On Short-Term Load Forecasting Using Machine Learning Techniques

Behnam Farsi

A Thesis
in
The Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science
(Quality Systems Engineering) at
Concordia University
Montréal, Québec, Canada

December 2020

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Behnam Farsi**
Entitled: **On Short-Term Load Forecasting Using Machine Learning Techniques**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science
(Quality System Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Walter Lucia	_____	Chair
Dr. Zachary Patterson	_____	External Examiner
Dr. Suryadipta Majumdar	_____	Internal Examiner
Dr. Ursula Eicker	_____	Supervisor
Dr. Manar Amayri	_____	Supervisor
Dr. Nizar Bouguila	_____	Supervisor

Approved _____

Dr. Mohammad Mannan Graduate Program Director

Dec 2020 _____

Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science
(Gina Cody School of Engineering and Computer Science)

Abstract

On Short-Term Load Forecasting Using Machine Learning Techniques

Behnam Farsi

Since electricity plays a crucial role in industrial infrastructures of countries, power companies are trying to monitor and control infrastructures to improve energy management, scheduling and develop efficiency plans. Smart Grids are an example of critical infrastructure which can lead to huge advantages such as providing higher resilience and reducing maintenance cost. Due to the nonlinear nature of electric load data there are high levels of uncertainties in predicting future load. Accurate forecasting is a critical task for stable and efficient energy supply, where load and supply are matched. However, this non-linear nature of loads presents significant challenges for forecasting. Many studies have been carried out on different algorithms for electricity load forecasting including; Deep Neural Networks, Regression-based methods, ARIMA and seasonal ARIMA (SARIMA) which among the most popular ones. This thesis discusses various algorithms analyze their performance for short-term load forecasting. In addition, a new hybrid deep learning model which combines long short-term memory (LSTM) and a convolutional neural network (CNN) has been proposed to carry out load forecasting without using any exogenous variables. The difference between our proposed model and previously hybrid CNN-LSTM models is that in those models, CNN is usually used to extract features while our proposed model focuses on the existing connection between LSTM and CNN. This methodology helps to increase the model's accuracy since the trend analysis and feature extraction process are accomplished, respectively, and they have no effect on each other during these processes. Two real-world data sets, namely "hourly load consumption of Malaysia" as well as "daily power electric consumption of Germany", are used to test and compare the presented models. To evaluate the performance of the tested models, root mean squared error (RMSE), mean absolute percentage error (MAPE) and R-squared were used. The results show that deep neural networks models are good candidates for being used as short-term prediction tools. Moreover, the proposed model improved the accuracy from 83.17% for LSTM to 91.18% for the German data. Likewise, the proposed model's accuracy in Malaysian case is 98.23% which is

an excellent result in load forecasting. In total, this thesis is divided into two parts, first part tries to find the best technique for short-term load forecasting, and then in second part the performance of the best technique is discussed. Since the proposed model has the best performance in the first part, this model is challenged to predict the load data of next day, next two days and next 10 days of Malaysian data set as well as next 7 days, next 10 days and next 30 days of German data set. The results show that the proposed model also has performed well where the accuracy of 10 days ahead of Malaysian data is 94.16% and 30 days ahead of German data is 82.19%. Since both German and Malaysian data sets are highly aggregated data, a data set from a research building in France is used to challenge the proposed model's performance. The average accuracy from the French experiment is almost 77% which is reasonable for such a complex data without using any auxiliary variables. However, as Malaysian data and French data includes hourly weather data, the performance of the model after adding weather is evaluated to compare them before using weather data. Results show that weather data can have a positive influence on the model. These results show the strength of the proposed model and how much it is stable in front of some challenging tasks such as forecasting in different time horizons using two different data sets and working with complex data.

Acknowledgements

I would like to express my very profound gratitude to my supervisors Prof. Nizar Bouguila and Prof. Ursula Eicker. I joined Concordia University in September 2019 to pursue my graduate studies in Master of Engineering in the Department of Electrical Engineering, and then in January 2020, Prof. Bouguila gave me the chance to join his research team as a machine learning researcher. After starting working on load forecasting, I got this opportunity to join Dr. Eicker's team too, which was a turning point in my life. My supervisors trusted me, helped me to never lose my hope about the future, and never let me give up in hard situations during pandemic. As a result of their kind endless supports, patience, encouragement and wise guidance, now I am able to follow my dreams and try hard to make the world a better place. I will be forever grateful to them for giving me the opportunity to work under their guidance.

I would like to thank my co-supervisor Dr. Manar Amayri who supported me during my Master's studies. Dr. Amayri is one of the kindest people I have ever met in my life and I learned a lot from her in the area of load forecasting and energy.

I intend to offer my special thanks to Ms. Narges Maouchehri. She was like a sister to me at Concordia University. While Narges and I were working in two different areas, she helped me in my research. She had spent a lot of time for me to teach me how I can become a successful person in my life and how I can improve my skills. She is smart, kind and trustful and I hope to find the chance to continue working with Narges.

It is noteworthy to mention my gratitude to Concordia University that let me join this university and supported international students during pandemic. What Concordia University did during pandemic was epic and I can not forget their kindness.

Last but not the least, I would like to thank my family as the most valuable gift of my life because of their unconditional love, respect and trust. Likewise, I can not explain how much I am grateful because of having Mr. Amirhossein Zamiri and Mr. Mehrdad Khosravi as my best friends. Leaving these two best friends and coming to Canada was the hardest challenge I've ever had. My family and my friends always encourage me to achieve my goals, realize my dreams and fly on my own which let me have the confidence to discover new worlds. I am truly blessed to have you. Thank you for everything you've ever done for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction and Related Work

According to the IEA report [24], in 2017, world electricity consumption reached 21,372 TWh, which is 2.6% higher than 2016 electricity consumption. Such an annual increase creates a new problem: how to reduce the consumption? Nowadays, many companies are working on this problem and trying to solve it. Demand Response Management, which is one of the main features in smart grids [2], helps to control the electricity consumption with the focus on the customer side. It is important to understand residential and non-residential building demand and use of electricity. It is obvious that carrying out a reduction in load consumption can lead to a number of economic and environmental benefits. Load forecasting methods can provide an alternative solution to electricity network augmentation as it can be useful to manage the electricity demand and provide more energy efficiency [3]. In addition, improving power delivery quality along with having secure networks is an important task in smart grids in order to monitor and support advanced power distribution systems [1] and in particular to improve load forecasting. Since future consumption could be predicted, they can be considered as tools to minimize the gap between electricity supply and user consumption. However, an inaccurate prediction may lead to huge loss. For instance, a small percentage of increase in forecast error was predicted in 1985, which led to more than 10 million pounds of yearly detriment in the UK thermal power systems [4]. Thus, many big companies have focused on accurate load forecasting and load managements. The Energy Supply Association of Australia

as an instance, invested about 80% of its budget on grid upgrade.

Load forecasting approaches are categorized in four different groups with respect to their functionalities for different purposes [5, 20]: Very Short-term Load Forecasting (VSTF), Short-term Load Forecasting (STLF) [19], Medium-term Load Forecasting (MTLF) and Long-term Load Forecasting (LTLF) [18]. VSTLF aims to forecast the next minutes of load consumption, STLF forecasts the following hour load to next week, while in MTLF this prediction time is more than one week to a few months and LTLF forecasts the next years load consumption. For each of these methods, there are diverse factors which influence the prediction. Due to the ability of STLF approaches, they have remarkable importance in energy management. Hence, they have been used to provide proper management in electric equipments and because of this contribution, they are known as an inevitable component in Energy Management Systems. An error in STLF can have immediate impact on electrical equipment. Several factors affect the STLF, including the following ones: (1) Time factor [5], which is the most important factor for STLF because of existence of some patterns such as daily patterns in a set of data (2) Climate, which contains temperature, humidity and some special evidences is playing an important role in load forecasting [5]. (3) Holidays can make huge changes in electricity demand.

The aforementioned factors can affect the MTLF and LTLF too. The authors in [7] have done a detailed study on LTLF using multiple linear regression (MLR). They discussed the effect of various factors including temperature, holidays, weekdays and weekend on LTLF application. However, due to its importance and less dependency on various variables, most of the researchers have focused on STLF. Since power companies need to control everyday power system operations, STLF is an essential tool for them to have an approximation of ranging everyday electricity usage. However,they are both delicate processes and any fault may bring huge additional costs for user [6]. Due to this reason and the importance of accurate load forecasted data, this thesis studies load forecasting with a major focus on STLF tasks.

As it mentioned before, many studies have tackled load forecasting with various methods. The authors in [8] reviewed different regression based methods [17] for STLF. In another study [9], the author investigated different MLR for load forecasting. The problem within these regression based methods is that they need some external variables such as weather, wind speed, etc to achieve accurate results. Auto-regressive models are another well-known methodology which are able to be used for load forecasting purposes. Among

these models, autoregressive integrated moving average (ARIMA) is highly used and it has been able to provide acceptable prediction results, for example the author in [10] has used ARIMA and Box-Jenkins methodology to carry out an hourly prediction. As with regression based methods, auto regressive models come with number of disadvantages in which complex computations and high computation time are the most important problems. Concerning the meaning of complex computation, it means that to predict future load data some mathematical calculation should be done to find out the parameters of formulas. Because of the existing problems within statistical and mathematical models, many researchers have started using deep learning [21,22] in load forecasting applications. In [11] the authors studied 7 different models on 3 real-world data sets and they showed that these deep learning methods have enough potential to be used in load forecasting applications instead of some mathematical techniques like ARIMA. Since the world is developing, innovation in each research field is significant and because of that in some works such as [12], the authors proposed a new parallel model which is a combination of convolutional neural network (CNN) and recurrent neural network (RNN). As RNNs use control theory in their structure, they are able to find the dependency between old data and new ones and they have become an interesting network for load forecasting applications in recent year. Regarding how RNNs work, [14] has done a suitable study on these networks. In another study close to [12], the authors in [13] proposed an hybrid of long short-term memory (LSTM) and CNN. The proposed model's results proved that it can have more stable performance in load forecasting compared to other learning machines. Likewise, the authors in [15] proposed a new DeepEnergy model which is a combination of 1-D CNN to extract the features and fully connected network to forecast future load data. To forecast the next 3 days data, they used an hourly electricity consumption data set from USA. To train the data previous 7 days was used. They compared the proposed model's result with 5 other machine learning technique through RMSE and MAPE. The results showed that the DeepEnergy model has more ability to carry out an accurate short-term load forecasting compare to other models. After DeepEnergy model, Random Forest technique [16] had done a reasonable performance. In another study [59], the authors proposed a new model which consists of three algorithms including Variational Mode Decomposition (VMD), Convolutional Neural Network (CNN) and Gated Neural Network (GRU), and for more convenience they called the proposed model SEPNet. This model aimed to predict hourly electric price and in order to evaluate the

model, an hourly data from city Newyork, USA which includes the hourly electricity price from 2015 to 2018. In terms of the model performance, compared to other models such as LSTM, CNN, VMD-CNN, the SEPNet model performed better where it improved the RMSE and MAPE 25% and 19%, respectively. Also some authors such as [60] used ANNs to forecast other type of load data like PV system output data. They proposed a powerful CNN based model called PVPNet and they evaluated the proposed model through using a daily data from 2015. They used 3 past days information to predict next 24h. Regarding the mean absolute error (MAE) and RMSE, the model has performed Random Forest (RF). However, with technology development, many studies deployed machine learning models in IOT. In terms of technical part, if these models are supposed to being used in IOT they must be able to perform online load forecasting. [61] presents some related machine learning methods which can be used in IOT through cloud. They also implemented a novel hardware technology including Arduino microcontroller. They implemented the device in a research lab to predict total power consumption in lab. Regarding the algorithms, Linear Regression, SVM Regression, Ensemble Bagged, Ensemble Boosted, Fine Tree Regression and Gaussian Process Regression (GPR) have been used. All of the mentioned models have performed appropriately. All of these studies have been carried out to achieve more accurate results in load forecasting, but which one is able to perform better and are the models powerful enough to predict accurately for all load data sets?

1.2 Contributions

Even though some studies in recent years have discussed different models for short-term load forecasting [8, 23], the lack of a comprehensive study to carry out a comparison between classic time series models, regression based models and deep learning is obvious. In addition, time series must be used correctly as input for machine learning models. In other words, some analysis on data are essential to compile machine learning models. The contribution of this thesis is that we focus on different models which are appropriate to be used for load forecasting, and we also review some different methodologies to find out the most effective models for forecasting applications. Even though various deep learning models have been introduced for load forecasting in recent years, only some of them have succeeded to achieve state of

the art results. Moreover, this thesis consecutively proposes a new hybrid parallel CNN-LSTM and LSTM-Dense neural networks to improve the accuracy of load forecasting. Regarding the model's architecture, it consists of two different paths (CNN and LSTM). CNN path extracts the input data features and LSTM path learns the long-term dependency within input data. After passing through these 2 paths and merging their outputs, a fully connected path combined with a LSTM layer has been implemented to process the output to predict final load data. This thesis firstly aims to evaluate various machine learning performance in STLF task while there is no exogenous variables available. In other words, it tries to find out a way to carry out STLF using just previous load data and compare all the results with each other. After finding the best model for STLF task, the model will be evaluated in different challenges including dealing with more complex data, prediction in different time horizons and adding exogenous data as input. In order to extend our study, all the models are implemented to forecast daily and hourly ahead load consumption with using two highly aggregated data sets, one of which is an hourly power consumption from the city of Johor in Malaysia [25] and the other one is a daily electric consumption which is collected from a power supply company in Germany. In addtion, a one-year hourly load and weather data from a research building in France is used to challenge the proposed model. To evaluate our models, we use root mean squared error (RMSE) due to the ability of showing how much predicted values spread around average and mean absolute percentage error (MAPE) as it is able to present the accuracy of our models and R-Squared to show the correlation between predicted results and actual value.

1.3 Thesis Overview

This thesis is organized as follows:

- ☐ Chapter 2 presents different machine learning techniques and explains how these techniques can be used for STLF task. Besides, preprocessing, mathematical analysis and visualization are discussed to provide a better overview of load forecasting process for readers. Besides, a novel hybrid deep learning model including LSTM, CNN and fully connected which is able to extract the features and the long dependency within

load data simultaneously is proposed in this chapter. Then, the experimental results of the application of the proposed model and other machine learning techniques on two real data sets are compared with each other.

- ☐ After figuring out which technique has the best performance in STLF task, the selected technique is challenged with various tasks such as prediction in different time horizons, one hour ahead forecasting of a single building data and adding weather data as an external variable to predict load data, in chapter 3.

- ☐ Finally in chapter 4, we conclude our work, highlight some challenges and suggest future works.

Chapter 2

Different Machine learning techniques for STLF task

2.1 Part1

In this part different machine learning techniques and their advantages as well as disadvantages are discussed. Load series data usually have particular attributes and so that before forecasting future load consumption, these attributes must be studied and discussed as following.

2.1.1 Definitions

As it has been mentioned before, load consumption data are time series. Thus, in order to forecast future load consumption some time series analysis are needed. Time series have important attributes such as trend or noise. In order to forecast future load consumption, some considerations of time series are needed to be taken into account.

Trend: Some of time dependent data have a linear trend in long term. It means there is an increase or decrease during the whole time, and this increase or decrease may not be in same direction throughout the given period time. However, in overall it will be upward, downward or stable. Load series data are a good example of a kind of tendencies of movement.

Seasonality: Data with seasonality or periodic fluctuations in a certain time repeat themselves. Many of time dependent data in a certain time have

same behavior. These kind of data are called seasonal data, and studying the seasonality within time series data is an important task.

Residuals (Noise): The combination of trend, seasonality and residual create the time dependent data, i.e. if the data decomposes to seasonality and trend, residuals (noise) will remain by subtracting both trend and seasonality.

Stationary: A stationary time series does not depend on observed time. In other words, a stationary time series does not have a pattern to predict the future by looking at it. If a data is stationary, it is easier to be processed and predict the future load data.

Most load series data have all trend, seasonal, noise attributes simultaneously. For instance, Fig. 2.1 shows decomposition of a seasonal load series data. Blue plot shows original data, red plot shows trend, black plot shows seasonality of data and green plot is noise. A library from Python called $seasonal-decompose()$ has been used to decompose all the seasonal data in this thesis. This function returns an object array including seasonal, trend and residuals. There is a $freq$ variable in this function which refers to the frequency of the input data, and it is important to assign a number to this variable. For instance, the $freq$ is 24 for an hourly data.

Figure 2.1: Original load series data and its decomposition. As this data is an hourly data, it has every 24 hours seasonality.

2.1.2 Machine Learning Models

Due to the importance of STLF, many authors have discussed how an accurate prediction of future load consumption can be obtained; thus, different methods have been introduced for this purpose such as Auto-Regressive Integrated Moving Average (ARIMA) [26], Seasonal Auto Regressive Integrated Moving Average (SARIMA), Regression [27], Artificial Neural Networks (ANN) [28], etc. However, in recent years, more researchers have been willing to use ANNs in load forecasting tasks since they are more flexible to work. Table 2.1 shows some studies on deep neural networks with different architectures in order to carry out a STLF with historical electric load consumption of some cities. In the following, different models including ANN, regression and classic time series analysis approaches will be discussed.

Reference	Model	City(Dataset)
[25]	FTS-CNN	Johor, Malaysia
[11]	CNN-RNN	North Italy
[11]	Parallel CNN-RNN	North China
[25]	LSTM	Johor, Malaysia
[11]	DNN-FNN	New york, USA
[28]	Seq2Seq	New England
[28]	GRU	New England

Table 2.1: Different neural networks architectures that are widely deployed for STLF with data sets from around the world. CNN: Convolutional Neural Networks, FTS: Fuzzy Time Series, LSTM: Long Short-Term Memories, DNN: Deep Neural Networks, FNN: Feedforward Neural Networks, GRU: Gated Recurrent Unit.

