University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Computer Science	Spring semester, 2018 Open
Writer: Azadeh Karimi	………………………………………… (Writer's signature)
Faculty supervisor: Professor Chunming Rong Dr. Songpu AI External supervisor(s):	
Thesis title: **Prediction of Energy Consumption Peak in Household by using LSTM & MLP**	
Credits (ECTS): 30	
Key words: LSTM, RNN, Peak Load Forecasting, Deep learning, MAPE, TensorFlow	Pages: ………39………… (Including Abstract + References +other: 49) Stavanger, 15/06/2018 Date/year

Front page for master thesis
Faculty of Science and
Technology
Decision made by the Dean October 30th
2009

Prediction of Energy Consumption Peak in Household by using LSTM & MLP

Azadeh Karimi

Department of Electrical Engineering and Computer Science

Faculty of Science and Technology

University of Stavanger

June 2018

Abstract

Accurate peak load forecasting plays a key role in operation and planning of electrical power generation. To minimize the operating cost, electric suppliers use forecasted peak load to control the number of running generator units. One of the most precise load forecasting methods is deep neural networks (DNNs), which is categorized under artificial neural networks (ANNs).

In the past few decades, DNNs have appeared as a powerful tool in machine learning filed. DNNs have been shown to significantly outperform the other traditional methods in many applications, and they have completely revolutionized some fields. Given their success in other machine learning problems, DNNs are applied in energy forecasting.

ANN has recently applied on short-term load forecasting in electrical utilities. In this thesis, two ANN algorithms for predicting peak load has been used. Multilayer Perceptron and Long Short-Term Memory. Then, the performance of the models was compared to find out the error in peak load forecasting. Error here refers to the difference between actual loads and predicted ones. The result based on in our study revealed that Long Short-Term Memory has less mean absolute percentage error (MAPE) in compare with Multilayer Perceptron.

Keywords: LSTM, RNN, Peak Load Forecasting, Deep learning; MAPE, TensorFlow.

Acknowledgements

I would first like to thank my thesis supervisor, Professor Chunming Rong at the University of Stavanger. Professor Rong has shown both a great interest and knowledge in the field and has provided me with interesting literature and feedback. Thank you for your guidance during the time of writing this thesis.

I would also like to thank Songpu Ai, my co-supervisor for his valuable advises and contributions. The thesis would have never been possible without his help. He was always available whenever I needed help. He made me keep my track on the work and further improvement of the project.

Most importantly, I am deeply grateful to my family, especially my husband Ahmad for everything. You have motivated me with kind and supportive words, laughter, encouragements and handled other tasks so that I could work with the thesis.

Last but not the least, I would like to thank entire friends for making my project successful.

Azadeh Karimi

University of Stavanger

Table of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

1.1 General

Load forecasting is a crucial component for energy management system (EMS). An accurate and reliable prediction model helps an electric utility to make valuable decisions on power market, to optimize the energy consumption, to save the energy and money, to achieve voltage control, to accomplish maintenance and to contribution infrastructure development.

Because of fast development of the technologies of Internet of Things (IoT), micro-generation and storage, we need house's EMS. Prediction of EMS is important issue because we need to shift the load, buy the load from local or share our generated data [36]. Thus power prediction is very critical for future (i.e. 5 or 10 years later).

The electricity price depends upon the efficient operation of an electrical power generation plant. This efficiency means, it should be a balance between consumption and supply of electric load. In order to achieve this balance, there should be an accurate forecasting model which can predict the load requirement in future or at any time. Inaccurate load forecasting will influence the scheduling and planning of power systems. On the other hand, supplying power less than consumption will bring about lack of energy and also supplying more than consumption leads to power waste. The efficient operation of a power system needs proper fuel scheduling and maintenance which can be achieved by using a sophisticated forecasting system.

The peak load prediction is even more important. Because if we find the predicted peak, we can struggle with moving potential peak load to the off peak period to smooth the peak, making consumption curve flat and reduce the prediction damages. An accurate peak load forecasting has a more substantial role today with the new open access operating environment of electricity supply industries. Reliable peak load prediction is necessary to determine generation, transmission, distribution and investment as required to reserve generation in the future on forecasted loads.

Load forecasting is one of the main research areas in the smart grid and can be classified depending on its target forecasting ranges. In general, load forecasting is divided into three categories:

- **Short-term load forecasting (STLF):** The forecast period of STLF generally, starts from a few minutes to a day or a week. Short-term load forecasting is playing an essential role in day to day operation because of its wide applicability on demand-side management, energy storage operation, peak load reduction and planning activities of the utility system [7]. The prediction target usually is the load capacity of a region or the daily and weekly electricity consumption data. The forecasting data generally indicates daily or monthly periodicity. One-year prediction also follows a similar periodic pattern [9].

- **Medium-Term Load Forecasting (MTLF):** The prediction period of MLTF starts from a week to one year and is used for the purpose of scheduling fuel supplies and unit maintenance [20]. The prediction target usually is the load capacity of a region or the monthly electricity consumption. The forecast data generally indicates cyclical growth; Each month of a year consists of the similar growth pattern [14]. The prediction goal is to arrange monthly maintenance plan, reservoir operation plans, operation mode and coal transportation plans. The main factors affecting medium-term load forecasting would be production planning from large users, weather conditions, industrial restructuring situations and national tariff policy and so on [15]. Currently, the relatively mature prediction methods are trend extrapolation based on the historical data of the same month and the time series prediction methods based on the yearly data.

- **Long-Term Load Forecasting (LTLF):** The prediction period of MLTF is normally longer than a year and up to 15 years or even longer. With accurate forecasting of LTLF, the power supply companies make investments and put a plan of regarding maintenance [19]. The prediction target is usually the annual electricity consumption. The prediction purpose is to provide the base data for the power grid planning that helps to determine the grid operation mode and to make annual maintenance plans. The main factors affecting long-term load forecasting are national economic development, population, industrial restructuring and national tariff policy and so on [23].

In this thesis, the focus of our research is on STLF to forecast the peak load for 10 minutes' intervals. Minutes period is very short load forecasting, which is important for the real-time scheduling of electricity generation.

In last few decades, STLF has been one of the most important research topics for the achievement of higher efficiency and reliability in power system operation. STLF plays a decisive role in the real-time control and the security functions of an EMS. Moreover, a correct peak load forecast can be helpful in developing a power supply strategy, financing planning, electricity management and market search. STLF is a very complex process because there are many factors can influence it, such as economic conditions, time, day, season, weather and random effects [1].

1.2 Contributions of the Study

In this study forecasting of peak power consumption using different ANN architectures based on previous studies is investigated. Long Short-Term Memory (LSTM) and Multilayer Perceptron (MLP) are then adopted to forecast the peak load in 10 minutes' intervals.

A case study with the above methods is performed. The result from the case study reveals that LSTM model performs more accurate than MLP.

1.3 Outline of the Thesis

The thesis is divided into six chapters. After an introduction to the subject in Chapter 1, then Chapter 2 describes the related works and reviews the existing literature in the subject matter. Chapter 3 starts with a brief background about deep learning discussing the machine learning and artificial neural network. In this section, our goal is first to introduce the two load forecasting models, MLP and LSTM with their differences in their approaches. Secondly, the activation functions and error analyses are also described here. Chapter 4 concentrates on data preparation leading to training and testing dataset and the neural network system design. Chapter 5 offers a comprehensive discussion on the MLP and LSTM models. Simulation results obtained here showed that the LSTM approach is superior to MLP model. Finally, last chapter, Chapter 6, draws final conclusions and also suggests further works in NN-based load forecasting.

Chapter 2

2. Related Work

Load forecasting is not a new research topic. In the following other researchers' works on this subject has briefly been addressed. This chapter has been divided into two sections; Section 2.1 consists of important prior research works on load forecasting subject as general.

Section 2.2 discusses our model and compares with other surveys that address the peak load prediction.

2.1 Related Works on Load Forecasting

For load forecasting, [2] applied several algorithms such as Weighted Moving Average (WMA), Multiple Linear Regression (MLR), quadratic (MQR), Regression Tree (RT), Support Vector Regression (SVR) and Multilayer Perceptron (MLP) and realized that MLP has the best performance. [4] concentrated on identifying different types of time series and concluded that a recurrent network provides a significantly better model. The result shows that recurrent neural networks (RNNs) are types of nonlinear autoregressive-moving average (NARMA) model which gives the best performance on electric load forecasting. [6] illustrated how deep Neural Network is applied for short-term load forecasting where various combinations of activation functions were tested to achieve better Mean Absolute percentage error (MAPE). The paper tested functions are Sigmoid, Rectifier linear unit (ReLU) and Exponential linear unit (ELU). The result shows significant savings in the MAPE values using the ELU function over the other activation functions. Paper [7] implemented a neural network (NN)-based method for the construction of prediction intervals (PIs) and introduced a method, called lower upper bound estimation (LUBE). His comparative results show that the proposed method can construct higher quality PIs for load and wind power generation forecasts in a short time. In [10] LSTM model for dynamic system identification investigated. The result indicates that LSTM structure performs much better than conventional RNN and even single LSTM network.

