liu, K.; Subbarayan, S.; Shoults, R.R.; Manry, M.T.; Kwan, C.; Lewis, F.L.; Naccarino, J., "Comparison of very short-term load forecasting techniques," *Power Systems, IEEE Transactions on* , vol.11, no.2, pp.877,882, May 1996
- [32] Papalexopoulos, A.D.; Hesterberg, T.C., "A regression-based approach to short-term system load forecasting," *Power Industry Computer Application Conference, 1989. PICA '89, Conference Papers* , vol., no., pp.414,423, 1-5 May 1989
- [33] Haida, T.; Muto, Shoichi, "Regression based peak load forecasting using a transformation technique," *Power Systems, IEEE Transactions on* , vol.9, no.4, pp.1788,1794, Nov 1994

- [34] Mbamalu, G. A N; El-Hawary, M.E., "Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation," *Power Systems, IEEE Transactions on* , vol.8, no.1, pp.343,348, Feb 1993
- [35] H.M. Al-Hamadi, S.A. Soliman, Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model, *Electric Power Systems Research*, Volume 68, Issue 1, January 2004, Pages 47-59, ISSN 0378-7796
- [36] Wei-Chiang Hong, Electric load forecasting by support vector model, *Applied Mathematical Modelling*, Volume 33, Issue 5, May 2009, Pages 2444-2454, ISSN 0307-904X
- [37] Rahman, S.; Bhatnagar, R., "An expert system based algorithm for short term load forecast," *Power Systems, IEEE Transactions on* , vol.3, no.2, pp.392,399, May 1988
- [38] Ku-Long Ho; Hsu, Yuan-Yih; Chuan-Fu Chen; Tzong-En Lee; Chih-Chien Liang; Tsau-Shin Lai; Kung-Keng Chen, "Short term load forecasting of Taiwan power system using a knowledge-based expert system," *Power Systems, IEEE Transactions on* , vol.5, no.4, pp.1214,1221, Nov 1990
- [39] S. Rahman, O. Hazim, Load forecasting for multiple sites: development of an expert system-based technique, *Electric Power Systems Research*, Volume 39, Issue 3, December 1996, Pages 161-169, ISSN 0378-7796
- [40] Dash, P.K.; Liew, A.C.; Rahman, S., "Fuzzy neural network and fuzzy expert system for load forecasting," *Generation, Transmission and Distribution, IEE Proceedings-* , vol.143, no.1, pp.106,114, Jan 1996
- [41] E.A. Mohamad, M.M. Mansour, S. El-Debeiky, K.G. Mohamad, N.D. Rao, G. Ramakrishna, Results of Egyptian unified grid hourly load forecasting using an artificial neural network with expert system interface, *Electric Power Systems Research*, Volume 39, Issue 3, December 1996, Pages 171-177, ISSN 0378-7796
- [42] Kwang-Ho Kim; Jong-Keun Park; Kab-Ju Hwang; Sung-Hak Kim, "Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems," *Power Systems, IEEE Transactions on* , vol.10, no.3, pp.1534,1539, Aug 1995
- [43] Ahmadi, S.; Bevrani, H.; Jannaty, H., "A fuzzy inference model for short-term load forecasting," *Renewable Energy and Distributed Generation (ICREDG), 2012 Second Iranian Conference on* , vol., no., pp.39,44, 6-8 March 2012
- [44] Soozanchi-K, Z.; Yaghibi, M.; Akbarzadeh-T, M.-R.; Habibipour, M., "Modeling and forecasting short-term electricity load based on multi adaptive neural-fuzzy inference system by using temperature," *Signal Processing Systems (ICSPS), 2010 2nd International Conference on* , vol.3, no., pp.V3-18,V3-22, 5-7 July 2010