Auto regressive models

Auto regressive models predict the future value using the correlation between future value and past value. An important forecasting method, Box-Jenkins methodology achieved significant results in time series forecasting. This method combines Auto Regressive model (AR) with Moving Average (MA), and the combined model is called Auto Regressive Moving Average (ARMA). When a differencing order is added to this model in order to remove non-stationary within data, it is called Auto Regressive Integrated Moving Average (ARIMA) [29]. Some studies done by authors such as in [10] discuss Box-Jenkins method for short-term load forecasting. However, some of these methods have been modified to achieve more accurate prediction. [10] used a modified ARIMA to forecast hourly load consumption and has been successful to achieve better results compared to standard ARIMA. The authors used load data and temperatures from operators in Iran and MAPE was between 1.5% and 2.0% while MAPE for standard ARIMA was higher (between 2.0% and 4.5%). In the following, ARIMA models are discussed.

AR: In Autoregressive model, future variable will be predicted from the past variables. This model has an order, p, which is the number of immediately preceding values in the series that are used to predict the value at a time t. AR can be formalized as following:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_p y_{t-p} + \mu_t \tag{2.1}$$

where μ_t is the mean of series, β_p are the parameters of models and y_t is data at time t.

MA: Moving Average is an indicator of technical analyst and used widely to smooth noise based on lagging. The order q in MA models refers to q previous errors. MA can be formalized as following:

$$X_t = \theta_0 + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_n \epsilon_{t-q} + \epsilon_t \tag{2.2}$$

where the θ_q are the parameters of models and ϵ_t are the errors until time t.

Integration: To predict future load consumption with ARIMA model, a stationary time series should be used. There are different methods to stabilize a time series such as logarithm or differencing and these operators reduce changes in time series with trend elimination; in other words, they convert a non-stationary data to stationary data. An ARIMA model usually is written as ARIMA(P,D,Q) to show the needed orders which should be used to achieve the best results from this model. D represents the number of integration used, P and Q represent the orders of AR and MA part of ARIMA. In order to find out the values of P, D and Q there are different approaches. However, many experts suggest using Auto-correlation (AC) and Partial Auto-correlation (PAC) plots to figure out the values of P and Q. Nevertheless, first of all, it is necessary to find out what is AC exactly. AC is the degree of similarity between a time series data and its lags (see Fig. 2.2) and it takes a value in range [-1,1]. If there is any seasonality in data, remarkable spikes in AC plot are shown. For instance, Fig. 2.2 shows the AC plot (or auto-correlation function (ACF) plot) of an hourly load consumption from a smart building in France. This data is an hourly load consumption data, and because of this hourly attribute a seasonal approach every 24 hours can be seen in this figure.

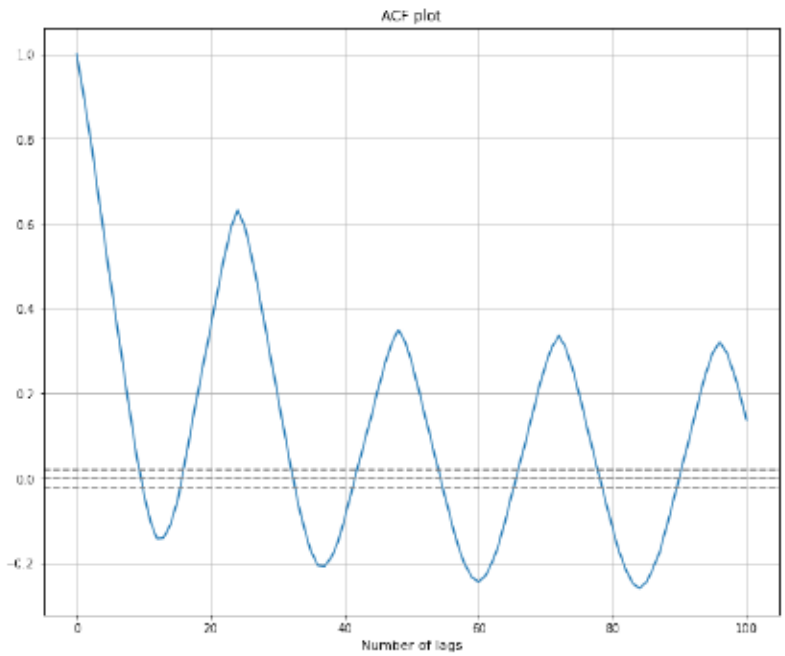

Figure 2.2: ACF plot for an example data. X axis shows number of lags, Y axis shows amount of autocorrelation

Likewise, P or, in other words, the order of AR which is a part of ARIMA model can be found by plotting PAC plot (or partial auto-correlation function (PACF) plot). F. 2.3 shows the PAC plot for same data in Fig. 2.2 .

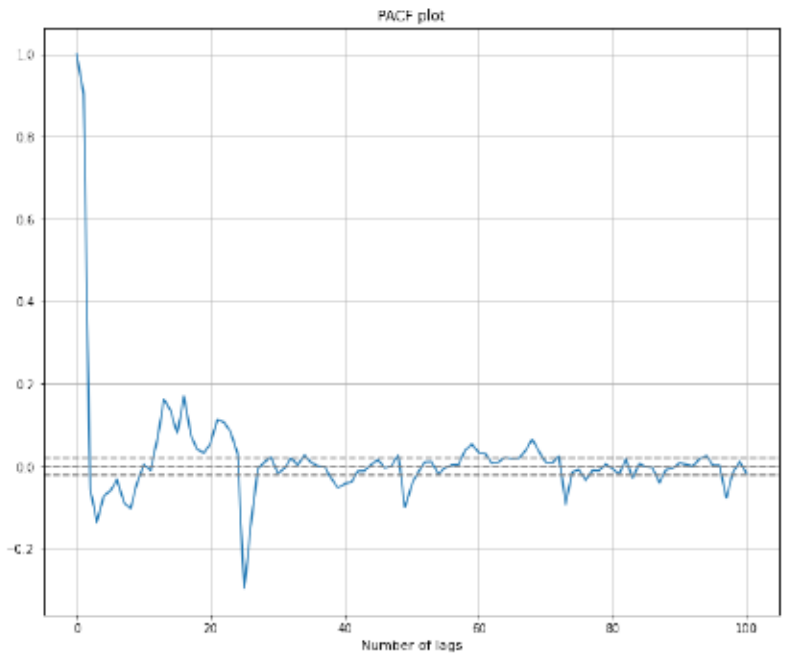

Figure 2.3: PACF plot of French data

However, there are some other tests to find the best values for D. In order to figure out whether the data is stationary or not, two different tests are proposed: Rolling statistic plot test and Dickey Fuller test. Rolling statistic plot test is a chart analysis technique to examine collected data by plotting Rolling Average. Figuring out the existence of trend in the Rolling average is the primary objective. Provided there is not any trend, data is determined as stationary. In Fig. 2.4, blue plot shows original data, and red plot shows the rolling average of data. Since there is not any trend in rolling average plot (red plot), the data is a stationary data. Besides, a Dickey-Fuller test has been applied on this data. This test is based on null hypothesis in which the nature of the series (i.e. Stationary or not) could be determined by evaluating the p-value received by Dickey Fuller test. The p-value is considered as a critical value for rejecting the null hypothesis. Thus, the smaller p-value provides the stronger evidence to accept the alternative hypothesis. In this example, the confidence interval is supposed 5%, and after applying the test the obtained p-values is less than 0.05, so data can be considered as stationary.

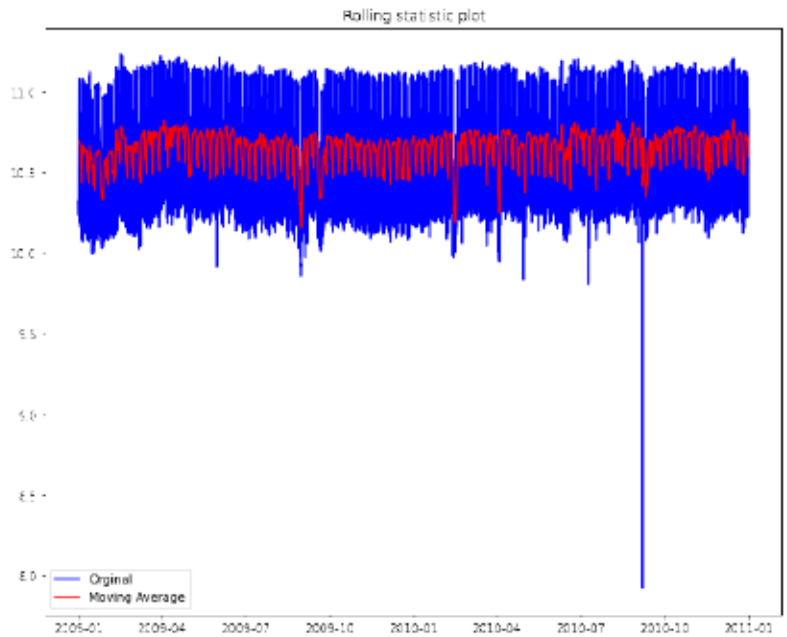

Figure 2.4: An example of Rolling test which proves that data is stationary.

Hence, one way to obtain the best values for D, after any integration of data, Rolling tests and Dickey-Fuller test can be applied and if these tests prove data is stationary, there is no need to carry out another integration. However, in case the results were different, it proves that data need more integration. It must be said that in some cases achieving stationary data is not possible. Therefore, this type of data cannot work with ARIMA models.

Seasonal ARIMA or SARIMA is another kind of statistical models which is widely used in seasonal data cases. In terms of mathematics, in addition to the same parameters with ARIMA, (P, D, Q), there are 4 other parameters for the seasonal part of these models which are called p, d, q and m. Like ARIMA, p represents the order of Autoregressive for seasonal part, d represents the order of integration for seasonal part and q represents the order of Moving Average for seasonal part. In addition, m shows the time horizon of seasonality. For example, for an hourly data, m will be 24 and for daily data, it will be 7. Therefore, SARIMA formulation is usually presented as SARIMA (P,D,Q)(p,d,q,m). The problem with ARIMA models is that these models can work with stationary data only, and they are not able to perform well with non-stationary data which is a significant limit for them.

Exponential Smoothing

Exponential Smoothing (ETS) is a well-known time series forecasting models for power systems. It can be used as an alternative to ARIMA models, in addition to its ability to be used for STLF, MTLF and LTLF. It uses weighed sum of past observation to make the prediction. Yet, the difference between ETS and ARIMA models is that ETS uses an exponential decreasing weight for previous observations [31]. It means recent observations have higher weight than past observations and the accuracy of ETS depends on some coefficients. The authors in [30] studied exponential smoothing for load forecasting and they tried to use different coefficients. They used 6 different data sets collected from China to evaluate their model. As it was assumed, they achieved high range of MAPE for different values of coefficients. Eq. 2.3 indicates the formula of the simple Exponential Smoothing. Even though there are various types of ETS, but the simple type of Exponential Smoothing is the most famous one type. However in simple ETS models there is only one hyper parameter (α) which is called damping ratio, and in some cases α is not able to help the model to predict well. However the advantage of using ETS instead of ARIMA as a statistical model is that there is no deal for ETS models whether data is stationary or not.

$$F_{t+1} = \alpha A_t + (1 - \alpha) F_t \tag{2.3}$$

where F_t and F_{t+1} indicate, predicted value in time t and $t + 1$ respectively, A_t indicates actual value at time t and α is the smoothing factor ($0 \leq \alpha \leq 1$).

Linear Regression

Since many years ago linear regression has had an inevitable role in regression based approaches. Some studies tried to use linear regression for time series or specifically for load forecasting [33]. The author in [32] studied RGUKT, R.K valley campus for STLF and achieved MAPE= 0.029 and RMSE=2.453. In another study, in [9] is used with different linear regression models including multiple linear regression (MLR), Lasso, Ridge for an hourly load data.

Linear regression is a statistical method to find the relation among variables. This method is useful to estimate a variable using influence parameters. The simplest linear regression equation is as below:

$$Y_i = \beta_0 + \beta_1 X_i + \mu_i \tag{2.4}$$

where Y is the dependent variable, β_0 is interceptor, β_1 is the slope, X is the independent variable and μ_i is residual of the model which is distributed with zero mean and constant variance. By increasing the number of variables, this model is called multiple linear regression (MLR). In order to evaluate this model, the Least Squared Error (LSE) technique is used. Our aim is that to find the best coefficients to minimize LSE. LSE evaluates the model by adding squares of error between two variables, which in our case, is between actual values and forecasted ones. Equation (2.5) shows LSE formula:

$$LSE = \sum_{i=1}^{n} (Y_i - X_i)^2 \tag{2.5}$$

where X is predicted value, Y is the actual value.

In order to use linear regression for load forecasting, some parameters such as temperature, humidity, time are needed to be used as independent variables. Likewise, the load consumption data are used as dependent variables in linear regression models. With this approach, it is possible to use linear regression as a model to forecast future load consumption. However, there are some ways to forecast load consumption without using exogenous variables. To carry out a prediction without exogenous data with linear regression, lags can be used as independent variable for load forecasting. Usually more than one lag is used as independent variable, so in this process MLR is used instead of simple linear regression. AC plot is a useful tool for time series analysis with linear regression. In this approach, those lags which their auto-correlation values are more than a certain threshold can be used as independent variables in linear regression. For instance, according to Fig. 2.2 lags [1, 2, 3, 24, 25] are chosen as independent variables with amount 0.6 for threshold. In total, in this model, lags are independent variables and actual load consumption is the dependent variable. A remarkable disadvantage of regression based models is that choosing the threshold for finding the lags as independent variables is subjective, so they can not be selected as a stable machine learning model for load forecasting.

Support Vector Regression (SVR)

Support vector machine is an approach which is used for classification and regression problems. Due to the ability of this model in different problems such as text or image analysis, SVM has become an interesting model among

machine learning techniques [34]. For instance the authors in [8] studied SVM for supervised learning methods. However, the first objective of SVM was classification. Nonetheless, after a while this model has been extended to regression problems and called support vector regression (SVR). In fact, SVR has the same procedure as SVM with some differences. Our objective in this model is to find the most appropriate hyperplane with minimum acceptable error from training samples (see Fig. 2.5). In other words, the best fit hyperplane has the maximum number of data points. In addition, the main objective is to try to minimize the coefficients through L2-norm while it is completely in contrast with LSE function in linear regression. As it can be seen in figure 2.5, there is a decision boundary (two red lines) which have ϵ distance with hyperplane. The accuracy of model depends on ϵ, so with adjusting ϵ, the desired accuracy will be achieved. Assuming equation (2.6) indicates the equation of hyperplane (in this case, it is a linear equation).

$$y_i = wx_i + b \tag{2.6}$$

Therefore, the solutions and constraints are as equations (2.7)-(2.9):
Solution:

$$min\frac{1}{2}||w||^2 \tag{2.7}$$

Constraints:

$$y_i - wx_i - b \leq \epsilon \tag{2.8}$$

$$wx_i + b - y_i \leq \epsilon \tag{2.9}$$

where x is input, y is target and w is the weight. w SVR also can be used for load forecasting problems. Authors in [35] proposed a new SVR for short-term load forecasting. They evaluated their model using two data sets, ISO New England and North-American Utility. They forecasted 24-hour ahead and 1-hour ahead and achieved reasonable MAPE between 0.75% and 2.25% for test and validation sets. In another study [36], authors applied SVR on electricity load demand recorded every half an hour from 1997 to 1998. They evaluated their model using exogenous variable (temperature) and without it. They trained the model once with winter data and then with Jan-Feb data. MAPE for different times and variables have been between 1.95% and 3.5%. However, they concluded that it is better to predict future load consumption without using temperature data, because it is difficult to predict future temperature and that leads to higher error.

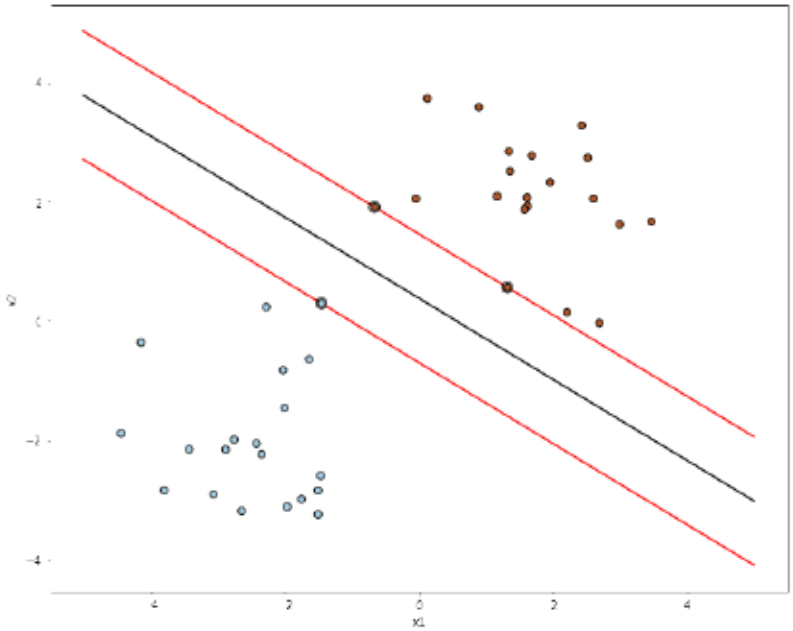

Figure 2.5: An illustrative example SVR. Red line show the boundary lines, black line shows hyper plane

Fully connected Neural Networks

Nowadays, numerous neural networks such as fully connected [37] ones have been introduced. However, it is difficult to train a fully connected network for load forecasting due to the over-fitting problem. Overfitting refers to the production of analysis from a statistical model in which the model is extremely trained. The problem is that when an overfitting happens, the model learns very well the parameters but it is not able to predict well (i.e weak generalization ability) [38]. Therefore, same approach with linear regression used to predict future load consumption through fully connected neural networks.