Paper [11] introduced a new approach to the short-term load forecasting of the next day by using autoregressive (AR) and artificial neural network (ANN) models. Inputs to the system are days of the weeks and output is next day. For the systems that use AR and ANN models, there are 7 neurons in the input layer and one neuron in the output layer. This paper found that systems which use both AR and ANN models can achieve a higher forecasting accuracy.

[19] introduced a methodology for long-term electric power demands using a semi-group based system-type neural network architecture. This methodology is applied to recent load data, and the next year's load data is satisfactorily forecasted. [8] Applied Cartesian Genetic Algorithm evolved Artificial Neural Network (CGPANN) for the prediction of peak loads 24 hours ahead. The network for every individual season and for the whole year trained. The results are the best for autumn 2009 with having less variation in peak load. In [9] two operative algorithms for predicting peak load is used, MLP and Radial Basis Function (RBF). The comparison then has been done between the methods to show error in peak load forecasting. The result shows that in this case, Multi-layer perceptron has more accuracy than Radial basis function. Another valuable paper around peak load forecasting is [12] which presents an ANN model for daily peak load forecasting. Moreover, the model presents unique adjustment algorithm to compensate for the negative impact of holidays' forecasts. The result indicates that the prediction of the model over a period of one year is reliable including holidays.

2.2 Related Works on Peak Load Forecasting

[13] proposes a new training method for the analyzable structured neural network (ASNN) in order to realize accurate daily peak load forecasting. The result presented confirms that ASNN trained by new training method can explain forecasting more properly than ASNN trained by the conventional methods. In [17] daily peak load forecasting has been performed for the part of a town supplied by 19 distribution feeders on weekdays by taking the historical maximum load (Lmax) and maximum temperature (Tmax) data into account. The optimized network performances were then compared in terms of the MAPE and the network complexity. [18] illustrates a peak load forecast models using ANN with MLP and back-propagation (BP) learning algorithm. It demonstrates the accuracy of the proposed method and shows that the forecast model is simple with high accuracy.

In our research, we have used ANN to forecast the peak consumption. Here we will compare our method with previous related work. Paper [2], [6], [8], [9] and [11] have got good result with MLP model for load prediction. [4] also chose the normal recurrent neural network (RNN) as the best solution for load forecasting.

However, the weakness of these researches is that they have applied simple ANN structure such as MLP or RNN without memory. Memory is necessary because each neuron or unit can use its internal memory to maintain information about the previous input and it is especially useful with sequential data. Traditional neural networks cannot do this. In our case special type of recurrent neural network with the memory cell which is called LSTM is applied. However, due to memory feature of the unites, LSTM could perform better than MLP.

Based on our result, we have discovered that the LSTM is a good solution for load forecasting. Paper [10] used the LSTM model only for the load forecasting, but our result still is necessary because, as it was discussed in the previous chapter, we have found the peak prediction is quite important as well.

Also, we have found [8], [9], [12], [13] and [18], all predicted the peak load, but with MLP model, not LSTM which is the weakness of these research.

According to the above papers, we can conclude that our investigation is the first research about peak load forecasting by LSTM in 10 minutes' intervals.

Chapter 3

3. Technologies

3.1 Background

Machine learning is the subfield of computer science that focuses on the construction of algorithms that make predictions based on input data. It is also can be used on the classification of unlabeled data and etc. A machine learning algorithm uses computational methods to learn like a human which maps the input data onto the output. On the other hand, this science is implemented like a human's brain. The main issue in machine learning is that it should have access to enough data in order to gain experience. In data analytics field, machine learning is a procedure contains complex models and algorithms that are used for prediction. These analytical models generate precise, repeatable decisions and results from historical relationships [22].

Deep learning (DL) is a type of machine learning that contains many layers of feature-detecting "neurons" between the input and the output. The input layer receives an input and passes on a modified version of the input to next layer. Deep learning subfield such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been applied to fields such as computer vision, prediction, speech recognition, language modeling and audio recognition. The phrase "deep learning" in ANN field was introduced by Igor Aizenberg in 2000 [34]. The structure of NN is shown in Figure (1).

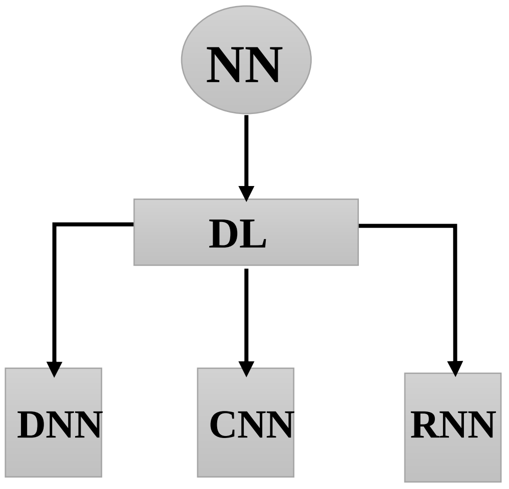

Figure 1: Structure of Neural Network.

DNN is a neural network with a certain level of complexity which contains certain the number of neurons with multiple hidden layers between the input and output layers. DNN is usually used to handle complicated recognition tasks and also, requires data with good quality as qualified data affects the result of the network.

RNN has become a widely-used research topic in machine learning field nowadays. It is used for the time series prediction with long temporal dependency. It can learn sequential information by considering information from previous time steps. The basic idea is recurrent networks have loops. These loops allow the network to use information from previous passes, which acts as a memory. The length of this memory depends on the number of factors, but it is important to note that it is not indefinite. We can use RNNs in different problems like time series analysis, natural language processing, and speech recognition.

CNN is a kind of feedforward neural network technique. A feedforward neural network is an ANN where connections between the units are not formed of the loop. CNNs is widely used in image and video recognition and natural language processing. Usually, the structure of CNN contains neurons with specific weights and biases. Each neuron gets inputs, takes a weighted sum over them, pass it through an activation function and generates the output.

In this thesis we tested MLP and LSTM networks to forecast the peak load by using real power consumption data approximately available in four months. The motivation for using deep learning techniques is to explore if they are able to produce better results than other models.

3.2. Load Forecasting Techniques

The load forecasting consists of various methods. Some of the widely used ones are mentioned here. One of the parametric methods for forecasting is regression techniques. Regression analysis is a statistical procedure for estimating the relationship between dependent and predictor (independent) variables. It allows one to see how the dependent variable changes with respect to the changes in the independent variable [37]. One limitation of this method is that it is unsuitable for the complex modeling techniques and heavy computational efforts. It is more convenient for the linear data without complexity relationship.

Exponential smoothing is one of the classical methods used for load forecasting. The approach is based on previous data like other forecast methods, then to use this model to predict the future load. This method is also easy to learn and apply. The disadvantage of these approaches is that it produces forecasts that lag behind the trend. The lag is a side effect of the smoothing process. There is a reason this method has “smoothing” in the naming because it neglects the ups and downs associated with random variation [38].

Another load forecasting technique is fuzzy logic. It is well known that a fuzzy logic system with centroid defuzzification can identify and approximate any unknown dynamic system. This approach is easy to understand and implement [39]. However, it has some disadvantages like hard to develop a model from a fuzzy system or finding suitable membership values for fuzzy systems.

ANN is one of the well-known methods of the load forecasting. The ANN is behaved same as human brain by learning and memorizing different tasks. ANNs are like humans, learn by example. It gets enough input as a training set and after processing them can predict the future data. ANNs are applied for prediction problems when known and reliable system input/output sets are available. Fast system prediction is required as well as the system is complicated and difficult to express in mathematical formulas. In general, the ANN is able to predict any system accurately and rapidly no matter how complex the system is.

An ANN is made up of simple processing units called neurons which are the basic processing components of artificial neural networks. Each part of the neuron plays a specific role in communicating information throughout the body. In fact, the neuron is a set of inputs, a set of weights and an activation function. Every neuron is connected with other neurons through a directional link. Each connection link is associated with a specific weight. Each neuron can take multiple weighted inputs and then applies the activation function to the summation of these inputs in order to generate an output. The inputs can be either the raw input features like prior load data or other external features such as temperature and wind speed for the model. The output of each layer can then be picked up as input for another layer of neurons later on. The network output is calculated with the following equation:

$$Y = f(w_1 . x_1 + w_2 . x_2 + \cdots + w_n . x_n) + b$$

In above equation b denotes bias. A bias unit is an extra neuron added to each pre-output layer that stores the value of 1. Bias units aren't connected to any previous layer and it is added into neurons to increase the flexibility of the model in order to fit the data.

Each neuron has a weight vector *w*= $(w_1, w_2, \ldots, w_n)$, where *n* is the number of inputs to that neuron and Y is the final output. Neuron takes the weighted inputs, sums them up, and then sends them to the activation function f. Figure 2 shows the structure of the artificial neuron.