- [45] Kodogiannis, V.S.; Petrounias, I., "Power load forecasting using adaptive fuzzy inference neural networks," *Intelligent Systems (IS), 2012 6th IEEE International Conference* , vol., no., pp.238,243, 6-8 Sept. 2012
- [46] Singh, A.K.; Ibraheem, I.; Khatoon, S.; Muazzam, M.; Chaturvedi, D.K., "Load forecasting techniques and methodologies: A review," *Power, Control and Embedded Systems (ICPCES), 2012 2nd International Conference on* , vol., no., pp.1,10, 17-19 Dec. 2012
- [47] Zhang Ming-guang; Li Lin-rong, "Short-term load combined forecasting method based on BPNN and LS-SVM," *Power Engineering and Automation Conference (PEAM), 2011 IEEE* , vol.1, no., pp.319,322, 8-9 Sept. 2011. doi: 10.1109/PEAM.2011.6134865.
- [48] Chia-Nan Ko, Cheng-Ming Lee, "Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter", *Energy*, Volume 49, 1 January 2013, Pages 413-422, ISSN 0360-5442.
- [49] Sajjad Kouhi, Farshid Keynia, A new cascade NN based method to short-term load forecast in deregulated electricity market, *Energy Conversion and Management*, Volume 71, July 2013, Pages 76-83, ISSN 0196-8904.
- [50] Vapnik, Vladimir, Golowich, Steven E. and Smola, Alex J.. "Support Vector Method for Function Approximation, Regression Estimation and Signal Processing.", *Proceedings of the 1996 Neural Information Processing Systems Conference (NIPS 1996)*, Dever, CO, USA, 1997.
- [51] Xin-Chen Guo; Yan-Chun Liang; Chun-Guo Wu; Hao-Yong Wang, "Electric Load Forecasting using SVMS," *Machine Learning and Cybernetics, 2006 International Conference on* , vol., no., pp.4213,4215, 13-16 Aug. 2006. doi: 10.1109/ICMLC.2006.258945
- [52] Turkay, B.E.; Demren, D., "Electrical load forecasting using support vector machines," *Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on* , vol., no., pp.I-49,I-53, 1-4 Dec. 2011
- [53] Ning Ye; Yong Liu; Yong Wang, "Short-term power load forecasting based on SVM," *World Automation Congress (WAC), 2012* , vol., no., pp.47,51, 24-28 June 2012
- [54] Hongzhan Nie, Guohui Liu, Xiaoman Liu, Yong Wang, Hybrid of ARIMA and SVMs for Short-Term Load Forecasting, *Energy Procedia*, Volume 16, Part C, 2012, Pages 1455-1460, ISSN 1876-6102
- [55] T. Down, "Introduction to Neural Computing & Pattern Recognition", *COMP3700*, School of Information Technology and Electrical Engineering, University of Queensland, Australia. URL: <http://www.itee.uq.edu.au/~comp3700/lectures/lecture1.pdf> (accessed on Jan. 31, 2014)

- [56] Hinton, G.E. “How neural networks learn from experience”, *Scientific American*, September 1992. URL: <http://www.cs.toronto.edu/~hinton/absps/sciam92.pdf> (accessed on Jan. 31, 2014)
- [57] W S McCulloch and W Pitts “A logical calculus of ideas immanent in nervous activity”, *Bulletin of Mathematical Biophysics*, Vol 5, pp 115-133, 1943
- [58] Stergiou, C.; and Siganos, D., “Neural Networks”, Department of Computing, Imperial College London. URL: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html (accessed on Jan. 31, 2014)
- [59] Beale, M.; and Demuth, H., “Neural Network toolbox For Use with MATLAB”, *User Guide*, ver. 4, The MathWorks 2001
- [60] Mandic, Donald P.; and Chambers, Jonathon A., “Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability”, John Wiley and Sons, Ltd, 2001.
- [61] Haykin, Simon, “Neural networks: a comprehensive foundation”, Prentice Hall PTR, 1994.
- [62] Hao Yu and B. M. Wilamowski, “Levenberg-Marquardt Training” *Industrial Electronics Handbook, vol. 5 – Intelligent Systems*, 2nd Edition, chapter 12, pp. 12-1 to 12-15, CRC Press 2011
- [63] The Back Propagation Algorithm,
http://www.fer.unizg.hr/download/repository/BP_chapter3_-_bp.pdf
- [64] Government of Canada, Climate Database
URL:http://climate.weather.gc.ca/data_index_e.html
- [65] Nova Scotia Power, Oasis Database
URL:<http://oasis.nspower.ca/en/home/oasis/default.aspx>
- [66] New Brunswick Power, Operations Data
URL:<http://tso.nbspower.com/Public/en/op/market/data.aspx>
- [67] Independent Electricity System Operator (IESO), Power Data
URL:<http://www.ieso.ca/Pages/Power-Data/default.aspx>