In neural networks, there are 3 different layers: input layer, hidden layer and output layer. The depth of network depends on the number of layers in the hidden layer. In fully connected neural networks, all the neurons in each layer are connected to the neurons in the next layer; in other words, every output of layers is used as input for next layer while each neuron has an activation function (usually a non-linear function). Fig. 2.6 shows a simple

network with just one hidden layer. Each neuron has a specific weight and for every layer a bias term is considered. In total, outputs of layers are computed as:

$$a_l = W_l h_{l-1} + b_l \tag{2.10}$$

$$h_l = f(a_l) \tag{2.11}$$

where l indicates layer number, f is activation function which in terms of the most popular one, it can be referred to ReLU or Sigmoid functions (see Fig. 2.7). W is weighted matrix of layer l, h_l is output of activation function and b_l is bias term of layer l. It is obvious if W is $r \times 1$ matrix, a and h will have $r \times 1$ dimension. Therefore, the transpose of W should be used. If $P(\alpha)$ is considered as predicted output from neural networks, α represents parameters of neural network and y is the actual value, for N input-output, the loss function is:

$$L = \frac{1}{N} \Sigma_{i=1}^{N} (y_i - p(\alpha))^2 \tag{2.12}$$

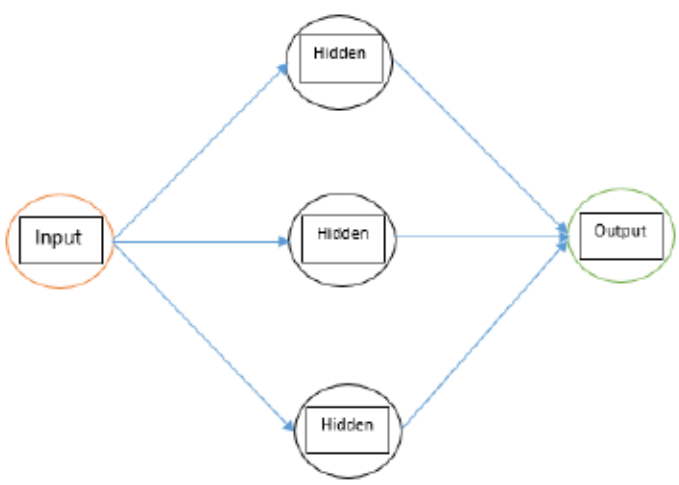

Figure 2.6: an one-layer neural network example

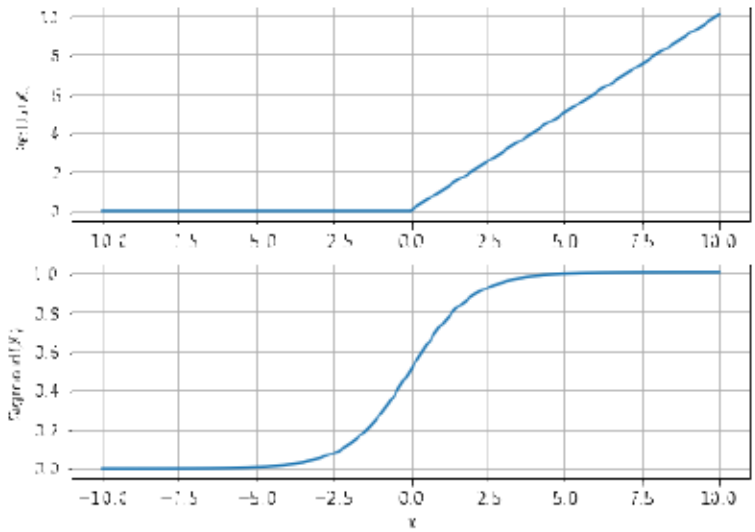

Figure 2.7: ReLU and Sigmoid function

The primary goal is to optimize the parameters of neural network. For the same purpose, the loss function must be minimized as much as possible. A regularization penalty term is usually used to avoid overfitting (see equation (2.13)).

$$L = \frac{1}{N}\Sigma_{i=1}^{N}(y_i - p(\alpha))^2 + \Omega(\alpha) \tag{2.13}$$

where $\Omega(\alpha) = \lambda||\alpha||^2$, using norm-2 and a hyperparameter to control the regularization strength. For the learning processing there are different algorithms such as RMSprop, SGD, ADAM [39]. However, due to its ability of working with non-stationary data, ADAM is the most appropriate choice for load forecasting.

LSTM

Long short-term memory (LSTM) [40] is a special case of RNNs. RNNs are based on control theory, and because of this reason they are used to process sequence of inputs [41]. However, experimental results have proven that RNNs due to the gradient vanishing problem are not able to perform well if long time interval is used as input. In order to overcome this disadvantage, in many recent researches RNNs were replaced by LSTM. In load forecasting, many studies used LSTM and improved the accuracy of their approaches

by finding the dependency within load series data. The authors in [42] used LSTM network to carry out load forecasting in different time horizons including 24 hours, 48 hours, 7 days and 30 days and compared LSTM with some traditional models such as SARIMA and ARMA. The authors in [43] also studied STLF by using an architecture including LSTM and fully connected layers. Moreover, they used historical as well as prediction data as input for their model. In addition to these particular research efforts of using LSTM for load forecasting, LSTM has been widely used in various hybrid models such as the one in [12].

In terms of LSTM structure, it consists of 3 gates, namely input gate, output gate and forget gate. Each of these gates are determined to perform a specific task. Equations (2.14) trough (2.18) show the formulation of LSTM in details. Eq. 2.14 indicates the calculation inside of input gate. This gate determines when C_t needs to be updated. Eq. 2.15 is related to forget gate and it aims to find out if the state of cell C_{t-1} must be forgotten or not. The output gate, is called in some references control gate, determines that to which part of cell C_t the output h_y must be added [13].

$$i[t] = \psi(L_i * h[t-1] + b_i) \tag{2.14}$$

$$f[t] = \psi(L_f * h[t-1] + b_f) \tag{2.15}$$

$$O[t] = \psi(L_o * h[t-1] + b_o) \tag{2.16}$$

$$C[t] = f[t] \odot C[t-1] + i[t] \odot (\phi(L_c * h[t-1] + b_c)) \tag{2.17}$$

$$h[t] = \phi(C[t]) \odot O[t] \tag{2.18}$$

where L_i, L_f, L_o are the learning parameters, b_i, b_f, b_o, b_c are biased vectors, ϕ represents hyperbolic tangent, likewise ψ represents Sigmoid function, $f[t]$ is forget gate, $i[t]$ is input gate, $O[t]$ is output gate, $C[t]$ is the state of this cell to encode information from the input sequence, $h[t]$ is network output and all of $[t]$ symbol refers to time t and finally, $\odot$ is used as a symbol for Hadamard product. Fig. 2.8 displays LSTM structure.

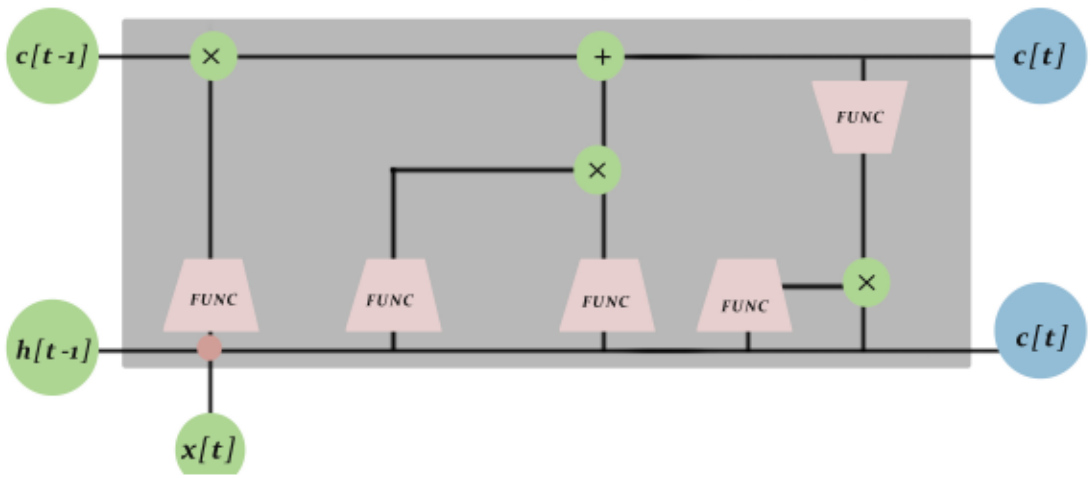

Figure 2.8: Inner structure of LSTM

CNN

CNNs are a big family of artificial neural networks [44] designed to filter and extract the features of input data. They have been widely used in various areas thanks to their ability to handle data with different dimensionalities. For instance two dimensional and three dimensional CNNs are recognized as a powerful network for performing image processing and classification [45, 46],as well as computer vision tasks [47]. Moreover, in recent years they have been deployed in different other fields including Natural Language Processing (NLP) [48], audio recognition [49], medical [50] and load forecasting [11]. Existing diversities among the load profiles using CNN networks may come up with some difficulty. The complexity of human behaviours, working days, time and weather data affect directly the load profiles [51]. To overcome the complexity of load profiles, CNNs need to have huge amount of input data as training set in order to learn all parameters.

From a technical point of view, CNNs are based on a discrete mathematics operator called convolution as shown in Eq. 2.19 shows the operation calculation. In Eq. 2.19, Y is used as an output and x is the input. In addition w represents the kernel. The i-th output is given as follows:

$$Y(i) = \Sigma_{i,j} x(i-j) w(j) \tag{2.19}$$

where j is ranging from 0 to $k-1$ and then it makes Y to have $n-k+1$ dimensions, and n is the input's dimension.

Despite the fact that convolution operation is a simple mathematical formula, CNNs work a little different. Fig. 2.9 shows the inner structure of this neural networks family, and as it can be seen in this figure convolution filter slides over the whole input data to extract the features. According to [52], in convolution operation, firstly kernel and filter are convolved and the results of this operation is added to a bias term. This mathematical operations is finished when a complete feature map is achieved. Equations (2.20) and (2.21) show the complete convolution operation in artificial networks.

$$Y_{ij}^{m} = sum(k_m \circledast xf_{ij}) + b_m \tag{2.20}$$

$$f^m = activation(Y^m) \tag{2.21}$$

where Y^m indicates the output, m represents the m-th feature maps, i, j indicate the vertical and horizontal steps of filter respectively, xf_{ij} is the filter matrix, k_m represents the kernel matrix, b_m is the bias term, and finally f^m is the activation function's output. It must be said that Eq. 2.20 shows the convolution operation formula while Eq. 2.21 shows the activation function for the m-th output.

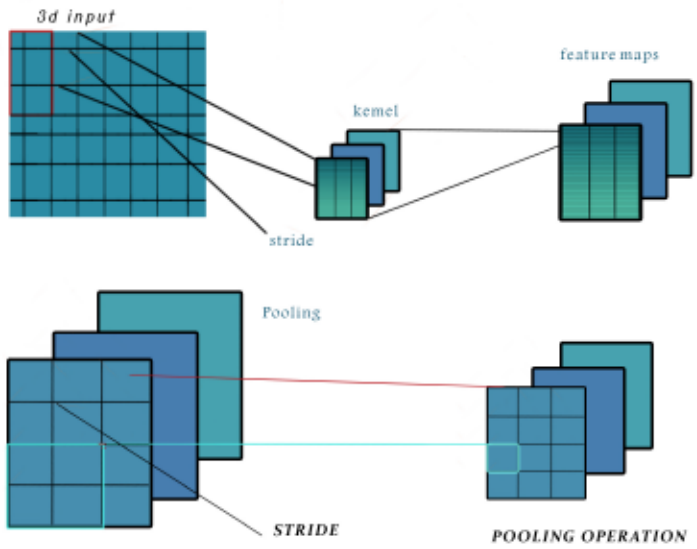

Figure 2.9: 2D-Convolution and Maxpooling operations. In this instance figure the filter size is [3,2], and for the sliding part the size is chosen [2,2]

In terms of the CNNs architecture there are usually convolutional layers, pooling layers and fully-connected layers. Pooling layers are used after CNN layers to carry out a down sampling operation while keeping the quality of input data. This dimension-reduction operation is useful because it makes the model prepared to learn the parameters through back-propagation algorithm. Finally fully connected layer is used to perform the final prediction by carrying out a combination of all features. However, according to the nature of load data, this thesis focuses on one-dimensional CNN.

2.1.3 The problems of the discussed model

Statistical models, including ETS and ARIMA, and regression based models, including linear regression and SVR, are two classic models which can be used for load forecasting. However, they come with some disadvantages, such as the limit of working with non-stationary data in ARIMA models, to work with regression based models data should be pre-processed well and being linearized which will take some time as well as increasing the error. Besides, all kind of deep learning models are not able to perform accurate and provide optimistic results, for instance fully connected networks can not

extract the features of input data lonely so they need to be fed by some pre-processed data, like linearized load data, to become able to predict future load data. It is better to used fully connected networks combined with CNN and LSTM models. However, most of the combined CNN-LSTM models, use these two networks consecutively which leads to having more errors since the extracted features from CNN will affect the LSTM units. Thus, the lack of a powerful model which can process non-linear as well as non-stationary data and provide more accurate results is completely obvious. Following part will propose a new deep learning model.

2.2 Part 2

This part first introduce the PLCNet model to address all the discussed problems in the existing models, then the experimental results are discussed to compare the performance of the different machine learning models in STLF.

PLCNet model

This thesis discusses a new methodology combined with CNN and LSTM to carry out load prediction. Despite other efforts such as those reviewed in the introduction that combined both approaches, the methodology presented here is completely different. For instance the authors in [53] proposed a CNN-LSTM model, so that CNN is first used to extract the features of input data and then output from CNN is used as LSTM input. The problem within this model is that extracted features affect the training of LSTM. In order to solve this problem, in our proposed model LSTM and CNN networks are used in two different paths without any correlation between those two paths. Fig. 2.10 shows the frame-work of the proposed methodology. As it can be seen input signals are firstly entered into two paths to be processed by LSTM and CNN paths. These two paths extract the features as well as the long dependency within data, and prepare the input data to make final prediction. In order to carry out the final prediction, a fully connected path including dense and dropout layer have been implemented, and finally predicted data are compared by actual values. Since the CNN and LSTM networks are implemented parallel, the model is called parallel LSTM CNN Network or PLCNet.

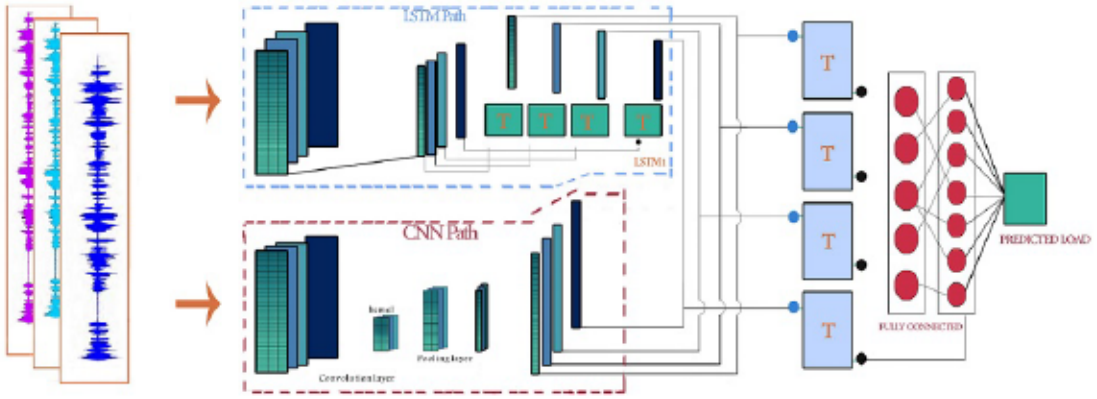

Figure 2.10: The framework of the PLCNet model

In the CNN path, capturing the feature of local trend is the main objective. In this path the data is convoluted through a Conv-1D layer within 64 units and filter size [2]. After convolution layer Maxpooling layer is used to reduce the dimensionality of the data by down sampling while keeping its quality. In the next layer, another Conv-1D layer, but within 32 units, is implemented. The data in final layer continue through flatten layer. All of the units are activated by Rectified Linear Unit (ReLU).

To capture the long-term dependency within data the LSTM path is used. To start working with LSTM network, the data go through a flatten layer. After passing through the flatten layer, input data is ready to be entered as input for LSTM layer. An LSTM layer with 48 units and the activation function is ReLU.

After passing through LSTM and CNN paths, the processed data is ready to be entered into fully connected layer. As it was mentioned before there is no correlation between two paths, thus in order to prepare data for prediction, the outputs are concatenated in a merge layer. The merged data are entered into a LSTM layer with 300 units and ReLU activation function to learn the long-dependency within output data from two paths, and then the output of LSTM layer will feed the next dense layer. After that a dropout (30%) [55] layer is implemented to avoid any overfitting. Getting back to the architecture of the PLCNet, two other dense layers are used to prepare the data for final prediction. Since this model aims to carry out a prediction for two data sets and various time horizons, the number of units in each dense layer is different. However, all the existing units in fully connected path are

activated by Sigmoid function. Concerning the fact that the PLCNet model also must be evaluated for different time horizons,the number of units in LSTM-Dense path can be different.

2.2.1 Experimental Results

To prepare the data sets for the considered models, city Johor data is divided into 2 sets, training set which contains year 2009 load consumption and year 2010 load consumption used as test set. German data set also is divided, so that 2012-2015 load consumption are used as training set and 2016-2017 are used as test set. All the models are implemented in Python. In order to implement deep neural networks (DNN), we used Keras library with backend of TensorFlow. In addition, Scikit-learn, Statsmodels and Pmdarima libraries were used for regression and time series modeling and analysis.