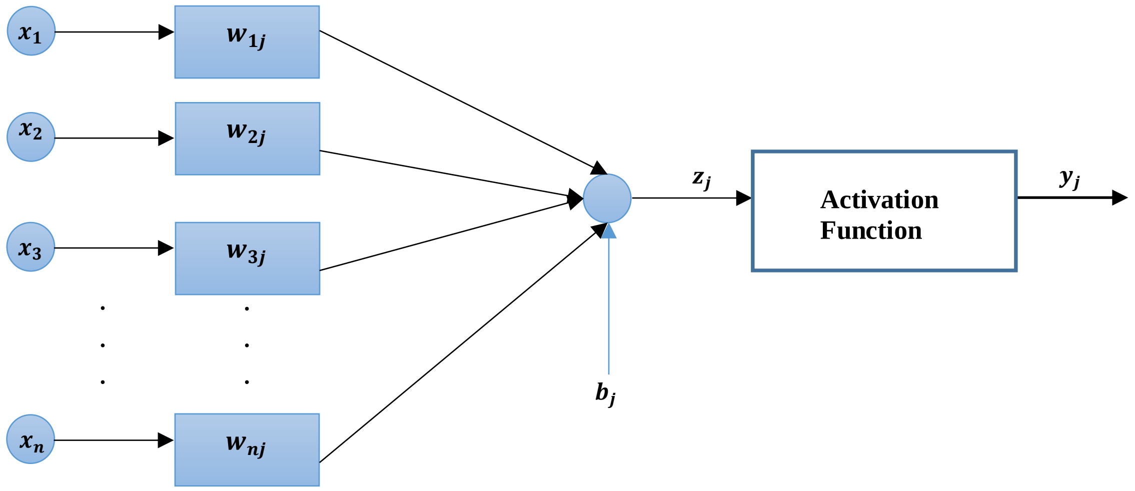

Figure 2 : Structure of the Artificial Neuron.

The activation function which is also known as a transfer function is added to the output end of any neural network. It maps out the resulting values to a value between 0 to 1.

Activation functions are really essential elements for an ANN. The role of the activation function in a neural network is to produce a non-linear decision boundary via non-linear combinations of the weighted inputs. Some types of activation functions will be explained in the following;

- logistic sigmoid (σ)

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- Hyperbolic tangent (*tanh*)

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

- rectified linear units (*ReLU*)

$$ReLU(x) = \max(0, z)$$

ANN is a system of connected artificial neurons, and it has the ability to model any arbitrary non-linear function [13], Whereas load forecasting data are non-linear in nature. Thus among the above-mentioned methods, artificial intelligence methods are a good candidate. ANN can solve complex problems with high accuracy such as pattern classification and nonlinear mapping.

ANN models, by using the historical data and training technique, can learn correlated factors between input datasets and matching target values. The created model can be used to forecast the outcome of new independent input data.

There are several types of the neural network structures. As an example is MLP which consists of the input layer, output layer, and more than one hidden layer. Another example is RNN that has internal memory to process sequences of inputs. In fact, this memory is a state function like a feedback loop between neurons. LSTM is another one that is kind of RNNs which are designed to avoid the long-term dependency problem by using forget gate. Gated Recurrent Unit (GRU) is also similar to the LSTM but GRU, unlike LSTM unit, does not have to use a memory unit to control the flow of information.

3.3 Load Prediction Models

In this section, the two DNN models are described. These models are used for forecasting of load energy consumption. In Section 3.3.1 standard MLP is explained, followed by the LSTM presented in Section 3.3.2.

3.3.1 Multilayer Perceptron (MLP)

A typical structure of ANN is based on multilayer perceptron. MLP is the most frequently used type of neural network model that forecasts a future load by using previous load data.

The MLP has a three-layered structure; input, hidden and output layers. However, between the layers, there exist full weighted connections among neurons. The Input layer is connected to the one or more hidden layer and the last hidden layer connected to the output layer. Outputs of each layer are submitted to inputs of the next layer and all these layers are connected with weights. The weighted inputs are summed up and passed through an activation function.

If neural network model has one hidden layer it is called single layer perceptron. This model is very simple and it is suitable for the non-complex problems. However, if the hidden layers increase more than one, for example, two or three it is then called multilayer perception. By increasing the number of hidden layers the complexity will be increased.

MLP is different from a RNN network. The difference is the connections between the nodes. In MLP network, the weights from each of the other nodes on the same level of the layer are pushed forward into the hidden layer. It means that the input variables will constantly be pushed forward to the output nodes.

The structure implies that the MLP has the power of manipulating the input space by adjusting weight matrices between layers. The main process of building MLP is to calculate appropriate weight matrices that produce the desired output corresponding to given data. The difference between the network outputs which is our prediction and the target values is called the error.

The backpropagation (BP) algorithm is an optimizer which is used to update and optimize weight matrices to minimize the error between the target value and the predicted value that produced by MLP. BP is a commonly used method for training an MLP network. The idea behind BP algorithm is quite simple, the output of NN is evaluated against desired output. If results are not satisfactory. The error is calculated and propagated backward from the output to the hidden layer and then to the input. The weights are adjusted to make the error between the actual response and the desired response smaller than a target value. The process is repeated again and again until to reach the minimum error.

An MLP model with two-hidden layers which is applied BP is shown in Figure (3).

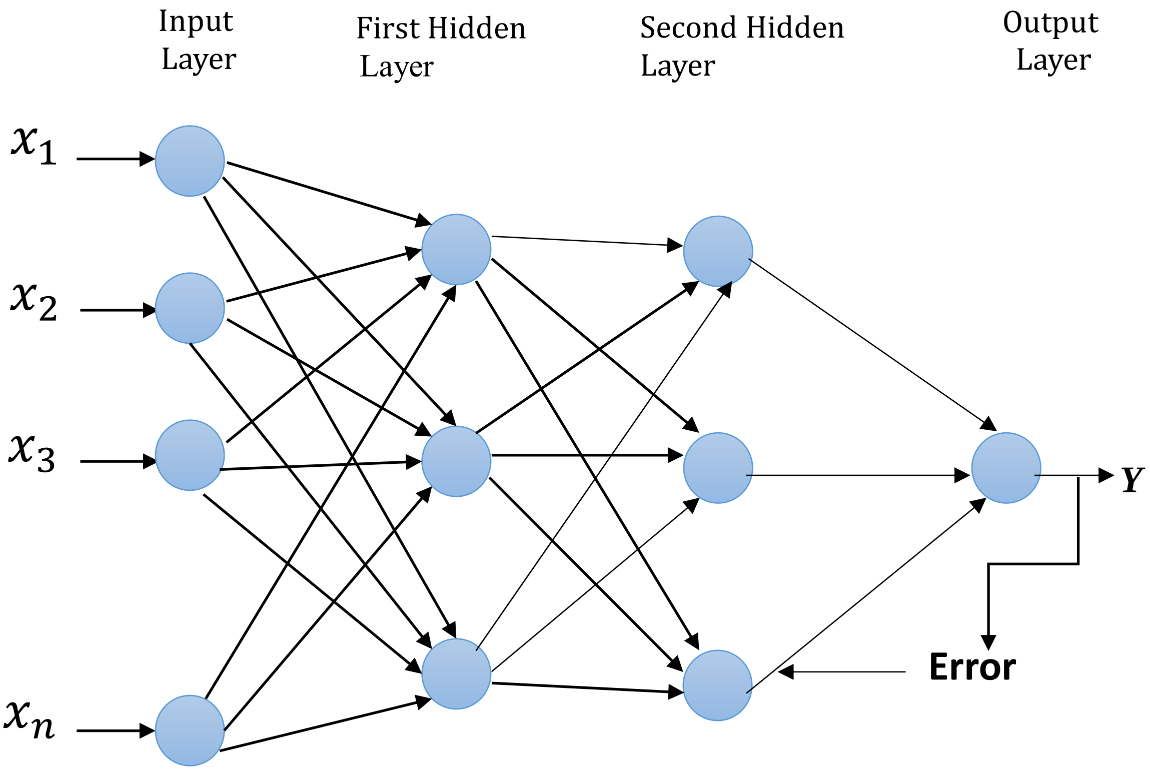

Figure 3: The MLP based peak load model by using feed forward and (BP).

3.3.2 Long Short-Term Memory (LSTM)

As it was discussed in Chapter 2, the LSTM model is required for peak load prediction.An LSTM has a special neuron structure called memory cell. These memory cells have the ability to store information over an arbitrary time. Three gates (forget, input and output gates) are controlling the information flow into and out of the neuron’s memory cell [24].

Each gate in the LSTM receives the same input as the neuron input. Furthermore, each gate possesses an activation function. Figure (4) displays LSTM unit cell.