Appendix A: Pseudo Code

There are a total of three MATLAB. One imports the raw data for Nova Scotia, New Brunswick and Ontario and makes a database. Second performs data preprocessing steps as explained in chapter 4. Third script makes use of the output of first two programs and generates forecast. Following are all three scripts.

Script 1: data import.m

```
%% Data Import from Excel file and reorganizing for data preprocessing
for i = 1 : 3

    % Raw Data Import
    switch i
        case 1
            clear all
            [num1,txt1,row1] = xlsread('NS0613');
            [num2,txt2,row2] = xlsread('NS0713');
            [num3,txt3,row3] = xlsread('NS0813');

            % Length of Data
            d10 = length(num1(:,1));
            d11 = length(num2(:,1));
            d12 = length(num3(:,1));
            dl = d10 + d11 + d12;

            t10 = length(txt1(:,1));
            t11 = length(txt2(:,1));
            t12 = length(txt3(:,1));

            % Initializing and organizing load, temperature and hour
            information
            % from Raw data to generate a database
            load = zeros(dl,1);
            temp = zeros(dl,1);
            hour = zeros(dl,1);

            for i = 1:d10
                load(i,1) = num1(i,1);
                temp(i,1) = num1(i,3);
                hour(i,1) = num1(i,2);
            end

            for i = 1:d11
                load(i+d10,1) = num2(i,1);
                temp(i+d10,1) = num2(i,3);
                hour(i+d10,1) = num2(i,2);
            end
        end
    end
end
```



```

for i = 1:d12
    load(i+d10+d11,1) = num3(i,1);
    temp(i+d10+d11,1) = num3(i,3);
    hour(i+d10+d11,1) = num3(i,2);
end

dates = {zeros(d1,1)};

for i = 2:t10
    dates{i-1,1} = txt1{i,1};
end

for i = 2:t11
    dates{i-2+t10,1} = txt2{i,1};
end

for i = 2:t12
    dates{i-3+t10+t11,1} = txt3{i,1};
end
dates_v = cellstr(datestr(dates, 'dd/mm/yyyy'));

data.Date = dates_v;
data.Hour = hour;
data.Temp = temp;
data.Load = load;

% Saving the database
dataexcel = [dates_v,num2cell([hour,temp,load])];
header = {'Date', 'Hour', 'Temp (C)', 'Load'};
dataexcel = [header;dataexcel];
save thesis\NS_Data data;
xlswrite('thesis\NS_Data.xlsx', dataexcel);

case 2
clear all
[num1,txt1,raw1] = xlsread('NB0613');
[num2,txt2,raw2] = xlsread('NB0713');
[num3,txt3,raw3] = xlsread('NB0813');

% Length of Data
d10 = length(num1(:,1));
d11 = length(num2(:,1));
d12 = length(num3(:,1));
d1 = d10 + d11 + d12;

t10 = length(txt1(:,1));
t11 = length(txt2(:,1));
t12 = length(txt3(:,1));

% Initializing and organizing load, temperature and hour
information
% from Raw data to generate a database