2.2.2 Case studies

Two different data sets are used to carry out STLF. The authors in [25] used load consumption of the city of Johor in Malaysia to predict day ahead load consumption (hourly prediction) using a model which is a combination of neural network and fuzzy time series. They used a new model which was a combination of Fuzzy Time Series and CNN (FTS-CNN). They firstly, through fuzzy logic, created a sparse matrix and then, through CNN extracted features and carried out STLF. They also tried other models including SARIMA, different LSTM models, different probabilistic weighted fuzzy time series and weighted fuzzy time series. But, their proposed model (FTS-CNN) could achieve better results compared to other models for two different years of Malaysia data and RMSE was 1777.99, 1702.70, respectively. This data is from a power company in this city for years 2009 and 2010 and consists of hourly electric consumption in MW. It has 17518 rows which show the aggregated load consumption of these two years in this city. Fig. 2.11 shows a Box plot of whole dataset which illustrates how the loads are distributed among days of a week.

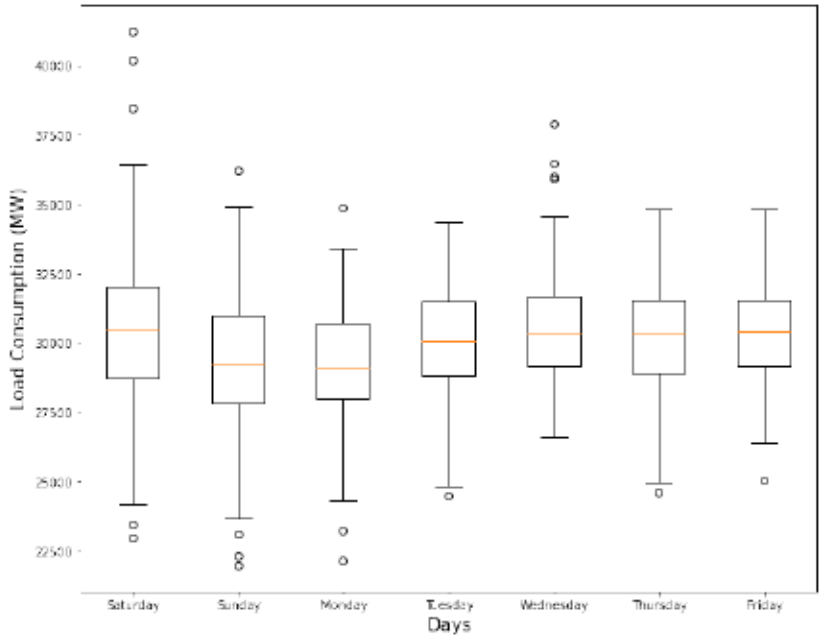

Figure 2.11: the Box plot of load consumption during a week in Johor (Malaysia)

Another data is Germany country-wide daily aggregated electric consumption since 2006 to 2017 in GWh. This data is provided by Open Power System Data (OPSD) and is used to predict day ahead load consumption. This data has 2186 recorded electric consumption in Germany. Fig. 2.12 shows the Box plot of this data during a week.

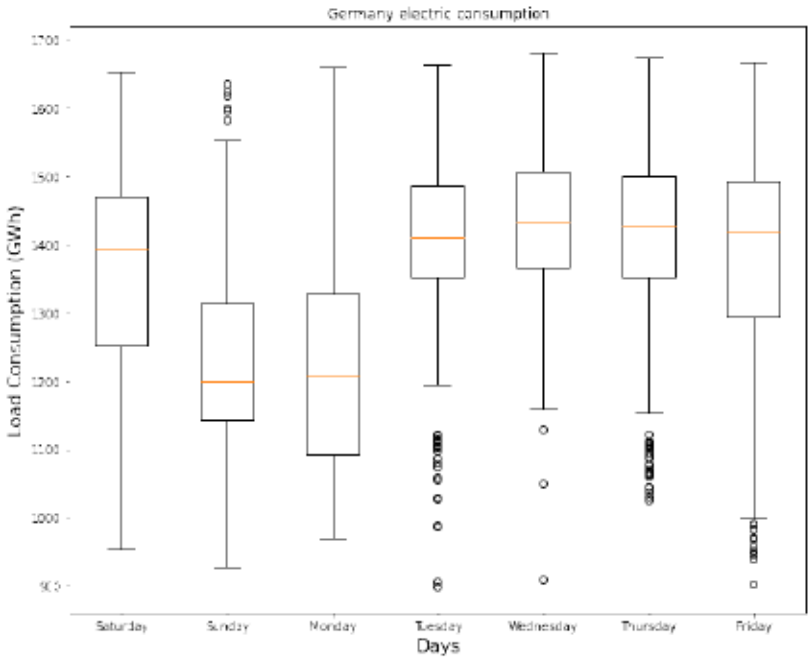

Figure 2.12: Box plot of electric consumption in Germany 2012-2017

Part of Malaysian data and German data have been decomposed into seasonal, trend and noise. Fig. 2.13 and 2.14 show the original data and their decomposition. Black plots in both figures show the seasonal part of each data.

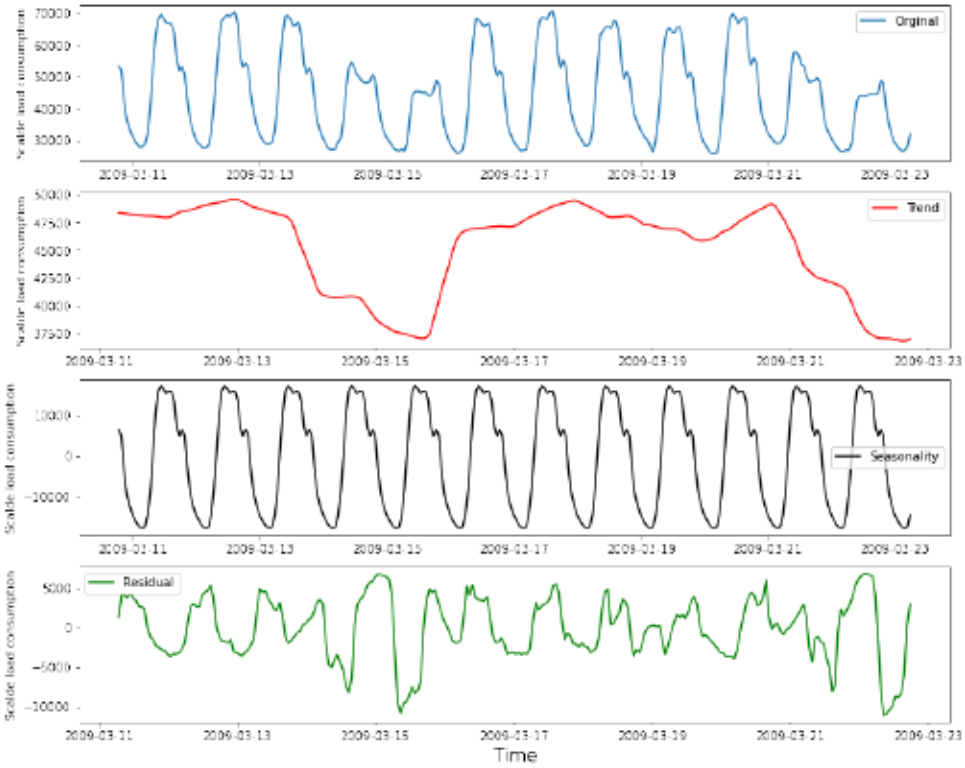

Figure 2.13: Part of the decomposition of Malaysian data.

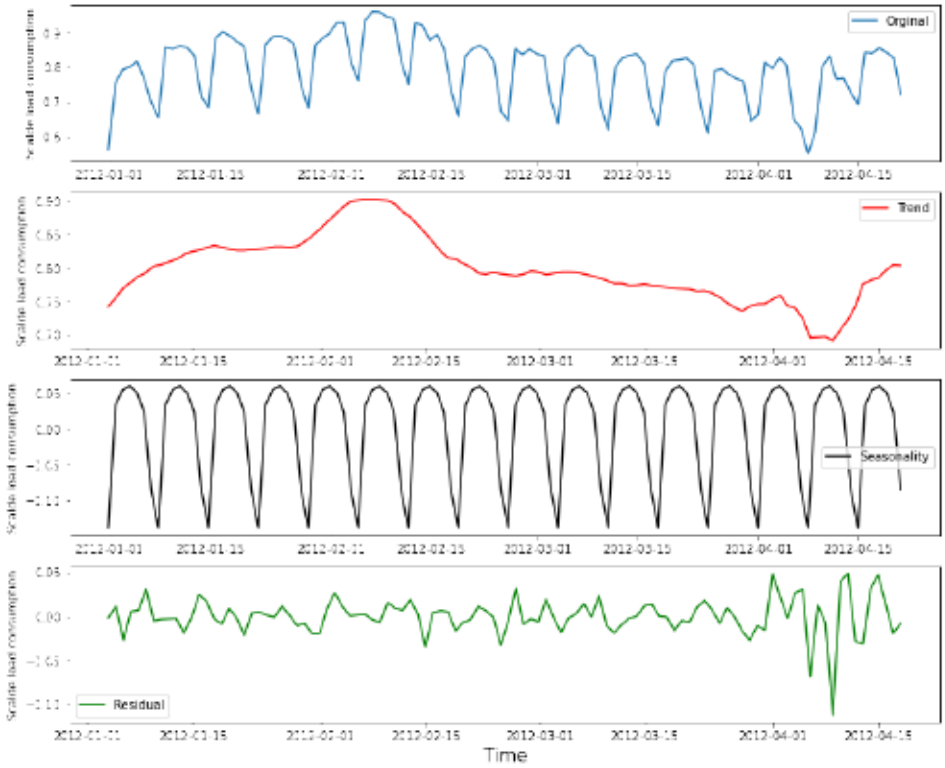

Figure 2.14: Part of the decomposition of German data.

One of the main reasons these two data sets were used in this study is that they are clean data sets which make users able to carry out load forecasting since there is no missing value. In terms of outlier, there were two outliers in Malaysian data set in which they were found easily and were fixed, while German data set has got no outlier. These two data sets are two highly aggregated hourly and daily data and the difference in the number of recorded sample provides the opportunity that models will be tested with high and low number of training data.

2.2.3 Data preparation

The acquired results from practical experiments proved that in order to work with deep learning models, data should be prepared well [54] and results showed pre-processing is more significant than training process. As discussed before, in load forecasting even though some parameters such as holidays,

temperature, humidity, etc. affect the model, our aim is to carry out STLF using just previous load consumption data. Therefore, data must be prepared specifically for each model. This thesis scaled the data between 0 and 1 through equation (2.22) which is written as follows:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.22}$$

2.2.4 Evaluation Metrics

In order to evaluate models performance, root mean squared error (RMSE), mean absolute percentage error (MAPE) and coefficient of determination (R^2) are used.

$$RMSE = \sqrt{(\frac{1}{N})\sum_{i=1}^{N}(A_i - F_i)^2} \tag{2.23}$$

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{A_i - F_i}{A_i}\right| \tag{2.24}$$

$$R^2 = 1 - \frac{SSR}{SST} \tag{2.25}$$

$$SSR = \sum_{i=1}^{N}(A_i - F_i)^2 \tag{2.26}$$

$$SST = \sum_{i=1}^{N}(A_i - \bar{A}_i)^2 \tag{2.27}$$

where A_i and F_i refer to actual and forecasted value of i-th data, N is the data size, $\bar{A}$ is the average of actual data. In addition, SSR stands for sum squared regression and SST stands for total sum of squared.

2.2.5 Implementation

After normalizing whole data sets, they must be prepared to be used as input for machine learning models. So in first step all of them are converted into a *Numpy* array to facilitate the training process. The type of model's

output is array and to visualize them, *Matplotlib* library is used. Finally from *scikit−learn* library all the afforementioned metrics are called to evaluate the model's performance. Following the evaluation of the model's performance are presented.

The Evaluation of ARIMA

In ARIMA models, time series data are decomposed into m series to eliminate of hourly/daily seasonality within data. According to Fig. 2.13 and 2.14 there is a daily seasonality in Johor data (every 24 hours), and weekly seasonality in German data (every 7 days). Therefore, instead of using simple ARIMA, seasonal ARIMA (SARIMA) is being used to carry out STLF. In order to find the parameters of SARIMA, Auto-arima function from pmdarima library in python was used. This function tries to find the best number for parameters through carrying out a comparison among different parameters. For German data, ARIMA (5,1,0)(5,0,5,[7]) became our final models and ARIMA (1,0,1)(2,0,0,[24]) achieved best results for city of Johor data. Fig. 2.15 and 2.16 illustrate predicted results for both data from ARIMA model.

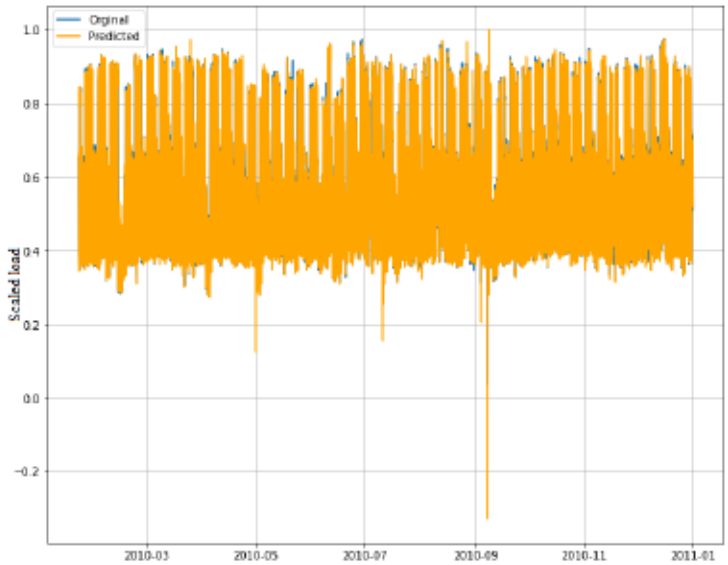

Figure 2.15: Predicted and actual results from SARIMA, Malaysian data.

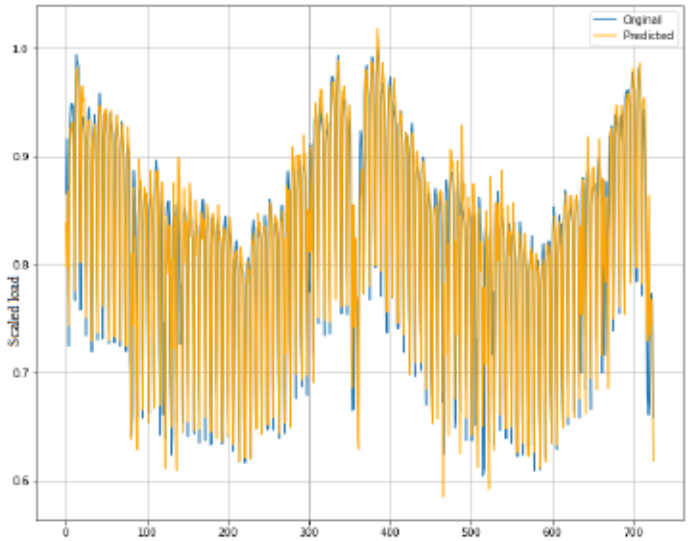

Figure 2.16: Predicted and actual data from SARIMA, German data.

Even though results show that ARIMA is able to perform well for both data sets, the problem with this technique is that it takes much computation time since it needs to solve some complex mathematical formulas.

The Evaluation of Exponential Smoothing:

Exponential Smoothing (ETS) is an alternative approach for load forecasting. Training and test sets of two data sets are applied to this model to carry out $t+1$ forecasting. Fig. 2.17 and 2.18 show predicted and actual test set for both data. These plots prove that ETS fails to perform accurately in STLF.

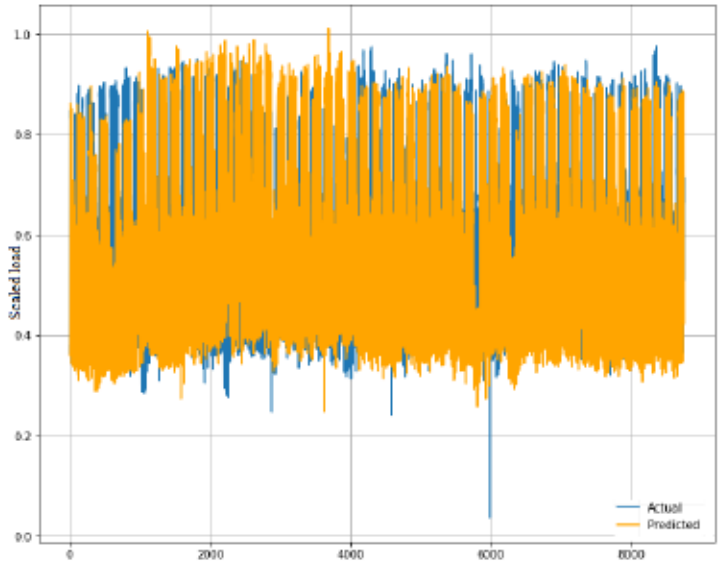

Figure 2.17: Predicted and actual data from ETS, Malaysian data.