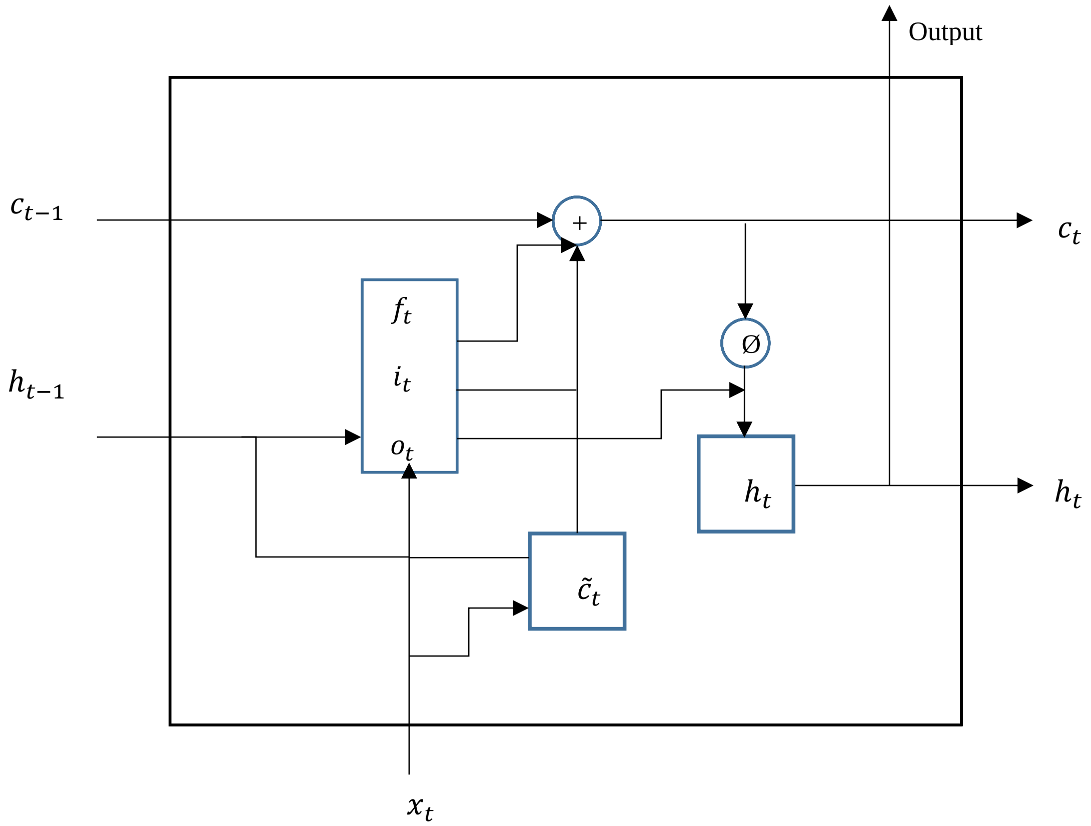

Figure 4: LSTM Memory Cell.

The main components of the LSTM unit are:

1. Input: input vector defined as x_t and output denoted by h_{t-1} from the previous time step is taken by LSTM unit. The weighted inputs are summed up and passed through $tanh$ activation function, the result is shown in z_t .

   $$z_t \; = tanh(W_z . x_t \; + R_z \, h_{t-1} \; + b_z \,)$$

   Where W_z denotes input weights, R_z is recurrent weight, and b_z represents the biases.

2. Input gate: This gate reads x_t and h_{t-1} , computes then the weighted sum, and applies sigmoid activation. The result i_t is multiplied with z_t calculated above. This is to provide the input flowing into the memory cell.

   $$i_t \; = \sigma(W_i . x_t \; + R_i \, h_{t-1} \; + b_i \,)$$

3. Forget gate: The forget gate reads x_t and h_{t-1} and applies a sigmoid activation to weighted inputs. The result, f_t is multiplied by the cell state at previous time step c_{t-1} which allows for *forgetting* the memory contents which are no longer needed. The

output is 0 or 1, which 1 represents “completely keep this” while 0 represents “completely get rid of this.”

$$f_t = \sigma(W_f . x_t + R_f h_{t-1} + b_f)$$

4. Memory cell: The current cell state c_t is computed by forgetting irrelevant information from the previous time step and accepting relevant information from the current input.

$$c_t = z_t \odot i_t + c_{t-1} \odot f_t$$

5. Output gate: Output gate takes the weighted sum of x_t and h_{t-1} and applies sigmoid activation to control what information would flow out of the LSTM unit.

$$o_t = \sigma(W_o . x_t + R_o h_{t-1} + b_o)$$

6. Output: The output of the LSTM unit, h_t, is computed by passing the cell state c_t through a $tanh$ (to push the values to be between -1 and 1) and multiplying it with the output gate, o_t .

$$h_t = tanh(c_t) \odot o_t)$$

3.4 Loss Function

Neural network model receives the inputs and generates outputs after passing through the hidden layers. If the model prediction is closer to the desired output the model is more accurate and has better performance: A simple function which is called loss function will compute this error with comparing the predicted output with the actual output. This error (loss) has a direct relationship with the model accuracy. If the loss is small enough it means that the model prediction is close to the actual value so, the model performance is good and vice versa.

Different loss functions will give different error measurements for the same prediction. Thus have a considerable effect on the performance of the model. Some well-known loss function is described as following:

MAE (Mean Absolute Error):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|Y_i - \hat{Y}_i\right|\right)$$

MSE (Mean Squared Error):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

RMSE (Root Mean Squared Error):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}$$

MAPE (Mean Absolute Percentage Error):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Y_i - \hat{Y}_i\right|}{Y_i}.100\%$$

MRE (Mean Relative Error):

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Y_i - \hat{Y}_i\right|}{Y_i}$$

where n denotes number of samples, Y_i is actual load at time *t* and $\hat{Y}_i$ represents forecasted load at time *t*.

After calculating the loss, it is possible to make the loss as small as possible by using optimization algorithms such as Adam optimizer, Gradient Descent and AdaGrad [40]. The task of the optimizer is to update weights and biases of the internal parameters of a model to reduce the error and minimize the loss [25].

Chapter 4

4. Empirical Case

In this chapter the data and methods used to implement the ANNs models for peak load forecasting will be described.

4.1 Implementation of Deep Neural Networks Platform

Python language as the programming platform has adopted in this thesis. Several Python libraries were applied here, such as *numpy* which provides some advanced math functionalities, *Matplotlib* which is numerical plotting library as well as using Pandas to define *DataFrames* in order to make data manipulation easier.

The neural network model of this work is developed by using the *TensorFlow* deep learning platform in Python [26]. *TensorFlow* is an open-source software library was released in 2015 by Google for numerical computation and is nowadays widely used by many large companies [26]. In addition, *TensorFlow* makes it easier for developers to design, build, and train deep learning models.

4.2 Data Description

In this research, we have access to the prior load data during 4 months' period. This data includes household information which are key, power consumption and timestamp.

This data consists of 118 blocks of data, each of these blocks denotes one day, so in total information of 118 days (approximately 4 months) exists.

In this data-set, the time period is at 10 seconds intervals. Our work forecasts the peak load data for the next 10 minutes because peak load forecasting in minutes' scale is an essential subject. As it was mentioned in Section 1.1, the very short time interval is needed for real-time load forecasting.

In our methodology peak load of the previous day and days before are inputs of the ANNs model and the machine will forecast the peak load of the next day in 10 minutes' intervals as output. In the following we will explain the reason behind selecting these inputs. A lot of different factors affect the peak load such as seasonal change, geographic conditions, wind speed, weather conditions and etc. As an example, load consumption pattern in winter is hugely different with the one in summer. We only have access to the limited prior data with no data for example about wind or weather conditions in the four-month period. On the other hand, if we select peak load of only one day before, the forecasting will not be reliable as well because the previous day might be a holiday so the peak load on holidays and weekdays is not the same. As a result, we have selected peak load of two days before.

This thesis focuses on prior load data which are collected from the residential community. This data contains approximately four-month data starting from February 07th until June 1st 2017. Each row of data is related to the one house with its data set. Raw data of this house is used in Section 4.3.1 to fix duplications and missing data. Generally, different houses have different working patterns, so it does not make sense to forecast the load for all the houses by using the same methodology. In this thesis therefore we have presented the forecast model for one household, so it can be extended to others.

In reality we have limited amount of data and do not have access to the other houses data. Moreover, different houses have different consumption patterns. For example, an old couple lives in a house and a young couple with children live in another house; pattern in these two houses are very different. Old couple might be retired and most of the time of day would stay at home so the peak cannot be huge in dinner time, but in the young couple house with children, they would go out for work and school so the peak will be significant when they would come back home. Thus it can be concluded that it is not easy to find the same neural network model for all the houses. There is only one house information so we make one neural network model to train the data for that house. The other houses might have the same topology, but different training data, so training of the network will be different.

Our existing data is related to one house and includes several rows. Each row consists of a key, power consumption, and timestamp. we catch up all the rows of data in order to make time series data. In our case 118 block data exist. Since the data might have overlap with each other, it needs to be like time series data.

One new list is defined to avoid mixing up the data. Timestamp will be changed in time series because it is easier for a process. As an example, if last 24 hours' data is needed as input, we then need to convert hours to seconds by calculating 24h x 60min x 60sec.

4.3 Data Preprocessing

In neural network modeling, most studies and researches focus on the design and implementation of the model. Data preparation is in the secondary importance. However, preparing data is an important step in the neural network. The quality of the input data may have an enormous effect on the results of the neural network model. Precise preparation has serious effects such as reducing the model complexity and increasing the generalization. Also, inaccurate data preparation may reduce the model accuracy and cause the detrimental effect on the model result. In practice, data-sets are not completely neat; it can have any lack of data like missing, destroying or duplication. Thus data preparation contains data cleaning, normalization and data deviation for both training set and testing set [27].

Thereupon, this work has briefly described different forms of irregularities found in the raw data such as gap or duplications and the methods to handle them before generating the data-set for the machine in 4.3.1.

After processing, the data is ready to split into training and testing data-sets. In our MLP model, peak load for a day before as the first input and peak load for two days before as the second input are chosen. Then both input data are concatenated after reshaping and ready for feeding to the machine. This structure is applied for both training and testing of data-set. Figure (5) describes the black box schematic of the system.