```

```

load = zeros(d1,1);
temp = zeros(d1,1);
hour = zeros(d1,1);

for i = 1:d10
    load(i,1) = num1(i,1);
    temp(i,1) = num1(i,3);
    hour(i,1) = num1(i,2);
end

for i = 1:d11
    load(i+d10,1) = num2(i,1);
    temp(i+d10,1) = num2(i,3);
    hour(i+d10,1) = num2(i,2);
end

for i = 1:d12
    load(i+d10+d11,1) = num3(i,1);
    temp(i+d10+d11,1) = num3(i,3);
    hour(i+d10+d11,1) = num3(i,2);
end

dates = {zeros(d1,1)};

for i = 2:t10
    dates{i-1,1} = txt1{i,1};
end

for i = 2:t11
    dates{i-2+t10,1} = txt2{i,1};
end

for i = 2:t12
    dates{i-3+t10+t11,1} = txt3{i,1};
end
dates_v = cellstr(datestr(dates, 'dd/mm/yyyy'));

data.Date = dates_v;
data.Hour = hour;
data.Temp = temp;
data.Load = load;

% Saving the database
dataexcel = [dates_v,num2cell([hour,temp,load])];
header = {'Date','Hour','Temp(C)','Load'};
dataexcel = [header;dataexcel];
save thesis\NB_Data data;
xlswrite('thesis\NB_Data.xlsx', dataexcel);

case 3
clear all
[num1,txt1,raw1] = xlsread('ON0613');
[num2,txt2,raw2] = xlsread('ON0713');
[num3,txt3,raw3] = xlsread('ON0813');

```

```

% Length of Data
d10 = length(num1(:,1));
d11 = length(num2(:,1));
d12 = length(num3(:,1));
dl = d10 + d11 + d12;

t10 = length(txt1(:,1));
t11 = length(txt2(:,1));
t12 = length(txt3(:,1));

% Initializing and organizing load, temperature and hour
information
% from Raw data to generate a database
load = zeros(dl,1);
temp = zeros(dl,1);
hour = zeros(dl,1);

for i = 1:d10
    load(i,1) = num1(i,1);
    temp(i,1) = num1(i,3);
    hour(i,1) = num1(i,2);
end

for i = 1:d11
    load(i+d10,1) = num2(i,1);
    temp(i+d10,1) = num2(i,3);
    hour(i+d10,1) = num2(i,2);
end

for i = 1:d12
    load(i+d10+d11,1) = num3(i,1);
    temp(i+d10+d11,1) = num3(i,3);
    hour(i+d10+d11,1) = num3(i,2);
end

dates = {zeros(dl,1)};

for i = 2:t10
    dates{i-1,1} = txt1{i,1};
end

for i = 2:t11
    dates{i-2+t10,1} = txt2{i,1};
end

for i = 2:t12
    dates{i-3+t10+t11,1} = txt3{i,1};
end
dates_v = cellstr(datestr(dates, 'dd/mm/yyyy'));

data.Date = dates_v;
data.Hour = hour;
data.Temp = temp;
data.Load = load;

```

```

        % Saving the database
        dataexcel = [dates_v,num2cell([hour,temp,load])];
        header = {'Date','Hour','Temp(C)','Load'};
        dataexcel = [header;dataexcel];
        save thesis\ON_Data data;
        xlswrite('thesis\ON_Data.xlsx', dataexcel);
    end
end

```

Script 2: data_preprocessing.m

```

%% Data Preprocessing for STLF
clc
clear all

for i = 1 : 3
    switch i
        case 1
            clear all
            load NS_Data
            % ===== Managing dates and days
            =====
            dates_hr = datenum(data.Date,'dd/mm/yyyy') +
            (data.Hour)/24; % Including hour information into Dates
            DOW = weekday(dates_hr);
            % Day of Week
            Day_type = ~ismember(DOW,[1 7]);

            % ===== Normalizing Data between 0 and 1
            =====
            % Normalizing Load Data
            L = data.Load';
            LM = minmax(L);
            L_min = LM(1,1);
            L_max = LM(1,2);
            for i = 1 : length(L)
                Normalized_Load(i) = (L(i) - L_min)/(L_max - L_min);
            end
            LN = Normalized_Load';

            % Normalizing Temperature Data
            T = data.Temp';
            TM = minmax(T);
            T_min = TM(1,1);
            T_max = TM(1,2);
            for i = 1 : length(T)
                Normalized_Temp(i) = (T(i) - T_min)/(T_max - T_min);
            end
            TN = Normalized_Temp';

            % ===== Data Separation into Weekdays and
Weekends =====
            Y = [dates_hr LN TN Day_type];
            k = 1;
            l = 1;