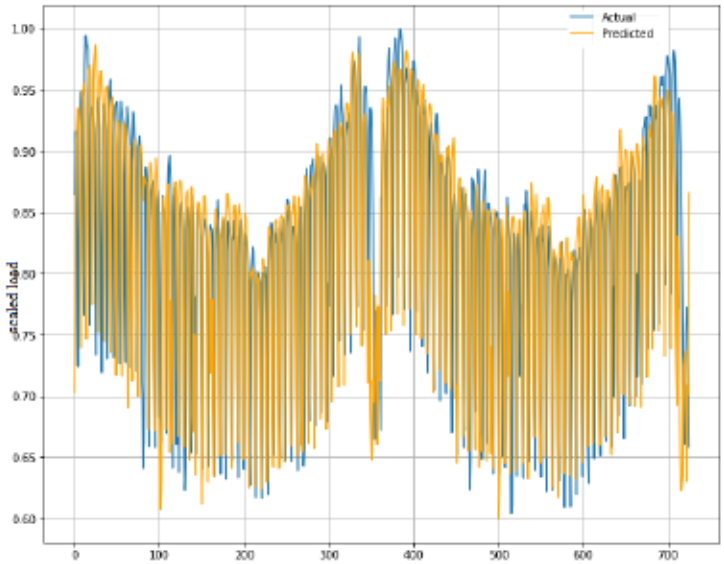

Figure 2.18: Predicted and actual data from ETS, German data.

The other problem with ETS is that same as ARIMA, various parameters through mathematical computation must be analyzed, and it leads to higher

computation time.

The Evaluation of Linear Regression

For the linear regression model, the ACF plot is used to find out how many lags can be used as linear regression variables (independent data). In Fig. 2.19, ACF plot of Malaysia is illustrated. For this data set, the threshold is chosen 0.75. Lags [1, 2, 23, 24, 25, 47, 48, 49, 71, 72] become the independent data and actual load consumption are used as targets (dependent data). Scaled data is divided into training and test set. As 10 lags are used as variables, the shape of train set is (8723,10) which started from the first day of 2009 to the first day of 2010 and test set has the shape of (8723,10) from first day of 2010 to the end of this year.

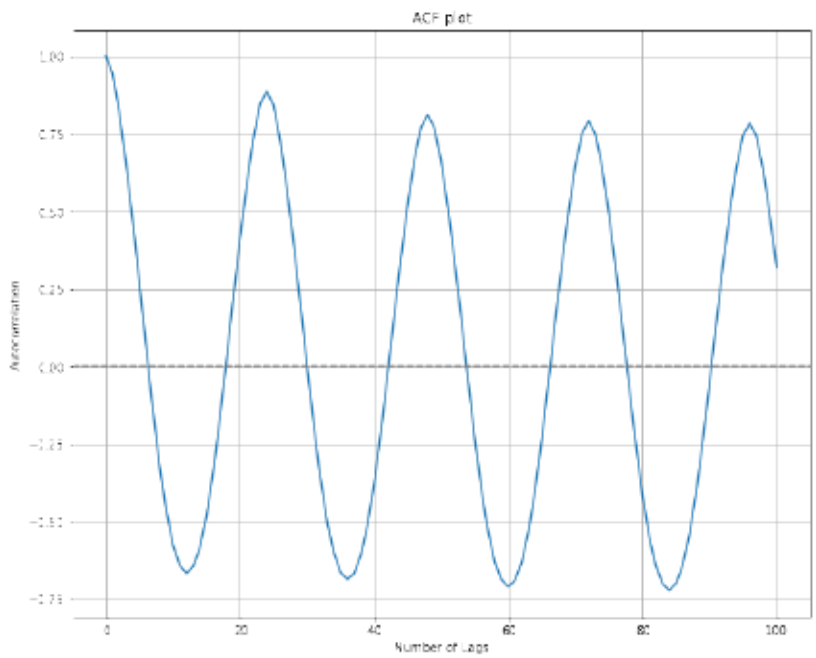

Figure 2.19: ACF plot of Malaysian data.

Fig. 2.20 shows predicted and actual data of load consumption for year 2010 in Malaysian data, and according to the plot linear regression has done an accurate prediction.

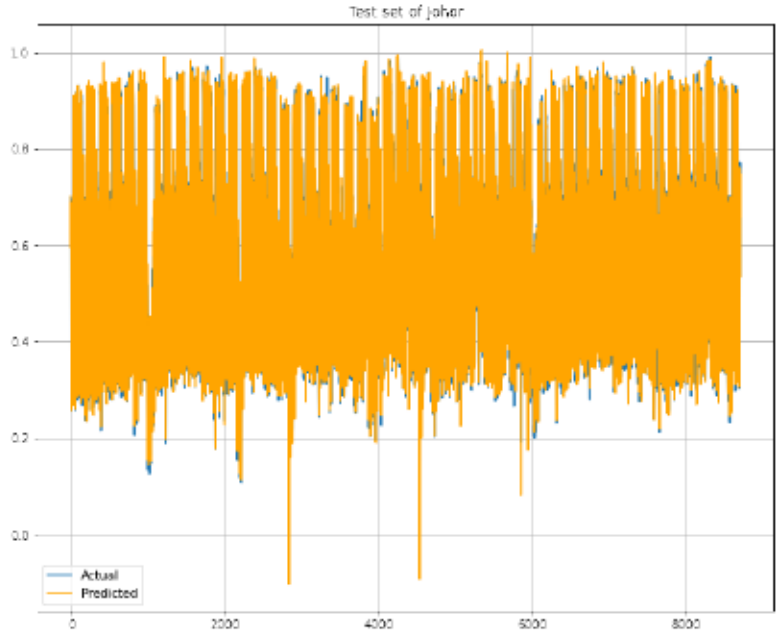

Figure 2.20: Actual and predicted results from linear regression, Malaysian data.

However, for German data set, there are some differences. The first difference is that, 0.69 is for the threshold. Fig. 2.21 shows AC plot of German data. According to this plot and threshold, lags [7, 14, 21, 28] are being used as independent variables for MLR. The shape of training data is (1456,4) and test set is from 2016 to end of 2017 with the shape of (702,4). As this data is a daily data, the model predicted daily load consumption but as not good as predicted results from Malaysian data. Fig. 2.22 shows actual and predicted results from test set. While linear regression is able to predict hourly load series accurately yet it fails to forecast accurately future load consumption of daily load series. The difference in the number of lags as variables explain well the reason why the results are not similar. The number of variables (lags) for city of Johor data is 10, while it is 4 for German data. This point is the main weakness of linear regression.

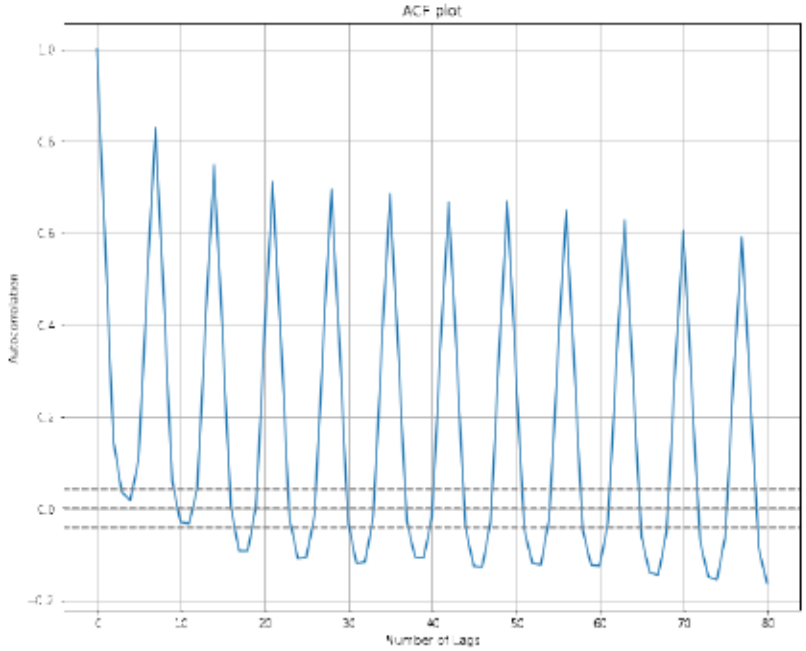

Figure 2.21: ACF plot of German data to find out which lags should be taken as variables for linear regression.

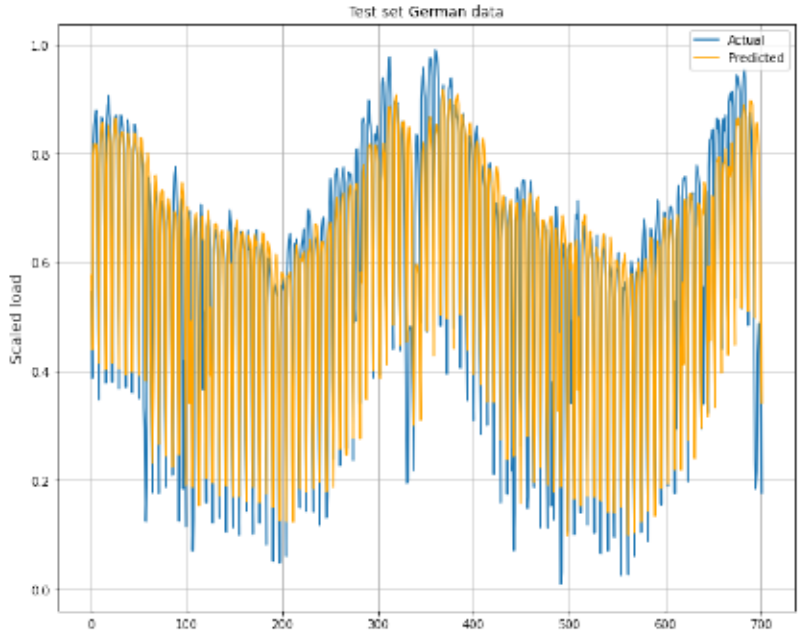

Figure 2.22: Actual and predicted results from linear regression, German data.

However compared to ARIMA and ETS, linear regression is a faster and more accurate technique but choosing the threshold is almost subjective. Another problem in terms of choosing threshold is that for those data sets which do not have sufficient autocorrelation in their nature, linear regression is not able to perform well. Because of these reasons this technique can not be considered as a powerful tool for STLF task.

The evaluation of SVR

SVR the other regression based technique is another machine learning which is evaluated in this section through applying same training and test sets from linear regression section. There are various parameters which affect SVR to perform well, and among all these parameters choosing the most appropriate kernel is a critical task. For city Johor data, 'linear' kernel had the best performance compared to other kernels and for German data 'radial bias function (rbf)' was used. Fig. 2.23 and 2.24 show the predicted load consumption from SVR for both data sets, respectively. As it can be seen from the figures, SVR forecasted future load consumption of Malaysian

case with less accuracy compared to linear regression. Even tough, SVR have achieved more accurate results than linear regression for German data, this model also can not be a good candidate for STLF, because of having same problem within linear regression. To work with these regression-based techniques some reprocessing such as selecting a value for threshold which is subjective is essential. In addition, as it discussed in the last section having a reasonable amount of autocorrelation within data is another serious problem with these models.

Figure 2.23: Actual and predicted results from SVR, Malaysian data.

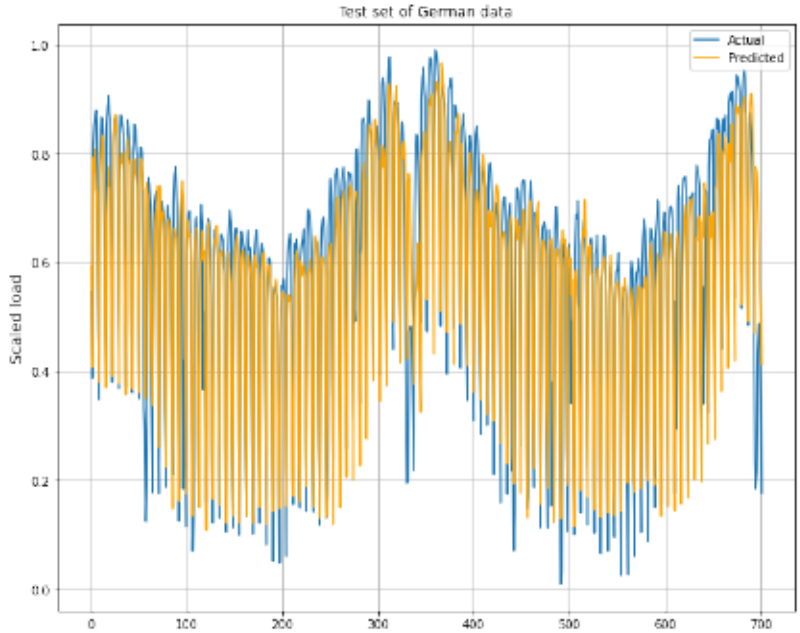

Figure 2.24: Actual and predicted results from SVR, German data.

The Evaluation of Fully connected Neural Networks

A fully connected network [56] which consists of only dense and dropout layers is implemented in this section. To evaluate the network same training and test set in regression based techniques is used here. This network has 3 hidden layers in addition to input and output layers. First layer is input layer, and hidden layers consist of 27, 18 and 18 dense layers, respectively. To avoid overfitting, the dropout technique is adopted in this network. This model learns the parameters through ADAM optimizer in 20 epochs and the size of each batch is 1. In addition, for whole layers, ReLU is used as activation function. ReLU stands for Rectified Linear Unit and it works like linear function with a difference which is its output for negative inputs is zero. This attribute helps DNN models to avoid vanishing gradient problem. Fig 2.25 and 2.26 show the results of DNN model.

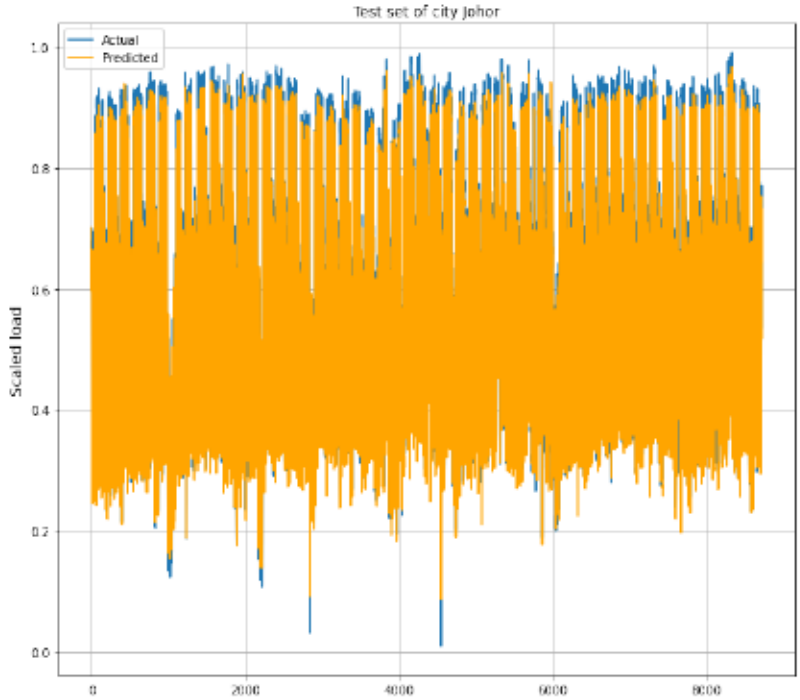

Figure 2.25: Actual and predicted results from fully connected, Malaysian data.

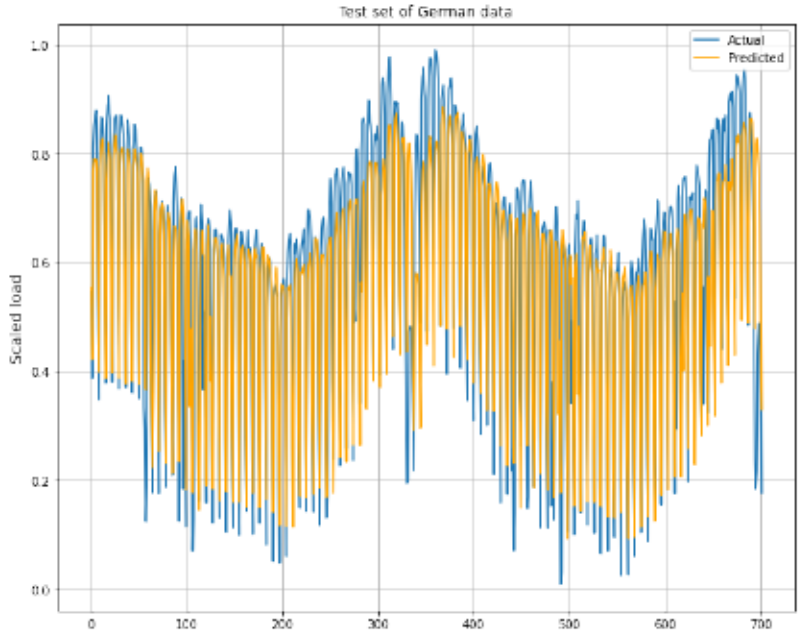

Figure 2.26: Actual and predicted results from fully connected, German data.

The results of this model is close to linear regression, and since the pre-processing is same as regression based this model comes with same regression based technique's disadvantages. It proves that fully connected network can not be our final choice for STLF task, however for more simple electric data which are not complex this model can be used. However, neural networks are widely used since they are able to process the data well and carry out a good prediction, while in the discussed approach in this section there are a lot of pre-processing and analysis same as regression based approaches. Thus, there is not any extra advantages for this methodology to be used as load predictor tool compared to regression based models.

The Evaluation of Vanila LSTM

In this thesis, the LSTM model [57] studies last 24 hours load consumption and predict next hour consumption in Johor data, while in German data, in order to predict next day load consumption, it studies last 7 days and predicts next day data. In terms of architecture, it has one LSTM layer, while one dense layer is used as output. Same with fully connected network in previous section, the used activation function is ReLU. In addition, model is trained

by ADAM optimizer for Johor data in 200 epochs and RMSprop [58] for German data in 150 epochs. This model proves that LSTMs are a powerful tool for STLF, due to the accurate results that our model has achieved as well as their independency to auto-correlation of input data.