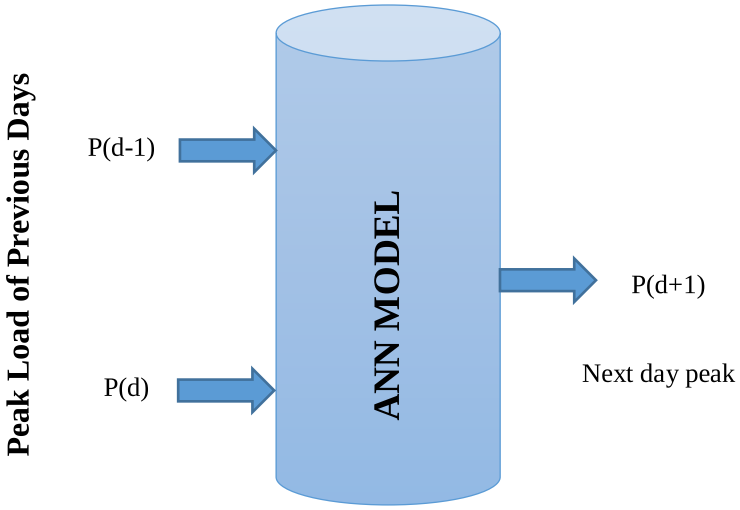

Figure 5: Black box modeling of used network.

4.3.1 Process Duplication, Missing and Gap

This sub chapter will focus on how to trim the data set. Our data also was not ready at the beginning. We experienced some disqualified data issues like errors in some data, missing of data and data duplication. After filling the gap between the missing data and removing all the duplication, data-set was prepared for use in the neural network model. This is an essential part of the thesis as the current raw data are not optimized for deep learning networks purposes. Then all the data has been put in two sets; training and testing data sets.

For filling the gap process, the iteration starts from the first row data-set to the last row with detecting the gap between consequent rows. If the gap is less than 15 seconds it is defined as normal gap and there is no need to do anything. We just add it to the data, whereas if it is equal to or greater than 15 seconds we then have to fill the gap by using the function which is already defined. This procedure is done for both training and testing data-set.

After data preparation, normalization is required. Normalization (or scaling) is one of the main parts of ANN learning process because by using the actual peak load directly in the machine the input and output data variables will have very different ranges. Therefore, normalization significantly helps as it scales the inputs to between 0 and 1. In order to calculate this the following equation is used:

$$z_i = \frac{x_i - \min(x)}{max(x) - \min(x)}$$

where x=(x_1,..., x_n) is denoted input and z_i is our i^{th} normalized data. $max(x)$ and $\min(x)$ are the maximum and minimum peak load data respectively.

Therefore, as it was mentioned in Section 4.3 the input data should be prepared in advance.

4.4. Model Details

In this research, peak load energy is predicted by MLPs and LSTM model for the next 10 minutes. For the neural network model to work appropriately, selection of proper historical load data sets and choosing the suitable activation function is necessary. For our models, the prior peak load data is used as input and the model predicts the next 10 minutes' load peak. Two inputs variables (which is mentioned in Section 4.2) with *tanh* activation function were selected for forecasting the output. The number of neurons for the hidden layers in MLP are picked until a better optimization algorithm is obtained. This will in details be described in the chapter (5).

4.4.1 Multilayer Perception Model (MLP)

In this study three MLP models have been applied; MLP models with two, three and four hidden layers respectively. In this section, MLP with two hidden layers will be explained in details and for the two other models the coding part is similar but just one and two hidden layers for MLP with three and four hidden layers is added respectively.

Our MLP network with two hidden layers is set with different epochs (iterations. The input layer contains two input neurons which are peak load data of the previous day and peak load of two days before. This input layer is connected to the first hidden layer with two neurons. These are forwarded to the second hidden layer with two neurons again and finally, they are forwarded to the output layer with one neuron. In this structure, the output of the first layer is multiplied to its specific weights. the result will then add to the bias. Next, this output passes to activation function which is a hyperbolic tangent function (*tanh*) here. In training part, the cost/loss function that is calculated by using MAPE will be passed to the applicable optimizer.

The code then generates the predicted and target value and print accuracy and error rate. All the measure like the number of hidden layers, neurons or the number of the epochs are obtained with trial and error method which will be discussed in the chapter (5).

The MLP with three and four hidden layers contain the same inputs and output, with just extra hidden layers added. In our study, MLP with different hidden layers is applied to compare the result of different models and observe effect of the number of hidden layers on the accuracy.

In MLP model, the number of neurons in the hidden layer is an important factor. Using less number of neurons in the hidden layers will cause underfitting [35]. Underfitting occurs when neurons in the hidden layers have too few neurons to detect the signals in a complicated data-set. Also using too many neurons in the hidden layers can result in several problems. One possible reason could be overfitting. Overfitting occurs when the neural network has too many neurons in hidden layers.

In reality, there is no clear and straight-forward way to find an optimized number of hidden layers or neurons in hidden layers. But there are some rule-of-thumb methods for determining the number of neurons to use in hidden layers such as the following given by [32];

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.
- (number of inputs + number of outputs) ^0.5 + (1 to 10). to fix the constant value (last part, 0 to 10), use trial and error and find the optimal no of hidden layer neurons for the minimum error.

According to the above rules, we picked 1,2 and 3 neurons and made the combination of them to figure out which one has better performance. The result of this combination is illustrated in chapter (5).

We test out different number of neurons in the hidden layers and analyze their performance in terms of MAPE for training data-set. We then select the network which has the least error. Figure (6) is the schematic drawing of our MLP model with 2 hidden layers.

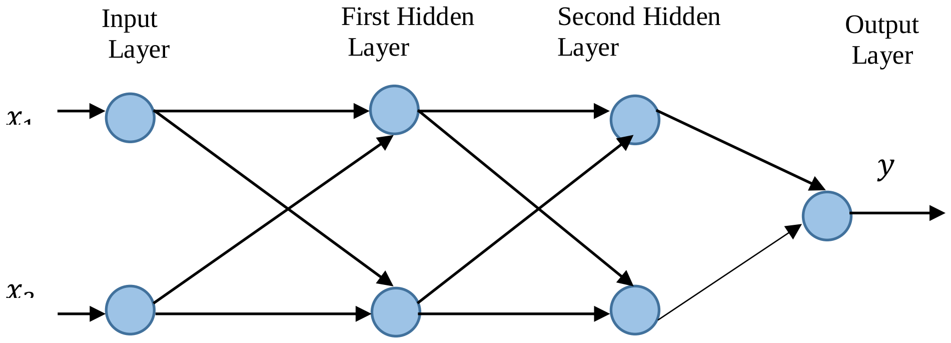

Figure 6: MLP model with 2 hidden layers.

4.4.2 Long Short Term Memory (LSTM)

After data preparation phase is completed, the prepared data will be scaled. First, the data is converted from 10 seconds intervals to the 10 minutes' intervals. Our LSTM model has 2 inputs and one output for both training and testing data-set.

In our LSTM model number of the epochs (iterations) are 100 and chunk size (number of inputs) is equal to two, number of classes (output) is one and the number of the LSTM cells are 40. It is worth mentioning that in order to choose the best and most accurate model all these numbers have obtained by trial and error method which is explained in the chapter (5).

In LSTM both the input and output data are originally in horizontal format whereas our input and output data are in vertical format. on the other hand, our input data are like a matrix with two rows and n columns, and output data are in one rows with n column. Therefore, it is compulsory here to reshape the input and output data and change it to the input matrix with n rows and two columns and output matrix with n row and one column. This Reshaping process should apply for both training and testing data.

In the training part of the model MAPE (loss function) is applied in order to figure out which gives the result with the least error. The next step is to select the appropriate optimizer in order to minimize the cost. Finally, the accuracy and error rate is calculated. Figure (6) shows the LSTM cell.

Chapter 5

5. Model Performance and Discussions

In this chapter, we will go through and analyze the results from each experiment that was done based on the two techniques; LSTM and MLP. Both of the techniques have been already discussed in this thesis.

In this research, we have access to the prior load data. This data consists of one house information in 118 block data. Each of these blocks corresponds to one day, hence in total 118 days of information (approximately 4 months) are available. Each block includes key, power consumption, and timestamp data. The MLP and LSTM models are applied for predicting the peak load power for the next 10 minutes' intervals. MAPE is used as a performance criterion and it has been calculated for all ANN models.

In order to give a general overview, Figure (7) is plotted to indicates the variation of peak load consumption in one month, which starts from February 07th until March 07th 2017. In Figure (7), the x axis is time in terms of day that shows the one-month period and y axis denote the peak power consumption. In reality our data are in in 10 minutes' intervals but, if we plot the time in 10 minutes' scale the graph will be chaotic with too many data-points that is not clear to analyze. That is why we have chosen to put time in terms of the day.