```

```

for i = 1 : length(Y)
    if Y(i,4) == 1
        Dates_workdays(k,1) = Y(i,1);
        for j = 2 : 3
            Data_workdays(k,j-1) = Y(i,j);
        end
        k = k + 1;
    else
        Dates_holidays(l,1) = Y(i,1);
        for j = 2 : 3
            Data_holidays(l,j-1) = Y(i,j);
        end
        l = l + 1;
    end
end

% ===== Inputs for Weekdays
=====
Load_WD = Data_workdays(:,1);
Temp_WD = Data_workdays(:,2);

Load
    YKW1 = [NaN(1,1); Load_WD(1:end-1)];           % Previous Hour
    YKW24 = [NaN(24,1); Load_WD(1:end-24)];       % Previous Day
Same Hour Load
    YKW48 = [NaN(48,1); Load_WD(1:end-48)];      % 2 days before
Same Hour Load
    YKW72 = [NaN(72,1); Load_WD(1:end-72)];      % 3 days before
Same Hour Load
    YKW120 = [NaN(120,1); Load_WD(1:end-120)];   % Previous Week
Same Hour Load
    YKW240 = [NaN(240,1); Load_WD(1:end-240)];   % Previous 2
Weeks Same Hour Load
    TKW = Temp_WD;                               % Current Temp
    TKW1 = [NaN(1,1); Temp_WD(1:end-1)];         % Previous Hour
Temp
    TKW24 = [NaN(24,1); Temp_WD(1:end-24)];     % Previous Day
Same Hour Temp

    Data_WD = [YKW1 YKW24 YKW48 YKW72 YKW120 YKW240 TKW TKW1
TKW24];

% ===== Inputs for Weekends
=====
Load_WE = Data_holidays(:,1);
Temp_WE = Data_holidays(:,2);

Load
    YKH1 = [NaN(1,1); Load_WE(1:end-1)];         % Previous Hour
    YKH48 = [NaN(48,1); Load_WE(1:end-48)];     % Previous Week
same hour load
    YKH72 = [NaN(96,1); Load_WE(1:end-96)];     % Previous Two
Weeks same hour load
    YKH120 = [NaN(144,1); Load_WE(1:end-144)]; % Previous
Three Weeks same hour load

```

```

    TKH = Temp_WE; % Current Temp
    TKH1 = [NaN(1,1); Temp_WE(1:end-1)]; % Previous Hour

Temp

    Data_WE = [YKH1 YKH48 YKH72 YKH120 TKH TKH1];

    % ===== Saving Processed Data
    =====
    save 'thesis\NS_pdata' Dates_workdays Data_WD Load_WD
    Dates_holidays Data_WE Load_WE L_max L_min;

    case 2
        clear all
        load NB_Data
        % ===== Managing dates and days
        =====
        dates_hr = datenum(data.Date, 'dd/mm/yyyy') +
        (data.Hour)/24; % Including hour information into Dates
        DOW = weekday(dates_hr);
    % Day of Week
        Day_type = ~ismember(DOW, [1 7]);

        % ===== Normalizing Data between 0 and 1
        =====
        % Normalizing Load Data
        L = data.Load';
        LM = minmax(L);
        L_min = LM(1,1);
        L_max = LM(1,2);
        for i = 1 : length(L)
            Normalized_Load(i) = (L(i) - L_min)/(L_max - L_min);
        end
        LN = Normalized_Load';

        % Normalizing Temperature Data
        T = data.Temp';
        TM = minmax(T);
        T_min = TM(1,1);
        T_max = TM(1,2);
        for i = 1 : length(T)
            Normalized_Temp(i) = (T(i) - T_min)/(T_max - T_min);
        end
        TN = Normalized_Temp';

        % ===== Data Separation into Weekdays and
Weekends =====
        Y = [dates_hr LN TN Day_type];
        k = 1;
        l = 1;

        for i = 1 : length(Y)
            if Y(i,4) == 1
                Dates_workdays(k,1) = Y(i,1);
                for j = 2 : 3
                    Data_workdays(k,j-1) = Y(i,j);
                end
            end
        end
    end
end