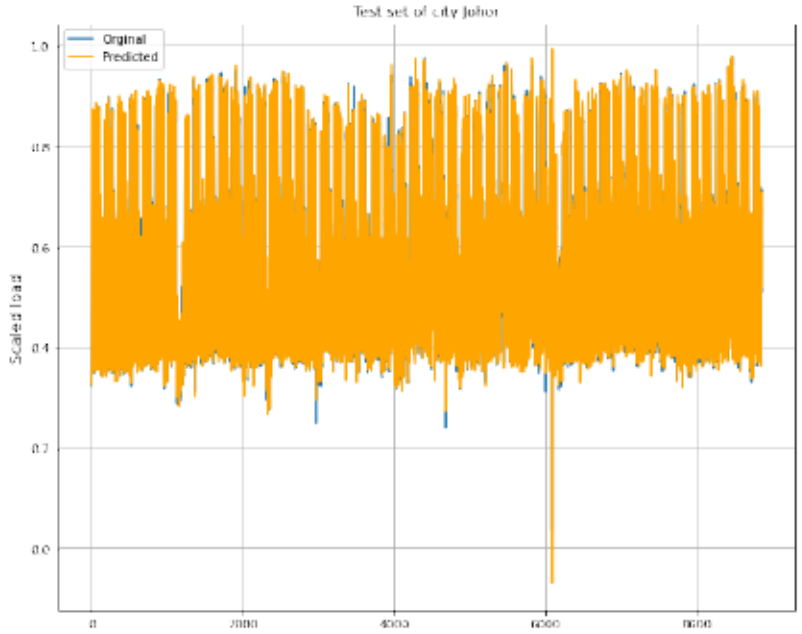

Figure 2.27: Actual and predicted results from LSTM, Malaysian data.

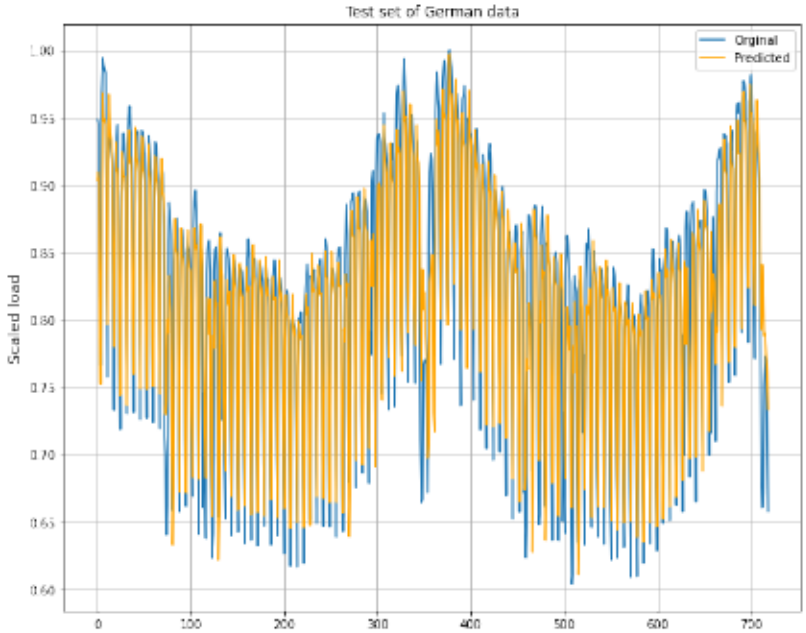

Figure 2.28: Actual and predicted results from LSTM, German data.

The Evaluation of CNN-LSTM

As discussed before, CNNs are well-known networks for feature extraction. LSTM also showed their ability to predict short-term load consumption. Therefore, a hybrid model of CNN and LSTM can come with a number of advantages. In order to work with this hybrid model, CNN layers should be implemented first to apply historical data. In the next step, extracted features from CNNs are used as input for LSTM layers. This section uses a 7-layer model to apply on the same test and training set which was used for LSTM model in previous section. In layer #1 and layer #2, 1-D CNNs with the ReLU activation function and filters=64 and kernel size=3 are implemented. After that a Maxpooling and Flatten layer are used to prepare data for LSTM layer. In layer #5, 200 LSTM neurons with ReLU activation function are implemented. To analyze the results and predict load consumption, two Dense layers with 200 and 1 neurons are implemented while ReLU used as activation function. Same with other DNN models ADAM optimizer has the role to compile the model for Malaysian data and RMSprop optimizer is using for German data. Fig. 2.29, 2.30 the results of forecasted data with

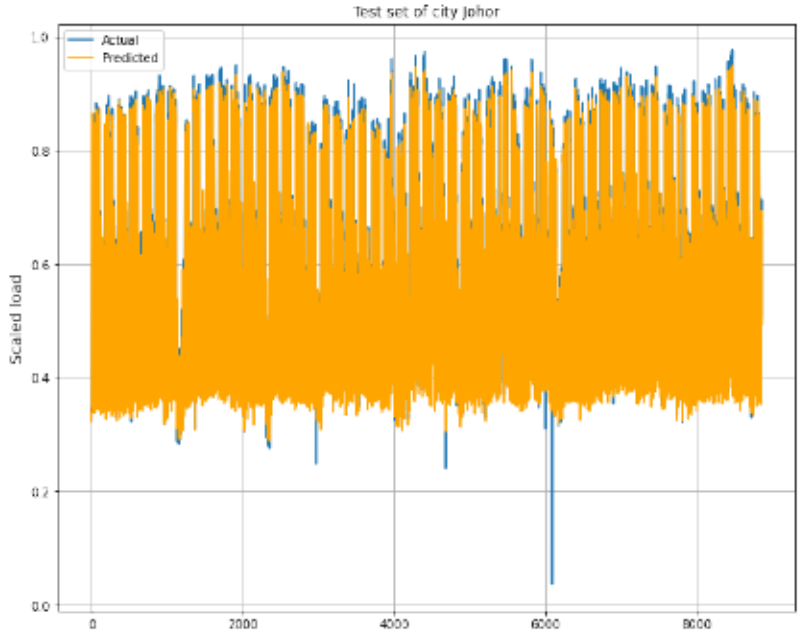

Figure 2.29: Actual and predicted results from CNN-LSTM, Malaysian data.

CNN-LSTM model.

The evaluation of PLCNet

As it mentioned before, PLCNet includes two different paths, CNN path and LSTM path, and these two paths are fed simultaneously by historical load data. According to the Fig. 2.31 and 2.32, the model has a good performance for both data sets.

2.2.6 Results

The detailed experimental results are presented numerically in tables 2.2 and 2.3. As shown in these two tables, the MAPE and RMSE of the PLCNet model are the smallest while the R^2 score is the highest value. Regarding the largest amount in error, the MAPE and RMSE of ETS have the highest error value in both German and Malaysian data sets, where it has got 0.36 and 8.81 for RMSE and MAPE for Malaysian data and 0.316 and 33.63 for RMSE and MAPE for German data. According to the MAPE and RMSE

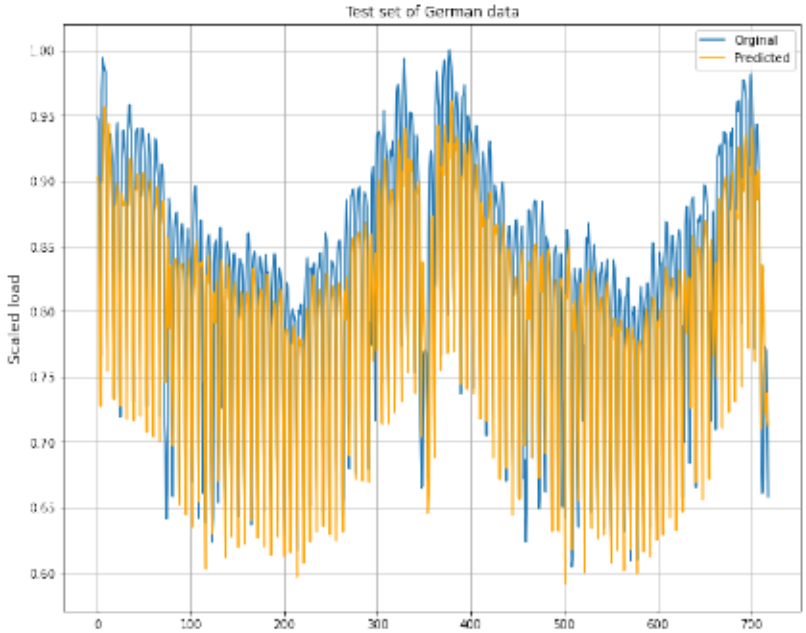

Figure 2.30: Actual and predicted results from CNN-LSTM, German data.

values, the short-term electric load forecasting accuracy of tested models in descending order is as follows: the PLCNet, LSTM-CNN, LSTM, ARIMA, linear regression, DNN, SVR and ETS.

Besides, it can be seen in the figures that PLCNet has performed far better than other models, especially in German data set. Since the Malaysian data is an hourly data, a lot of samples are available thus all the models can be trained well, while German data is a daily one, and with less number of samples all the model have been trained. This leads to let our model shows its power more with an accuracy of 91.18%. After that LSTM has performed well and its accuracy is 83.17%. However, the accuracy of PLCNet model for Malaysian data is the highest, too, 98.23%, but there is not any remarkable difference between the most accurate one and the second one which is LSTM-CNN and its accuracy is 97.49%.

Regarding the run time in the tables, linear regression and SVR are the fastest models in both German and Malaysian case. However, they are outperformed by deep learning models. Besides, ARIMA and ETS are two computational techniques which takes significant time for training. Even though all deep learning models in the tables took much time to be trained compared

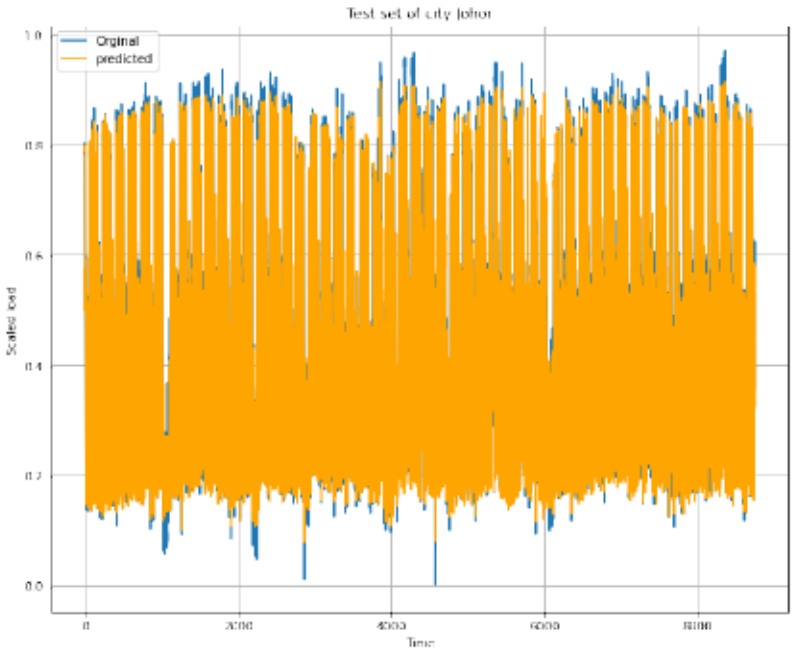

Figure 2.31: Actual and predicted results from proposed model, Malaysian data.

Model	Performance metrics		$R^2 score$	Runtime(s)
	RMSE	MAPE		
ARIMA	0.102	3.56	94.19%	451.12
ETS	0.36	8.81	90.06%	380.35
Linear Regression	0.092	2.335	95.50%	12.41
SVR	0.272	7.63	90.40%	10.23
DNN	0.128	3.62	95.38%	199.12
LSTM	0.097	3.11	96.63%	902.56
LSTM-CNN	0.053	2.43	97.49%	487.33
PLCNet	0.031	2.08	98.23%	92.47

Table 2.2: Models Performance for Malaysian data

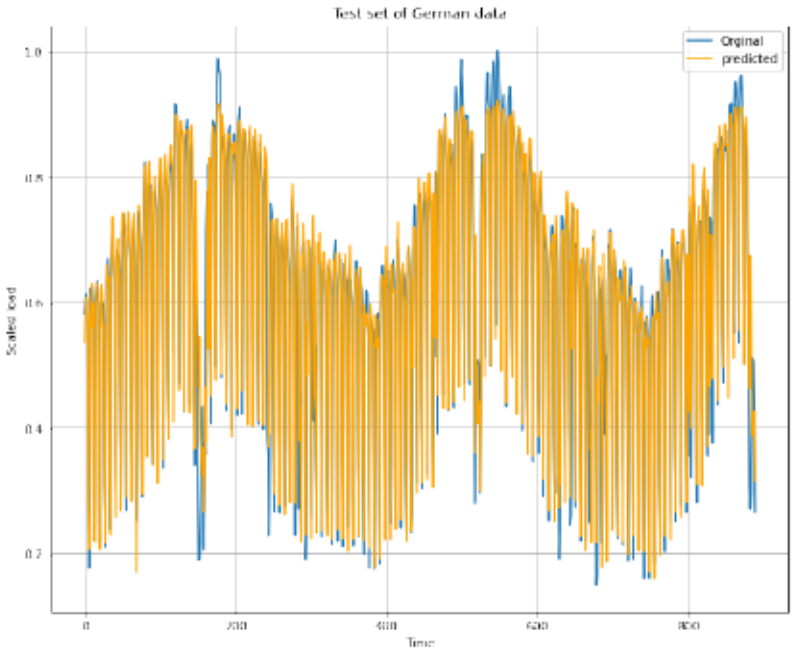

Figure 2.32: Actual and predicted results from proposed model, German data.

Model	Performance metrics		$R^2 score$	Runtime(s)
	RMSE	MAPE		
ARIMA	0.201	18.40	80.04%	179.89
ETS	0.316	33.63	70.1%	167.03
Linear Regression	0.214	19.12	79.86%	4.32
SVR	0.247	22.41	74.39%	3.11
DNN	0.25	26.47	73.47%	80.35
LSTM	0.197	13.20	83.17%	431.11
LSTM-CNN	0.207	15.02	79.75%	180.22
PLCNet	0.061	5.120	91.18%	65.34

Table 2.3: Models Performance for German data

Model	Runtime per epoch(s)	
	Malaysian data	German data
DNN	9.95	3.2
LSTM	4.61	1.87
LSTM-CNN	9.74	3.01
PLCNet	4.5	0.93

Table 2.4: The training time per epoch of deep learning models

to regression-based approaches, their acquired accuracy is acceptable. The difference between the training time in deep learning models depends on the number of epochs considered. Table 2.4 indicates the runtime per epoch of each deep learning model for both Malaysian and German data sets. According to the tables 2.2 and 2.3, LSTM has the highest runtime but the main reason is that this model needs more epochs to be trained and predict future load. It can be seen that in table 2.4 LSTM is faster than LSTM-CNN and DNN models in both data sets. However, PLCNet model results show that this model not only has the highest accuracy and lowest error amount, but also it is the fastest model between deep learning models where the runtime per epoch in Malaysian data is 4.5(s) and in German data is 0.93(s).

Therefore, it is proven that the novel hybrid STLF algorithm proposed in this thesis is practical and effective. Although LSTM has good performance when dealing with time series, its accuracy in the case of our data set which does not have large amount of samples, is not good enough. Therefore, the Vanila LSTM is not suitable for this kind of prediction. Finally, the experimental results show that the proposed hybrid network provides the best results in electricity load forecasting.

Statistical Analysis

A common approach to compare the performance of the machine learning models is that to use statistical methods to select the best one. This section aims to do a comparison between PLCNet and LSTM results since both of them achieved acceptable results in both German and Malaysian cases. In order to become able to compare these two models through statistical analysis, more available data is needed, so these two models (LSTM and PLCNet) were run 10 times to provide more result data. In terms of visualization, Fig. 2.33 and 2.34 show the results histogram of PLCNet and LSTM for

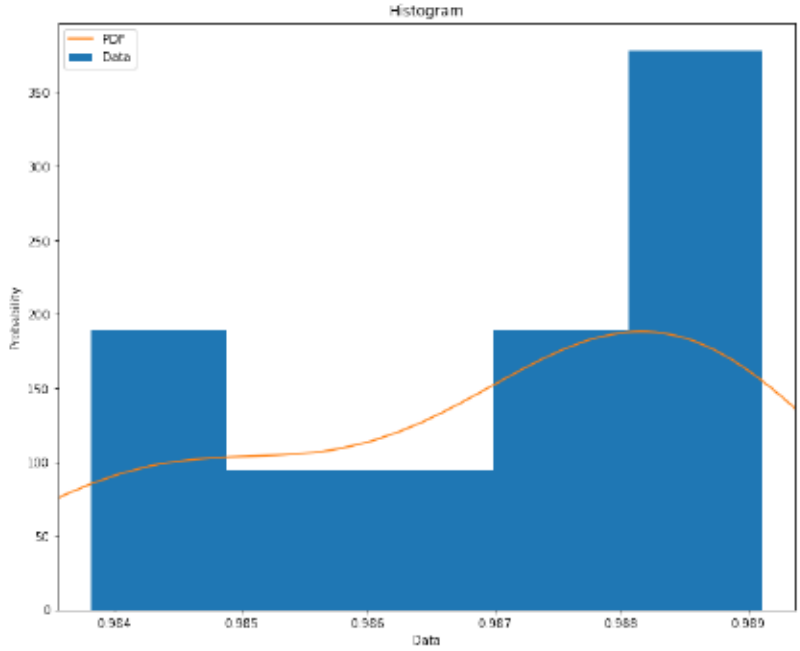

Figure 2.33: Histogram of PLCNet model for Malaysian data

Malaysian data set, respectively.

To perform statistical analysis, in this section t-test is used to understand the achieved results are just some stochastic results or they are trustful. This analysis works based on null hypothesis, and the null hypothesis is that two models (PLCNet and LSTM) are similar to each other and there is no difference between them while the alternative hypothesis is that two models perform differently. The considered significance level is 5%, therefore if the acquired P-value is less that 5%, the null hypothesis can be rejected and it can be conclude that the PLCNet model performs better than LSTM model. After carrying out some statistical analysis the obtained P-value is 0.0411 (or 4.11%) which is less than 5%.