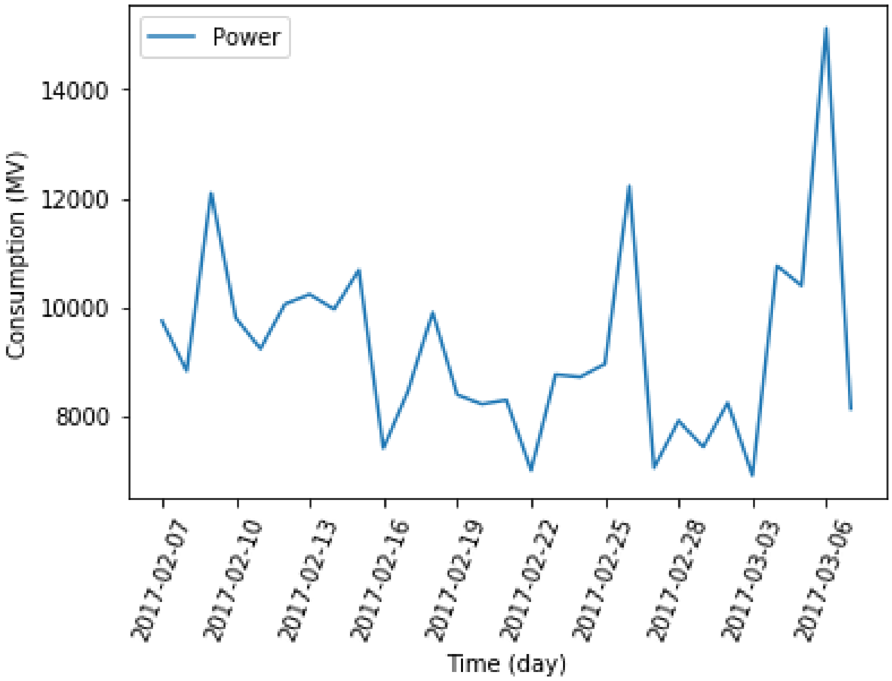

Figure 7: Variation of peak load in a month.

.

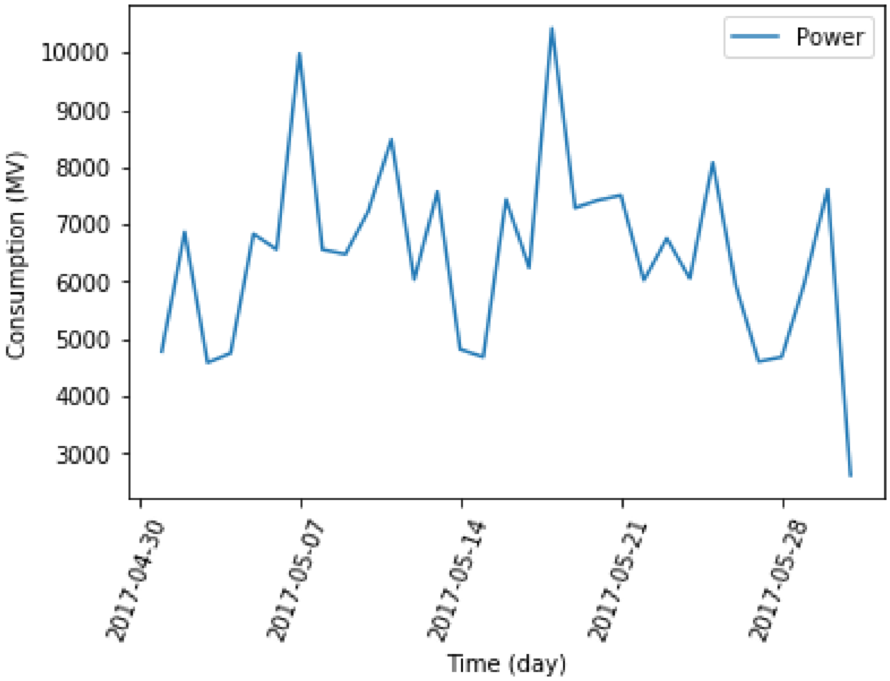

Figure 8:Variation of peak load in testing data.

Figure (8) describes the prediction targets in testing data-set, which starts from May 1st until May 31st 2017.

As it was mentioned in Section 4.4.1, by using rule-of-thumb [32], we have chosen 1, 2 and 3 neurons for the number of hidden layers. Table (1), (2) and (3) below list and compare the performance of the MLP models by using combination of different number of hidden layer neurons.

Table (1), shows the result of MLP (2-2-1) model; 2 input neurons, 2 hidden layers and one output.

Table 1: MLP (2-2-1) with different neurons in hidden layers.

Number of Neurons at Hidden Layers	**MAPE**	**Number of Neurons at Hidden Layers**	**MAPE**
1-1	18.7	2-3	21.8
1-2	22.0	3-1	21.3
1-3	18.6	3-2	20.4
2-1	18.0	3-3	75.9
2-2	**17.3**		

As it has been shown in table above, the MLP (2-2-1) with 2 neurons in first hidden layer and 2 neurons in second hidden layer achieve the minimum MAPE thus it is the best choice.

We then repeat the same procedure for MLP (2-3-1) and MLP (2-4-1), the results of those has are in Table (2) and Table (3) respectively.

Table (2) describes the performance MLP (2-3-1) with 2 neurons in input layer, 3 hidden layers and one output.

Table 2: MLP(2-3-1) with different neurons in hidden layers.

Number of Neurons at Hidden Layers	**MAPE (%)**	**Number of Neurons at Hidden Layers**	**MAPE (%)**
1-1-1	38.2	2-2-3	18.6
1-1-2	19.7	2-3-1	20.2

1-1-3	17.0	2-3-2	56.7
1-2-1	18.5	2-3-3	33.2
1-2-2	18.7	3-1-1	18.5
1-2-3	19.0	3-1-2	51.3
1-3-1	18.3	3-1-3	24.2
1-3-2	17.7	3-2-1	18.3
1-3-3	28.0	3-2-2	19.8
2-1-1	18.0	3-2-3	16.6
2-1-2	**16.1**	3-3-1	19.2
2-1-3	17.5	3-3-2	18.7
2-2-1	18.9	3-3-3	18.9
2-2-2	18.6		

According to the Table (2), the MLP (2-1-2) with 2, 1 and 2 neurons respectively in first, second and third hidden layers has the smallest MAPE, which is the best one.

Table (3) compares the performance of MLPs with (2-4-1) result with different neurons in hidden layers.

Table 3: MLP (2-4-1) with different neurons in hidden layers.

Number of Neurons at Hidden Layers	**MAPE (%)**	**Number of Neurons at Hidden Layers**	**MAPE (%)**	**Number of Neurons at Hidden Layers**	**MAPE (%)**
1-1-1-1	20.21	3-3-1-1	19.33	1-2-3-1	26.43
1-1-1-2	19.92	3-3-1-2	18.55	1-2-3-2	17.65
1-1-1-3	18.93	3-3-1-3	18.67	1-2-3-3	19.13
1-2-1-1	18.61	1-1-3-1	18.72	1-3-2-1	18.54
1-2-1-3	19.03	1-1-3-2	23.09	1-3-2-2	21.75
1-2-1-2	18.84	1-1-3-3	18.96	1-3-2-3	22.54
2-1-1-1	20.01	1-3-1-1	18.65	3-2-1-1	20.22
2-1-1-2	23.03	1-3-1-2	18.80	3-2-1-2	20.67

2-1-1-3	22.26	1-3-1-3	18.93	3-2-1-3	16.58
1-1-2-1	18.74	3-1-1-1	18.46	3-1-2-1	18.62
1-1-2-2	18.81	3-1-1-2	17.22	3-1-2-2	18.86
1-1-2-3	19.97	3-1-1-3	21.68	3-1-2-3	21.32
2-2-1-1	18.59	2-2-3-1	31.9	2-1-3-1	27.48
2-2-1-2	18.92	2-2-3-2	18.76	2-1-3-2	19.34
2-2-1-3	36.51	2-2-3-3	28.44	2-1-3-3	18.83
2-1-2-1	18.62	2-3-2-1	20.20	2-3-1-1	21.31
2-1-2-2	18.25	2-3-2-2	22.60	2-3-1-2	19.70
2-1-2-3	19.05	2-3-2-3	23.23	2-3-1-3	17.86
1-2-2-1	18.61	3-2-2-1	47.33	3-3-2-1	21.51
1-2-2-2	**16.49**	3-2-2-2	18.22	3-3-2-2	35.90
1-2-2-3	19.99	3-2-2-3	18.81	3-3-2-3	21.49
1-3-3-1	21.91	3-3-3-1	18.12	2-3-3-1	19.56
1-3-3-2	19.36	3-3-3-2	20.49	2-3-3-2	25.71
1-3-3-3	19.44	3-3-3-3	22.14	2-3-3-3	17.64
3-1-3-1	18.65	2-2-2-2	16.63	3-2-3-1	19.39
3-1-3-2	22.37	2-2-2-1	18.64	3-2-3-2	18.90

Table above shows the in MLPs with (2-4-1) structure, the one with 1 neuron in first hidden layer and 2, 2 and 2 neurons in second, third and fourth hidden layers respectively attains the best performance.

As can be seen in tables above, MLP with 2 hidden layers is more accurate compared with the two other structures as it has the lowest error. It is observed that there is a tendency of performance attenuation, the reason could be overfitting. To avoid overfitting, one way is to limit the number of hidden layers [33].

Then, LSTM model with different number of LSTM cells are evaluated and the obtained results are tabulated in Table (4).

Table 4: LSTM with different number of cells.