```

```

        end
        k = k + 1;
    else
        Dates_holidays(1,1) = Y(i,1);
        for j = 2 : 3
            Data_holidays(1,j-1) = Y(i,j);
        end
        l = l + 1;
    end
end
end

% ===== Inputs for Weekdays
=====
Load_WD = Data_workdays(:,1);
Temp_WD = Data_workdays(:,2);

Load
    YKW1 = [NaN(1,1); Load_WD(1:end-1)];           % Previous Hour
Same Hour Load
    YKW24 = [NaN(24,1); Load_WD(1:end-24)];       % Previous Day
Same Hour Load
    YKW48 = [NaN(48,1); Load_WD(1:end-48)];       % 2 days before
Same Hour Load
    YKW72 = [NaN(72,1); Load_WD(1:end-72)];       % 3 days before
Same Hour Load
    YKW120 = [NaN(120,1); Load_WD(1:end-120)];    % Previous Week
Same Hour Load
    YKW240 = [NaN(240,1); Load_WD(1:end-240)];   % Previous 2
Weeks Same Hour Load
    TKW = Temp_WD;                               % Current Temp
    TKW1 = [NaN(1,1); Temp_WD(1:end-1)];         % Previous Hour
Temp
    TKW24 = [NaN(24,1); Temp_WD(1:end-24)];      % Previous Day
Same Hour Temp

    Data_WD = [YKW1 YKW24 YKW48 YKW72 YKW120 YKW240 TKW TKW1
TKW24];

% ===== Inputs for Weekends
=====
Load_WE = Data_holidays(:,1);
Temp_WE = Data_holidays(:,2);

Load
    YKH1 = [NaN(1,1); Load_WE(1:end-1)];          % Previous Hour
same hour load
    YKH48 = [NaN(48,1); Load_WE(1:end-48)];       % Previous Week
Weeks same hour load
    YKH72 = [NaN(96,1); Load_WE(1:end-96)];       % Previous Two
Three Weeks same hour load
    YKH120 = [NaN(144,1); Load_WE(1:end-144)];    % Previous
Temp
    TKH = Temp_WE;                               % Current Temp
    TKH1 = [NaN(1,1); Temp_WE(1:end-1)];         % Previous Hour

    Data_WE = [YKH1 YKH48 YKH72 YKH120 TKH TKH1];

```

```

% ===== Saving Processed Data
=====
save 'thesis\NB_pdata' Dates_workdays Data_WD Load_WD
Dates_holidays Data_WE Load_WE L_max L_min;

case 3
clear all
load ON_Data
% ===== Managing dates and days
=====
dates_hr = datenum(data.Date, 'dd/mm/yyyy') +
(data.Hour)/24; % Including hour information into Dates
DOW = weekday(dates_hr);
% Day of Week
Day_type = ~ismember(DOW, [1 7]);

% ===== Normalizing Data between 0 and 1
=====
% Normalizing Load Data
L = data.Load';
LM = minmax(L);
L_min = LM(1,1);
L_max = LM(1,2);
for i = 1 : length(L)
Normalized_Load(i) = (L(i) - L_min)/(L_max - L_min);
end
LN = Normalized_Load';

% Normalizing Temperature Data
T = data.Temp';
TM = minmax(T);
T_min = TM(1,1);
T_max = TM(1,2);
for i = 1 : length(T)
Normalized_Temp(i) = (T(i) - T_min)/(T_max - T_min);
end
TN = Normalized_Temp';

% ===== Data Separation into Weekdays and
Weekends =====
Y = [dates_hr LN TN Day_type];
k = 1;
l = 1;

for i = 1 : length(Y)
if Y(i,4) == 1
Dates_workdays(k,1) = Y(i,1);
for j = 2 : 3
Data_workdays(k,j-1) = Y(i,j);
end
k = k + 1;
else
Dates_holidays(l,1) = Y(i,1);
for j = 2 : 3
Data_holidays(l,j-1) = Y(i,j);
end
l = l + 1;
end
end