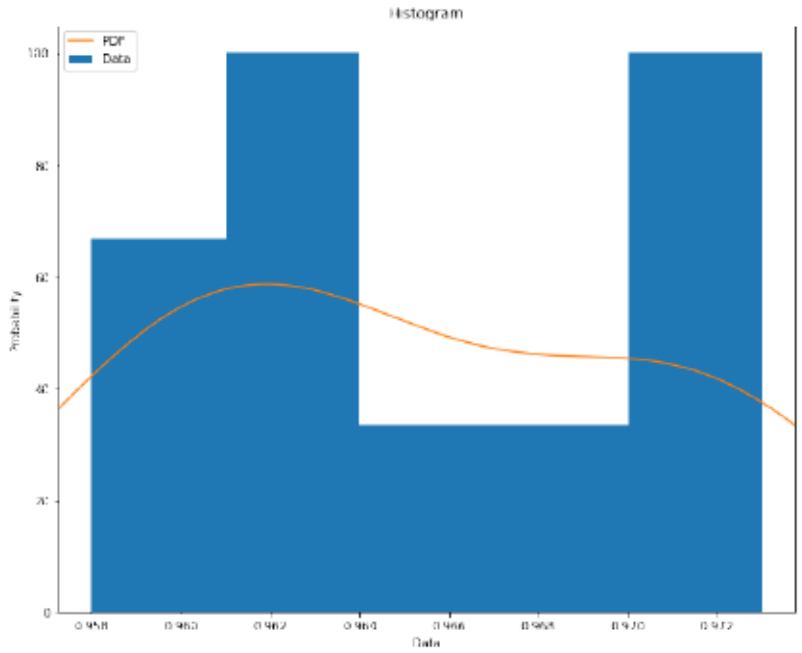

Figure 2.34: Histogram of LSTM model for Malaysian data.

Chapter 3

Validation of the proposed model

According to the results from last chapter, the proposed model has proven that it can carry out STLF task accurately. This chapter aims to challenge the proposed model in terms of the model's performance in different time horizons, working with single building data and evaluate the model's performance in the existence of weather variables. However, a modification in the architecture of the proposed model should be added to use the weather data as an input data.

3.1 Different Time Horizons

Previous sections discussed some machine learning techniques to predict next time step load data consumption. In other words, at time t they predicted the load data at time $t + 1$. Since the Malaysian data is an hourly data and German data is a daily one, all the models predicted next hour load of Malaysian data and next day load of German data. This section aims to challenge the proposed model in different horizons. In Malaysian case, the proposed model will predict next 24 hours, next 48 hours and next 10 days load data, and in German case it will predict next 7 days, next 10 days and next 30 days. In terms of evaluation, RMSE and R^2 scores are the two metrics used to evaluate the model's performance. Because of the existing soft computing errors the model is tested 5 times for each horizon and the average value is calculated. Same as mentioned before, the model uses year 2009 as training set and year 2010 as test set in Malaysian data, and years

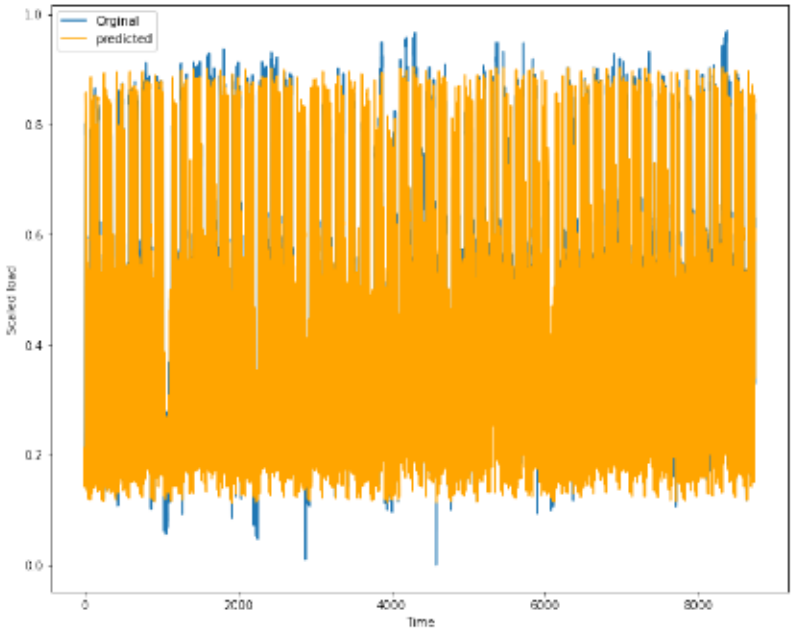

Figure 3.1: 1 day ahead results using the proposed model, Malaysian data

2012-2015 are used as training set and 2016-2017 as test set in German data set.

3.1.1 Malaysian Case

Since in previous sections, the prediction of next hour load data through the proposed model was discussed completely, this section studies next one day, two days and ten days load data in the following.

The Malaysian data is an hourly data, so to predict one day ahead load data next 24 time steps should be predicted and it leads to require a subtle modification in the model's architecture. The last layer of the model which is a dense layer will have 24 neurons to provide next 24 hours prediction. To predict one day ahead data, the model looks back to 72 hours ago data firstly to train the algorithm within data and then it will predict next 24 hours. Figure 3.1 shows the results of the prediction.

In order to forecast next two days load data, next 48 time steps should be predicted so another modification is needed to make the model able to forecast next days load data. Thus, the model will have 48 neurons in its last

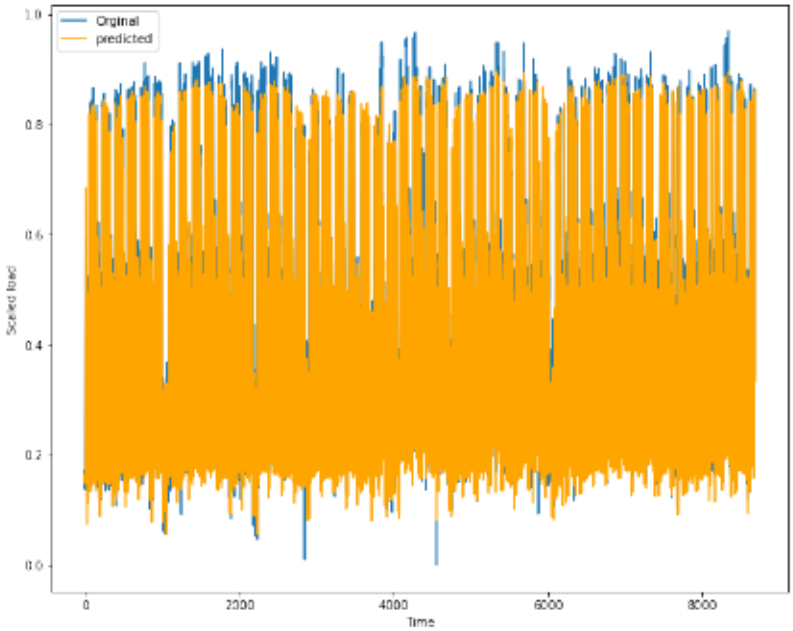

Figure 3.2: 2 days ahead results using the proposed model, Malaysian data

layer (which is dense layer). In terms of the training procedure, the model looks back to 4 days ago ($4days \times 24h$) to be trained and then it predicts next 2 days. Figure 3.2 illustrates the prediction and actual load data.

This section aims to predict next ten days load data which is a MTLF task but is a big challenge for the model. Since the samples have been recorded hourly in Malaysian data the model must predict 240 values ($10days \times 24h$) therefore there are 240 neurons in the last layer. In training process, the model looks back to ten days ago and predicts ten days ahead load data. According to figure 3.3, the results show that the proposed model has an acceptable performance in this task where the results are close to next hour, one day and two days ahead outputs.

Results

Tables 3.1 and 3.2 show the results of next days prediction. However, in order to have a comprehensive knowledge in terms of the model's performance, the results of the next hour prediction are added to these tables again.

Even though forecasting future load data in longer time horizons is a chal-

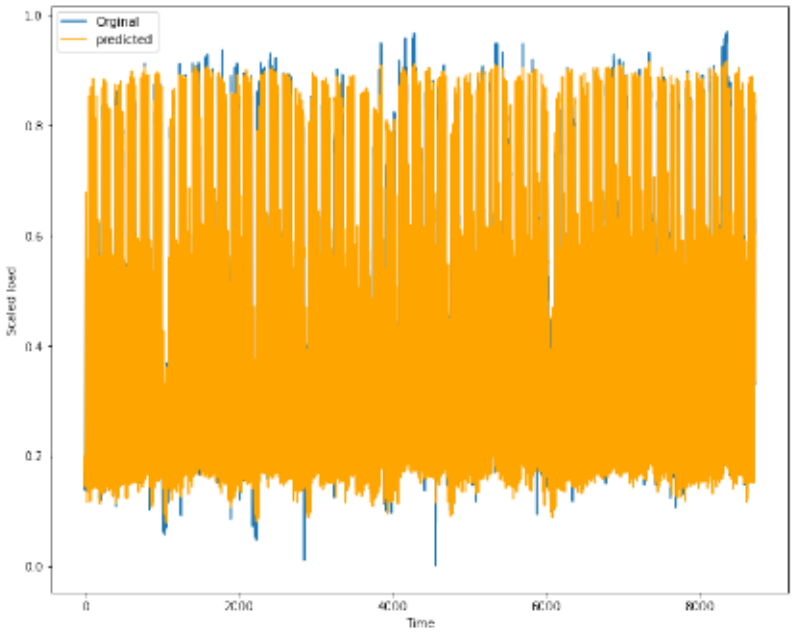

Figure 3.3: 10 days ahead results using the proposed model, Malaysian data

Model	Test-1	Test-2	Test-3	Test-4	Test-5	Average
Next hour	98.06%	98.10%	98.12%	98.19%	98.23%	98.14%
Next day	97.09%	97.60%	97.56%	97.63%	97.89%	97.55%
Next 2 days	96.80%	96.20%	96.31%	97.01%	96.88%	96.64%
Next 10 days	94.10%	93.89%	94.25%	94.35%	94.24%	94.16%

Table 3.1: The experimental results of Malaysian data in terms of R^2 score.

Model	Test-1	Test-2	Test-3	Test-4	Test-5	Average
Next hour	0.0341	0.0337	0.0335	0.0328	0.0316	0.0331
Next day	0.0412	0.0374	0.0382	0.0376	0.0355	0.0379
Next 2 days	0.0410	0.0413	0.0401	0.0387	0.0398	0.0401
Next 10 days	0.0609	0.0590	0.0582	0.05476	0.05477	0.0575

Table 3.2: The experimental results of Malaysian data in terms of RMES.

Model	Vanila LSTM		CNN-LSTM		Proposed model	
	RMSE	R^2 score	RMSE	R^2 score	RMSE	R^2 score
Next hour	0.097	96.63%	0.053	97.49%	0.033	98.14%
Next day	0.121	95.21%	0.069	96.62%	0.0379	97.55%
Next 2 days	0.189	92.65%	0.0782	94.31 %	0.0401	96.64%
Next 10 days	0.197	92.03 %	0.082	92.88%	0.0575	94.16%

Table 3.3: The comparison table for Malaysian data

lenging task, according to these tables, there is almost 4% difference between the accuracy of the next hour prediction and the next ten days prediction which is an acceptable difference and the model has a good performance in Malaysian case.

Likewise, in order to make sure the proposed model has a better performance rather than two other discussed deep learning models including Vanila LSTM and CNN-LSTM table 3.3 has been provided to compare the results of all these models in different time horizons in terms of RMSE and R^2 score.

3.1.2 German Case

The other case for model's evaluation is German data discussed in previous sections. Same as Malaysian data, the model is challenged in other horizons, but for this case it predicts next seven day, next ten days and next 30 days load data.

The model looks back to seven days ago to perform a seven days ahead prediction and to do this task it needs seven neurons in its last layer. Figure 3.4 shows the result of using the proposed model to forecast next seven days load data.

This section discusses the results of ten days ahead prediction. As the model is supposed to predict next ten days, there are 10 neurons in the last layer of the model. Besides, since forecasting next days data is a bit harder than next seven days, the model looks back to 10 days ago data to understand the algorithm within load series better. The results are shown in figure 3.5.

If the proposed model is being able to carry out a LTLF task too, it can be introduced as a well-performance tool in load forecasting applications. To evaluate the performance of the model in LTLF task, this section aims to

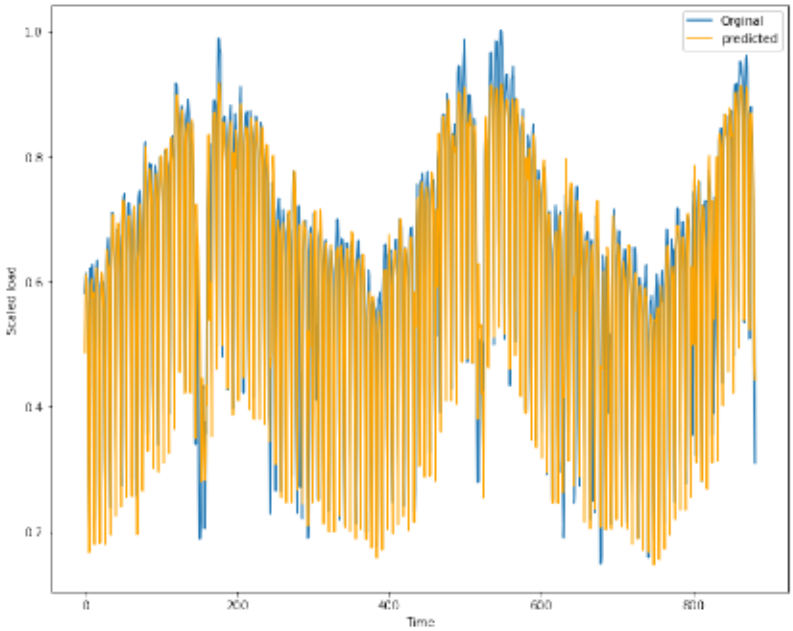

Figure 3.4: 7 days ahead results using the proposed model, German data

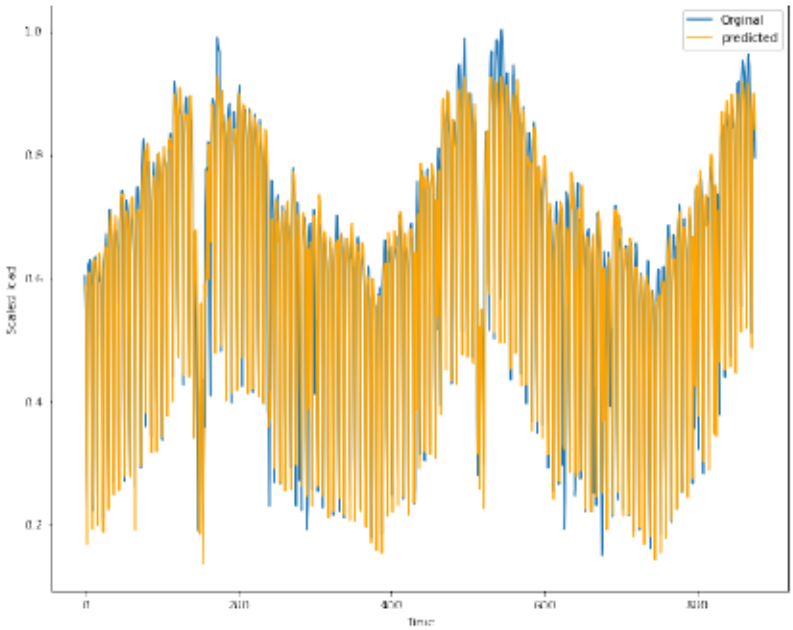

Figure 3.5: 10 days ahead results using the proposed model, German data

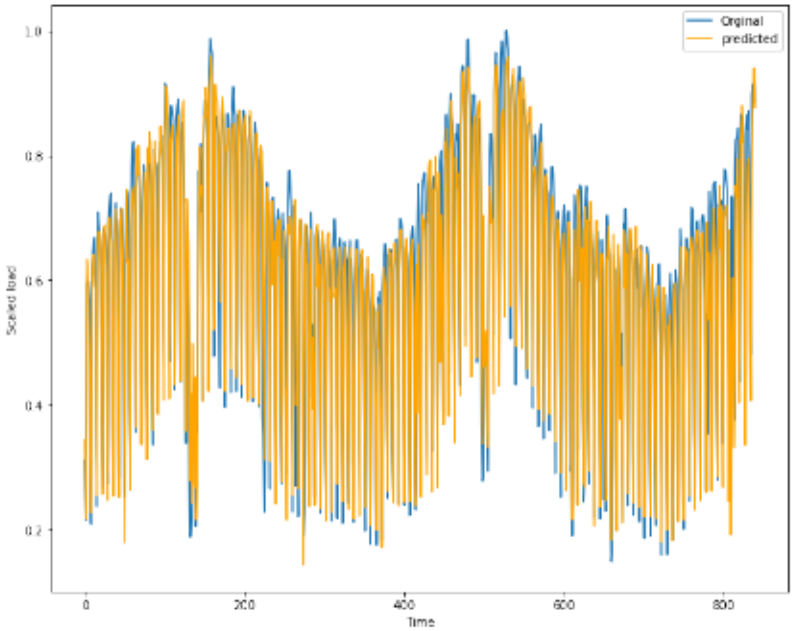

Figure 3.6: 30 days ahead results using the proposed model, German data

predict next 30 days German load data, and the results are shown in Fig. 3.6. Like previous sections, in terms of the model architecture there are 30 neurons in the last layer of the model to predict future load data.