Number of Cells	MAPE (%)
1	39.7
5	26.2
10	28.9
15	20.2
20	17.5
30	05.6
40	**04.4**
50	06.2
60	07.5
70	07.7

based on Table (4) above, LSTM model with 40 cells has the smallest MAPE, which is the satisfactory model. After increasing the number of cells to more than 40, the MAPE also increases. As a result, we stopped the testing at 40 cells. Moreover, three optimizers are applied to figure out which one has better result. In Table (5) and (6) result of three optimizers are compared based on MAPE;

Table 5: Different Optimizers Performance for MLP model.

ANN structure	Number of Neurons at Hidden Layers	Gradient Descent	Adam	Adagrad
MLP (2-2-1)	2-2	17.3	31.7	21.5
MLP (2-2-1)	2-1-2	16.1	36.4	18.2
MLP (2-2-1)	1-2-2-2	16.4	38.2	18.0

Table 6:Different Optimizers Performance for LSTM model.

ANN structure	Number of cell	Gradient Descent	Adam	Adagrad
LSTM	40	4.41	36.3	13.6

By empirical result in Table (5) and (6), the Gradient Descent is the best choice for our ANN model, because as we see it has minimum MAPE.

After the proper model structures based on result of Tables (1), (2), (3) and (4) are selected, it was tested with different epochs. The same procedure is repeated for all selected models. The results are tabulated in Tables (7), (8) and (9) as following:

Table 7: Performance of MLP (2-2-1) models with different number of epochs.

Number of Epochs	**MAPE (%)**
10	21.8
20	20.6
50	16.5
100	**15.8**
120	16.9
150	18.5
200	18.7
220	19.3

Table 8: Performance of MLP (2-3-1) models with different number of epochs.

Number of Epochs	**MAPE (%)**
10	21.3
20	19.4
50	17.7
100	**16.6**
120	18.4
150	18.9
200	19.6
220	21.1

Table 9: Performance of MLP (2-4-1) models with different number of epochs.

Number of Epochs	MAPE (%)
10	19.3
20	18.5
50	18.0
100	**17.0**
120	17.5
150	18.5
200	19.0
220	19.5

Based on Table (7), (8) and (9), MLP with 100 epochs has the smallest error, so 100 epochs will be our best choice for the MLP models.

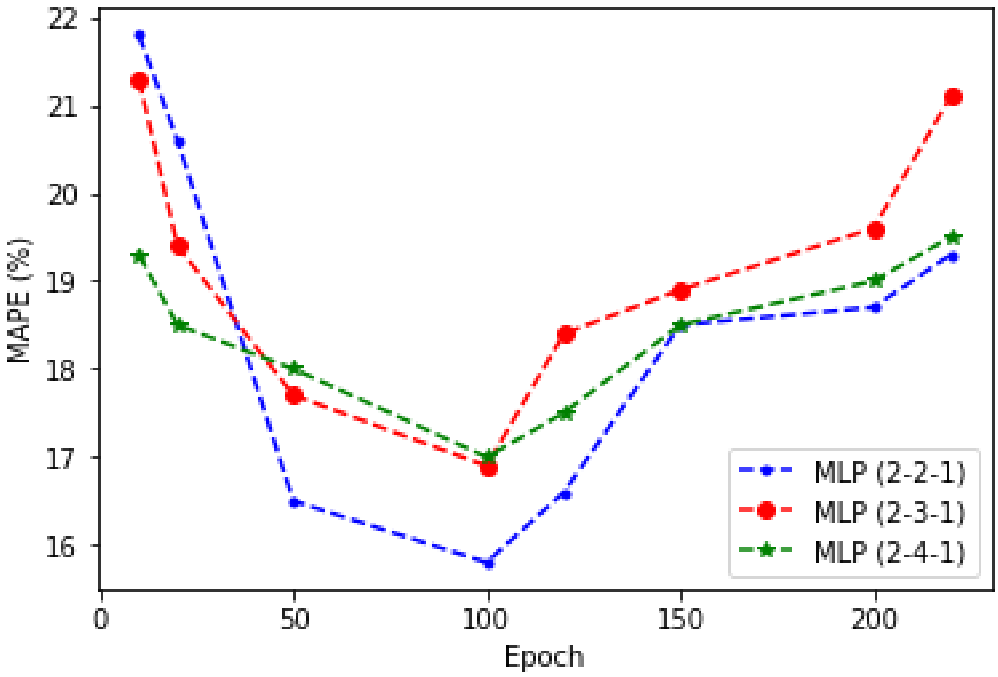

Figure 9: MLP models performance.

It is able to observed that, the MLP models according to number of epochs and error rate are compared in Figure (9) this figure MLP model with 2 hidden layers has the minimum error amongst all. Therefore, we have chosen MLP (2-2-1) as the best MLP model. Table (10) shows the LSTM performance base on the number of epochs.

According to Table (10), the error decreases in LSTM model by increasing the number of epochs until 200 epochs. Thereafter by increasing the epochs above 100 tendency of performance attenuation increases. Thus we do not increase the epochs anymore. In above table the best LSTM model is the one with 100 epochs as it has the minimum error. Figure (10) in the next page illustrates the LSTM performance.

Table 10: LSTM models with different number of epochs.

Number of Epochs	**MAPE (%)**
10	17.9
20	11.3
50	1.70
100	**1.27**
120	1.35
150	1.56
200	1.89
220	8.42

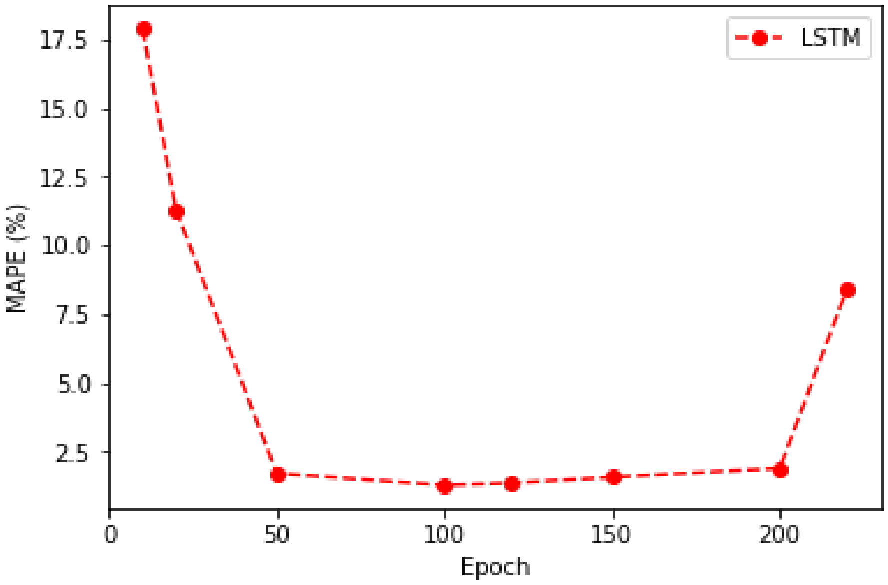

Figure 10: LSTM model performance

We now compare the best MLP model which is MLP (2-2-1) and LSTM with 100 epochs. The result has been shown in Figure (11).

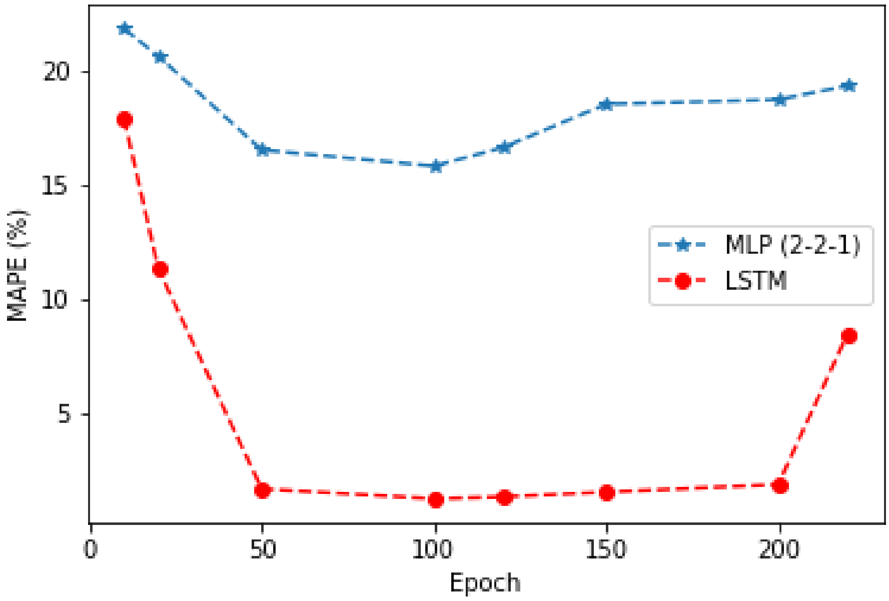

Figure 11: MLP & LSTM performance comparison.

According to the Figure (11), LSTM performs better than MLP in error. Therefore, based on this study we can conclude LSTM is more accurate than MLP.

In Table (11) comparison of the MAPE between the LSTM and some methods utilized in other related works are illustrated;

Table 11: Comparison of the MAPE (%) between this work and some other studies.