```



```

        end
        l = l + 1;
    end
end

% ===== Inputs for Weekdays
=====
Load_WD = Data_workdays(:,1);
Temp_WD = Data_workdays(:,2);

Load
    YKW1 = [NaN(1,1); Load_WD(1:end-1)];           % Previous Hour
Same Hour Load
    YKW24 = [NaN(24,1); Load_WD(1:end-24)];        % Previous Day
Same Hour Load
    YKW48 = [NaN(48,1); Load_WD(1:end-48)];        % 2 days before
Same Hour Load
    YKW72 = [NaN(72,1); Load_WD(1:end-72)];        % 3 days before
Same Hour Load
    YKW120 = [NaN(120,1); Load_WD(1:end-120)];     % Previous Week
Same Hour Load
    YKW240 = [NaN(240,1); Load_WD(1:end-240)];    % Previous 2
Weeks Same Hour Load
    TKW = Temp_WD;                                 % Current Temp
    TKW1 = [NaN(1,1); Temp_WD(1:end-1)];           % Previous Hour
Temp
    TKW24 = [NaN(24,1); Temp_WD(1:end-24)];        % Previous Day
Same Hour Temp

Data_WD = [YKW1 YKW24 YKW48 YKW72 YKW120 YKW240 TKW TKW1
TKW24];

% ===== Inputs for Weekends
=====
Load_WE = Data_holidays(:,1);
Temp_WE = Data_holidays(:,2);

Load
    YKH1 = [NaN(1,1); Load_WE(1:end-1)];           % Previous Hour
same hour load
    YKH48 = [NaN(48,1); Load_WE(1:end-48)];        % Previous Week
Weeks same hour load
    YKH72 = [NaN(96,1); Load_WE(1:end-96)];        % Previous Two
Three Weeks same hour load
    YKH120 = [NaN(144,1); Load_WE(1:end-144)];    % Previous
Temp
    TKH = Temp_WE;                                 % Current Temp
    TKH1 = [NaN(1,1); Temp_WE(1:end-1)];           % Previous Hour

Data_WE = [YKH1 YKH48 YKH72 YKH120 TKH TKH1];

% ===== Saving Processed Data
=====
save 'thesis\ON_pdata' Dates_workdays Data_WD Load_WD
Dates_holidays Data_WE Load_WE L_max L_min;
end

```

end

Script 3: stlf.m

```
%% Performance Evaluation of New and Advanced Neural Networks
clc
clear all

forecast = 1;
while forecast>0,
    clear all
    % ===== Region Selection
    =====
    region = menu('Select a Region for forecast',...
        'Nova Scotia',...
        'New Brunswick',...
        'Ontario');

    switch region
    case 1
        load NS_Data
        load NS_pdata.mat
    case 2
        load NB_Data
        load NB_pdata.mat
    case 3
        load ON_Data
        load ON_pdata.mat
    end

    % ===== Data Selection and Splitting
    =====
    dtype = menu('Select Day type',...
        'Weekday',...
        'Weekend');

    switch dtype
    case 1
        period = menu('Select Time period for which forecast is
requires',...
            'Day Ahead',...
            'Week Ahead');
        switch period
        case 1
            % Training data: from June 1 to August 25, 2013
            train_data = floor(Dates_workdays) <
datenum('08/26/2013');
            trainX = Data_WD(train_data,:);
            trainY = Load_WD(train_data);
            train_Dates = Dates_workdays(train_data);

            % Test data: August 26, 2013
```

```

        test_data = floor(Dates_workdays) ==
datenum('08/26/2013');
        testX = Data_WD(test_data,:);
        testY = Load_WD(test_data);
        test_Dates = Dates_workdays(test_data);
    case 2
        % Training data: from June 1 to August 25, 2013
        train_data = floor(Dates_workdays) <
datenum('08/26/2013');
        trainX = Data_WD(train_data,:);
        trainY = Load_WD(train_data);
        train_Dates = Dates_workdays(train_data);

        % Test data: August 26, 2013
        test_data = floor(Dates_workdays) >=
datenum('08/26/2013');
        testX = Data_WD(test_data,:);
        testY = Load_WD(test_data);
        test_Dates = Dates_workdays(test_data);
    end
case 2
    % Training data: from June 1 to August 25, 2013
    train_data = floor(Dates_holidays) < datenum('08/24/2013');
    trainX = Data_WE(train_data,:);
    trainY = Load_WE(train_data);
    train_Dates = Dates_holidays(train_data);

    % Test data: August 26, 2013
    test_data = floor(Dates_holidays) >= datenum('08/24/2013');
    testX = Data_WE(test_data,:);
    testY = Load_WE(test_data);
    test_Dates = Dates_holidays(test_data);
end

% ===== Network Selection and Initialization
=====
network = menu('Select the network',...
    'FFNN',...Feedforward NN
    'CNN',...Cascaded NN
    'RNNL',...Recurrent NN with Local feedback
    'RNNG',...Recurrent NN with Global feedback
    'CRNNL',...Cascade-Recurrent NN with Local feedback
    'CRNNG',...Cascade-Recurrent NN with Global feedback
    'FFNN (3L)',...3-layered feedforward NN
    'CNN (3L)',...3-layered cascaded NN
);
switch network
case 1 % FFNN
    net = fitnet(9);
    net.biasConnect = [0; 0];
case 2 % CNN
    net = fitnet(9);
    net.biasConnect = [0; 0];
    net.inputConnect = [1; 1];
case 3 % RNNL
    net = fitnet(9);

```

```

        net.biasConnect = [0; 0];
        net.inputConnect = [1; 0];
        net.layerConnect = [1 0; 1 0];
        net.layerWeights{1,1}.delays = 1;
    case 4 % RNNNG
        net = fitnet(9);
        net.biasConnect = [0; 0];
        net.inputConnect = [1; 0];
        net.layerConnect = [0 1; 1 0];
        net.layerWeights{1,2}.delays = 1;
    case 5 % CRNNL
        net = fitnet(9);
        net.biasConnect = [0; 0];
        net.inputConnect = [1; 1];
        net.layerConnect = [1 0; 1 0];
        net.layerWeights{1,1}.delays = 1;
    case 6 % CRNNG
        net = fitnet(9);
        net.biasConnect = [0; 0];
        net.inputConnect = [1; 1];
        net.layerConnect = [0 1; 1 0];
        net.layerWeights{1,2}.delays = 1;
    case 7 % FFNN (3L)
        net = feedforwardnet([9 18]);
        net.biasConnect = [0; 0; 0];
    case 8 % CNN (3L)
        net = feedforwardnet([9 9]);
        net.biasConnect = [0; 0; 0];
        net.inputConnect = [1; 1; 1];
        net.layerConnect = [0 0 0; 1 0 0; 1 1 0];
end

% ===== Network Training and Forecasting
=====
% Network Training
net = train(net, trainX', trainY');
view(net)
% Forecasting
Output = sim(net, testX')';

% ===== Error Analysis
=====
% Data Post Processing
for i = 1 : length(testY)
    Actual(i,1) = (testY(i)*(L_max - L_min)) + L_min;
    Predicted(i,1) = (Output(i)*(L_max - L_min)) + L_min;
end

% Calculating Errors (MAPE and RMSE)
P = 0;
M = 0;
V = 0;
D = 0;
N = length(Actual);
Per = zeros(N,1);
Hour = zeros(N,1);

```

```

for i = 1 : N
    E(i,1) = (Predicted(i)-Actual(i));
    D = D + abs(E(i));
    Per(i,1) = (E(i)/Actual(i))*100;
    P = P + ((abs(E(i)) / Actual(i)));
    M = M + (Predicted(i)-Actual(i))^2;
    Hour(i,1) = i;
end

Table = [Hour Actual Predicted E Per]
MAPE = P*(100/N)
RMSE = sqrt(M/N)

% ===== Plotting Results and View Network
=====
figure;
subplot(2,1,1);
plot(Hour,Actual, '-k',Hour,Predicted, '--r');
l = legend('Actual', 'Forecasted');
subplot(2,1,2);
plot(Hour,E, 'b');

% ===== Another forecast or end the program
=====
question = menu('Do you want to generate another forecast', 'Yes',
'No');
if question == 1
    forecast = 1;
elseif question == 2
    forecast = 0;
end
end

```