References	**MAPE%**
Our LSTM	1.31
[9]	1.57
[28]	2.43
[29]	2.02
[30]	1.74
[31]	4.04

The references that are presented in Table (11), have included MLPs and other models with 8 inputs and one output. They have considered 7 inputs as past peak loads as a week contains 7 days. The one other input left represents the characteristic of forecasting day. This one more input is added because of the different nature of days in a week (holiday and week days). We made our LSTM model with 8 inputs (peak load of 7 days before and 1 characteristic parameter) and 1 output, exactly same as them. For this form of LSTM we got 1.31 MAPE% which means better performance. The remarkable reason for this result is LSTM model has memory cell and stores the last output in memory and use that as input for next successive step. Both MLP and LSTM networks can have same number of inputs and outputs. However, due to memory feature of the nodes in LSTM, LSTM could perform better than MLP as it was proven in chapter 5 it performed better

Chapter 6

6. Conclusions and Future work

6.1 Conclusion

The main issue of this thesis have been to perform an analysis of the provided input data and to develop the qualitative forecasting models for them. In order to solve this problem, the data preprocessing, forecasting methods and other essential aspects of time series analysis has been discussed in the theoretical part of the thesis. It was investigated, that very often, the proper data preprocessing plays the key role of the whole process.

Due to the prediction results, the LSTM model in our problem has minimum MAPE, hence is the best structure for peak load forecasting in this research.

The results indicate that the LSTM model has advantages in time series analysis in comparison to other Neural Networks like MLP, it might because of LSTM structure. LSTM stores the last output in memory because it has memory cell and uses that as input for next successive step. In the other hand LSTM performs better than other MLP models when evaluated against MAPE value, so in this case LSTM network was a good choice in peak load forecasting.

6.2 Future Work

As this approach only use the peak load as an input to the artificial neural network and doesn't take into account other relevant parameters like atmospheric temperature, humidity, rain and pressure, thus, there is a potential research to be conducted highlighting the effect of these parameters on load forecasting systems in future. Moreover, our data is included load data in limited several months, in future if whole year data or more will be available then we can forecast future load by adding season feature and analyze the peak in different seasons

References

1. H. G. Stoll, Least-cost electric utility planning, New York: John Wiley Sons Inc; 1989.

2. Stefan Hosein and Patrick Hosein, “Load Forecasting using Deep Neural Networks,” IEEE 2017.

3. E. Busseti, I. Osband, and S. Wong, “Deep learning for time series modeling,” Stanford, Tech. Rep., 2012

4. J. Connor,L.E.Atlas,andD.R.Martin,“Recurrent networks and NARMA modeling,” in *Advances in Neural .*

5. J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015.

6.T. Hossen, S.J. Plathottam, R. K. Angamuthu, P. Ranganathan and H. Salehfar, “Short-Term Load Forecasting Using Deep Neural Networks (DNN),” in Department of Electrical Engineering, 2018.

7. H. Quan, D. Srinivasan and A. Khosravi, “Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals,” IEEE 2013

8. G. Muhammad Khan,Shahid Khan and Fahad Ullah, “Short-Term Daily Peak Load Forecasting Using Fast Learning Neural Network, ” *Department of Electrical Engineering,IEEE 2011*

9. Moh. Ghomi, M. Goodarzi and Mahmood Goodarzi, “Peak Load Forecasting of electric Utilities for West province of IRAN by using Neural Network without Weather information,”

Electrical engineering department 2010

10. Yu Wang, “A New Concept using LSTM Neural Networks for Dynamic System Identification,” 2017 American Control Conference

11. U. Basaran Filik, M.Kurban ,“A New Approach for the Short-Term Load Forecasting with Autoregressive and Artificial Neural Network Models,” International Journal of Computational Intelligence Research, ISSN 0973-1873 Vol.3, No.1 (2007), pp. 66–71.

12. Dr. M. Aly Aboul-Magd, Emad El-Din El-Sayed Ahmed “AN ARTIFICIAL NEURAL NETWORK MODEL FOR ELECTRICAL DAILY PEAK LOAD FORECASTING WITH AN ADJUSTMENT FOR HOLIDAYS,” IEEE 2001.

13. Tatsuya Iizaka,Tetsuro Matsui and Yoshikazu Fukuyama,“ A Novel Daily Peak Load Forecasting Method using Analyzable Structured Neural Network,” IEEE 2002.

14. P. Xinxing ve B. Lee, «A Comparison of Support Vector Machines and Artificial Neural Networks for Mid-Term Load Forecasting, » 2012 IEEE International Conference on Industrial Technology (ICIT), pp. 95-101, 2012.

15. A. Laouafi, M. Mordjaoui ve F. Laouafi, «An Evaluation of Conventional and Computational Intelligence Methods for Medium and Long-Term Load Forecasting in Algeria,» 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT), pp. 1-6, 2015.

16.Draidi Abdellah, Labed Djamel “Forecasting the Algerian Load Peak Profile Using Time Series Model Based On Backpropagation Neural Networks,” IEEE 2013.

17. Mohan B. Tasre, Prashant P. Bedekar, and Vilas N. Ghate “Daily Peak Load Forecasting Using ANN,” IEEE 2011

18. Santisouk Phimphachan, Kosin Chamnonglhai, Pinit Kumhom, Nutthacha Jittiwar[angkul and Anawach Sangswang “Energy and Peak Load Forecast Models Using Neural Network for Fast Developing Area,” Interntional Symposium on Communicationsand Information Technologies 2004 (ISCIT 2004) Sapporo. Japan, October 26- 29,2004.

19. Nathaniel J. Hobbs, Byoung H. Kim, Kwang Y. Lee, “Long-Term Load Forecasting Using System Type Neural Network Architecture,” Electrical and Computer Engineering Department, Baylor University, Waco, Texas 76798 USA (phone: (254) 710-4195, e-mail: Kwang_Y_Lee@baylor.edu).

20. S. R. Khuntia, J. L. Rueda ve M. A. van der Meijden, «Forecasting the load of electrical power systems in mid- and long-term horizons: a review,» IET Generation, Transmission & Distribution, cilt 10, no. 16, pp. 3971-3977, 2016.

21. Tareq Hossen, Siby Jose Plathottam, Radha Krishnan Angamuthu, Prakash Ranganathan and Hossein Salehfar “Short-Term Load Forecasting Using Deep Neural Networks (DNN),” Department of Electrical Engineering.

22. Gor Aizenberg, Naum N. Aizenberg, Joos P.L. Vandewalle (2000). Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications. Springer Science & Business Media.

23. EA. Feilat1*, D. Talal Al-Sha’abi2 and MA. Momani, “Long-term load forecasting using neural network approach for Jordan's power system,” Eng Press. 2017; 1(2): 43-50. doi: 10.28964/ EngPress-1-108.

24. Leshno, M.; Lin, V.Y.; Pinkus, A.; Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.* 1993, *6*, 861–867.

25. Loss function, backpropagtion. 2017. url: https: / / enwikipedia. org/wiki/Backpropagation.

26. https://www.tensorflow.org/

27. https://link.springer.com/chapter/10.1007%2F978-0-387-71720-3_3

28. Lalit Mohan Saini, Peak load forecasting using Bayesian regularization, Resilient and adaptive backpropagation learning based artificial neural networks, Electric Power Systems Research 78 (2008) 1302–1310

29. N. Amjady, F.Keynia, Short-term load forecasting of power systems by combination of wavelet transform and neuro- evolutionary algorithm , Energy 34 (2009) 46–57

30. E. Gonzalez-Romera, M.A. Jaramillo-Moran, D. Carmona- Fernandez, Monthly electric energy demand forecasting with neural networks and Fourier series, Energy Conversion and Management 49 (2008) 3135–3142

31. Y ouchan Zhu, Y ujun He, Short-term Load Forecasting Model Using Fuzzy C Means Based Radial Basis Function Network,, Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06) 0-7695-2528-8/06 © 2006.

32. Foram S. Panchal1, Mahesh Panchal2, Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network, IJCSMC, Vol. 3, Issue. 11, November 2014, pg.455 – 464

33. Khalil el Hindi, Mousa AL-Akhras, "SMOOTHING DECISION BOUNDARIES TO AVOID OVERFITTING IN NEURAL NETWORK TRAINING," December 20, 2010

34. Xiangang Li, Xihong Wu (2015). Constructing Long Short-Term Memory based Deep Recurrent Neural Net- works for Large Vocabulary Speech Recognition arXiv: 1410.4281

35. Foram S. Pancha, Mahesh Panchal, "Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network," International Journal of Computer Science and Mobile Computing, Vol.3 Issue.11, November- 2014, pg. 455-464.

36. Chi-Cheng Chuang, Jimi Y. C. Wen, Ray-I Chang, "Consumer Energy Management System: Contract Optimization using Forecasted Demand," 2011, ISBN: 978-1-61208-136-6.

37. Nolin Rural Electr. Cooperative Corp.,"Short-term load forecasting using multiple regression analysis," IEEE 06 August 2002.

38. WR.Christinaanse, ,"Short-Term Load Forecasting Using General Exponential Smoothing," IEEE March 1971.

39. D.K.Ranaweera, N.F.Hubele, G.G.Karady "Fuzzy logic for short term load forecasting," International Journal of Electrical Power & Energy Systems ,May 1996, Pages 215-222.

40. Aylien Ltd., Dublin, "An overview of gradient descent optimization algorithms," 19 January 2016.