Results

So far, the illustrative results of German data set have been shown, and in the following the numerical results are available. Besides, the one day ahead prediction results are available in tables 3.5 and 3.4 to demonstrate the comparison between different horizons for same data sets. These two tables indicate that there are not a lot of differences between one day ahead prediction and ten days ahead prediction, but it is big challenge for the model to predict next 30 days load series, because the average accuracy for the next day prediction is 91.31% while it is 82.49% for next 30 days prediction. The problem is that the German data is not a big data set and it has only 2186 recorded samples, so it is difficult for model to learn all the parameters well while it is looking back 30 previous steps and predict next 30 steps.

Same as the Malaysian data, a comparison table (see table 3.6) is provided

Model	Test-1	Test-2	Test-3	Test-4	Test-5	Average
Next day	90.76%	92.26%	91.18%	90.47%	91.88%	91.31%
Next 7 days	89.75%	89.92%	89.63%	88.82%	89.57%	89.53%
Next 10 days	89.02%	89.06%	89.59%	89.26%	88.99%	89.18%
Next 30 days	83.05%	83.25%	82.27%	81.84%	82.08%	82.49%

Table 3.4: The experimental results of German data in terms of R^2 score

Model	Test-1	Test-2	Test-3	Test-4	Test-5	Average
Next day	0.0659	0.0606	0.0622	0.0686	0.0612	0.0637
Next 7 days	0.0741	0.0734	0.0753	0.0787	0.0745	0.0752
Next 10 days	0.077	0.0768	0.0748	0.0733	0.0788	0.0761
Next 30 days	0.132	0.124	0.155	0.183	0.16	0.117

Table 3.5: The experimental results of German data in terms of RMES

for German results to prove that the proposed model not only has a better performance in one step ahead prediction, but also it can perform better than other deep learning models in different time horizons.

Model	Vanila LSTM		CNN-LSTM		Proposed model	
	RMSE	R^2 score	RMSE	R^2 score	RMSE	R^2 score
Next day	0.207	79.75%	0.197	83.17%	0.063	91.31%
Next 7 days	0.215	78.88%	0.201	80.87%	0.0752	89.53%
Next 10 days	0.231	78.02%	0.209	79.65 %	0.0761	89.18%
Next 30 days	0.312	74.14 %	0.279	76.88%	0.117	82.49%

Table 3.6: The comparison table for German data

3.2 Single Building Data

So far, all the technical parts have been evaluated by deploying highly aggregated data from two cities in Germany and Malaysia. Usually it is mentioned that highly aggregated data are not very complex data, so it is not difficult for machine learning models to learn from these data. But, applying a single building electric data can be a big challenge for the models to learn all the parameters since this type of data has unpredictable behaviours. In the following, a single building data set from a research lab in France is used. This data set includes 8784 samples which are hourly electric load data of this building in year 2016. Fig. 3.7 shows the illustrative plot of electric load data of whole data set. It can be seen that this plot does not follow a repeatable pattern and air conditioning causes this unexpectable behaviour. Decomposing the data into seasonal, trend and noise have been a helpful technique for better understanding. Fig. 3.8 shows part of French data decomposition. According to the plot, as the data consists of hourly recorded samples there is a 24 hour seasonality within data. Understanding the seasonality and overall trend of data set is a leading point and it can help to find out how the data must be processed.

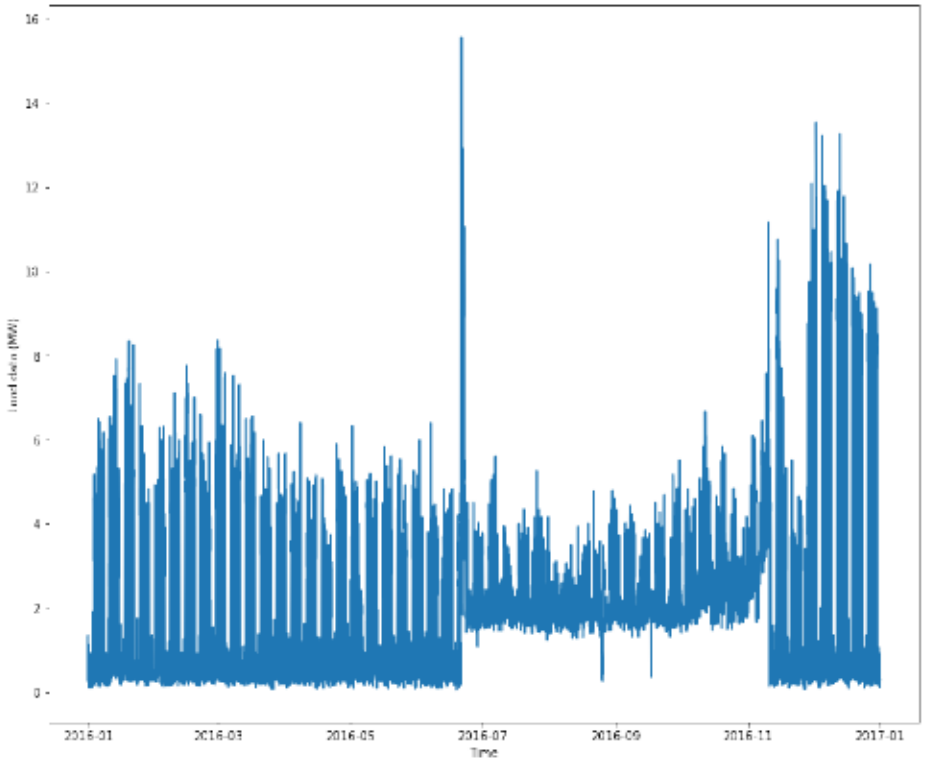

Figure 3.7: The electric load plot of French data in year 2016

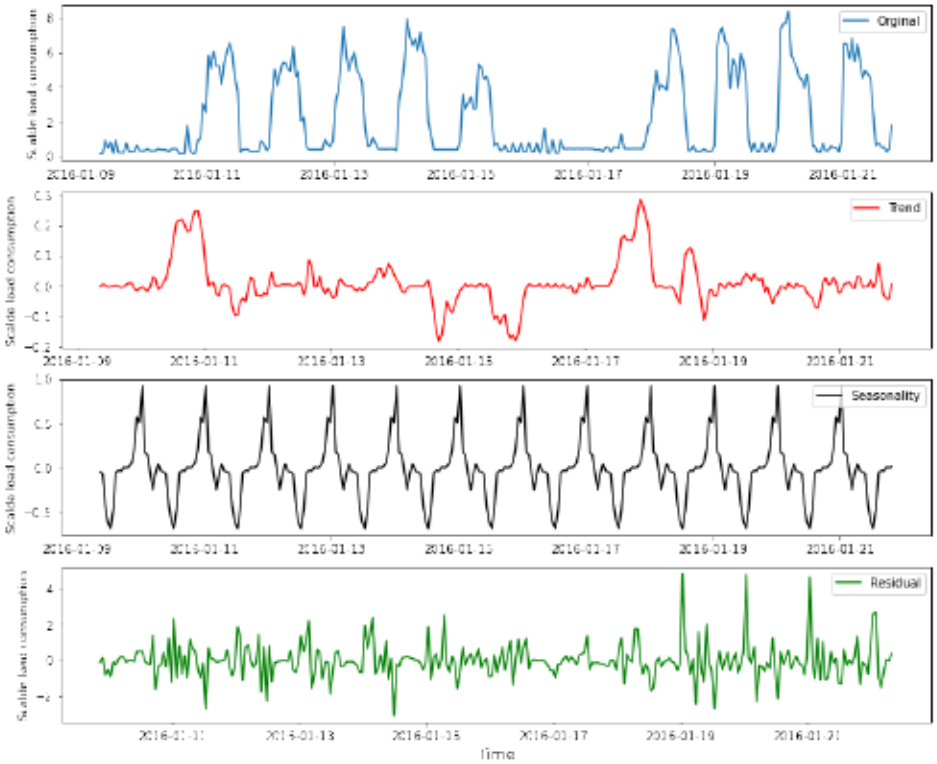

Figure 3.8: Part of decomposition plot of French data

Likewise, to have a comprehensive knowledge of this data, the boxplot of French load series is shown in Fig. 3.9.

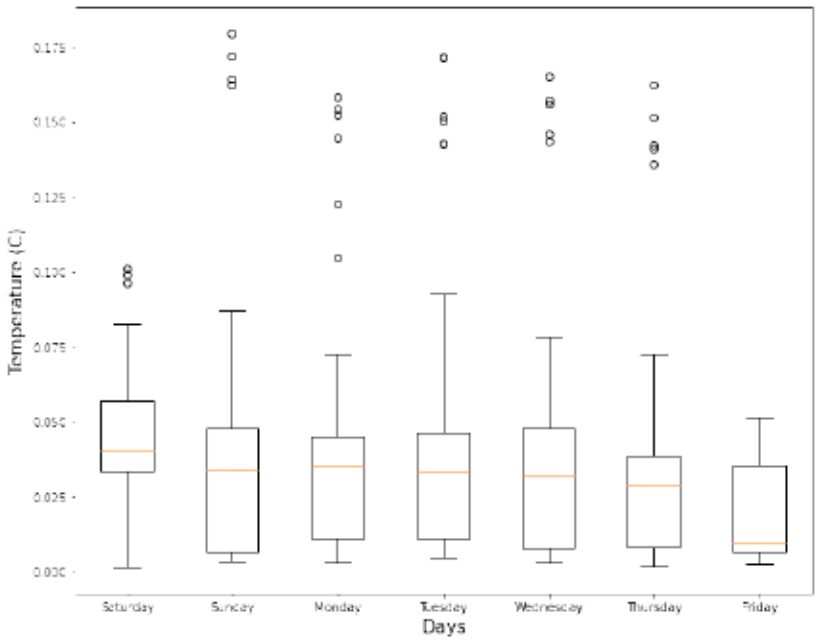

Figure 3.9: Load boxplot, French data

The main objective of this section is predicting next hour load data. Three different approaches are being used for this purpose; first training the model with 3 months data and then testing it on next three months for the whole year, and the second approach is to divide the data to two halves; and finally data is divided to 9 months as training set and 3 months as test test. In the second method, the model is trained by data associated to half of the year and then it is tested by the other half.

3.2.1 3-Month approach

In this approach, first model is trained by Jan-Mar load data and then is tested by Apr-Jun. This redundancy is applied for the whole year but the only important point is that after using Oct-Dec batch as training set, the model is tested by the Jan-Mar of year 2016 batch. All the illustrative results are shown in Fig. 3.10-3.13.

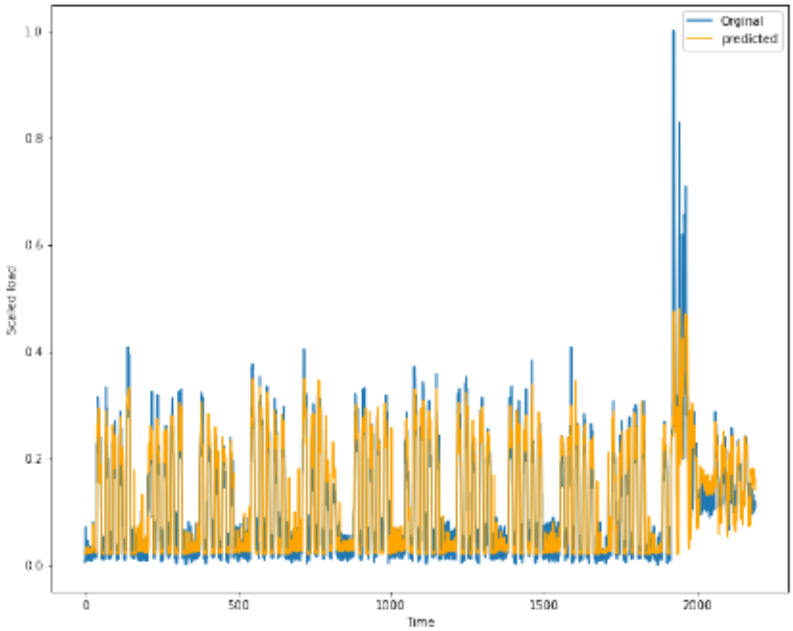

Figure 3.10: The results of first cycle of French data

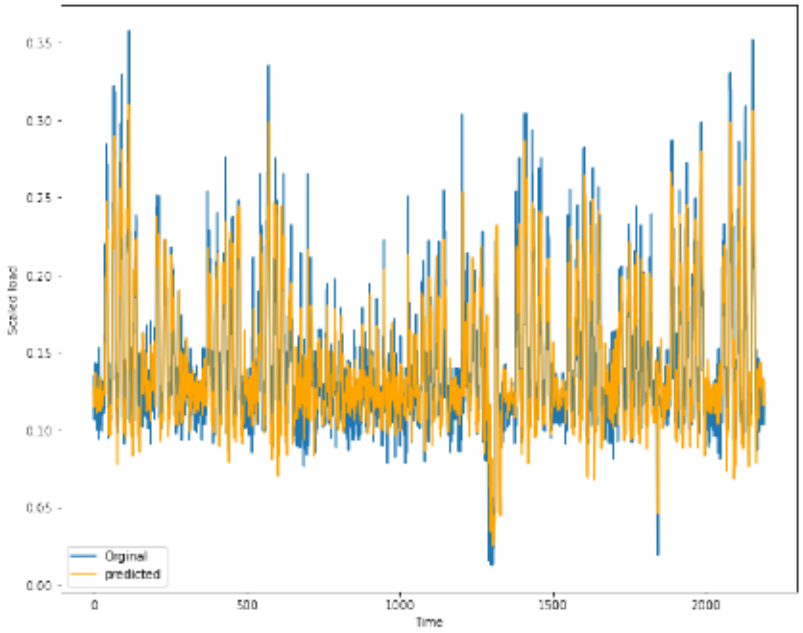

Figure 3.11: The results of second cycle of French data

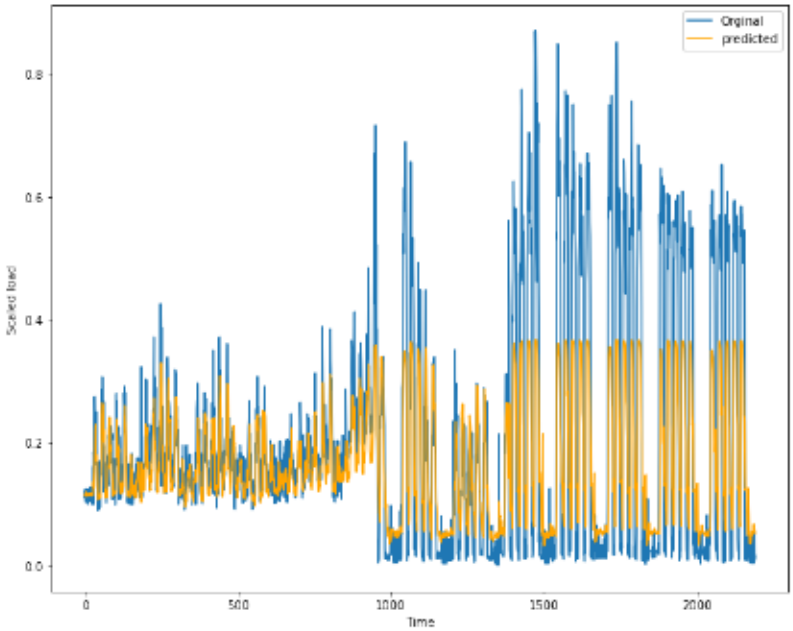

Figure 3.12: The results of third cycle of French data

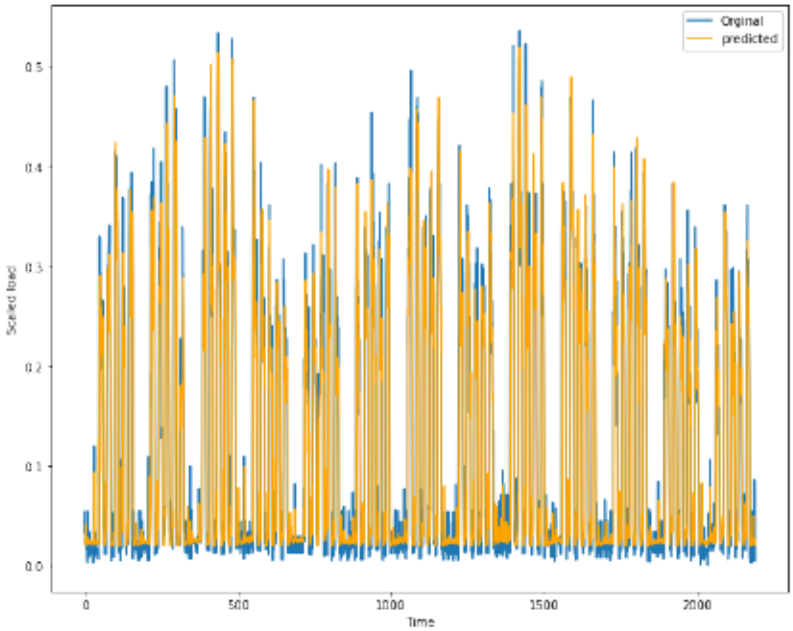

Figure 3.13: The results of fourth cycle of French data

As it can be seen in Fig. 3.12, the model has some problems to predict in this time region. The problem is that after training process, the model is faced with unpredictable behaviours in testing process, so predicting this unexpectable behaviours specially the peak values is tough. It can be referred that the 3-Month approach may come with some disadvantages and less accurate results.

3.2.2 6-Month approach

While in the last approach the model was trained and tested 4 times, there are only 2 times training and testing process in this approach. First the model is trained by the first half of the year (Jan-Jun) and then is tested by the other half (Jul-Dec). In the second try, the model is trained by the second half of the year and is tested by the first half. Fig. 3.14 and 3.15 illustrate the results of the two halves, respectively.

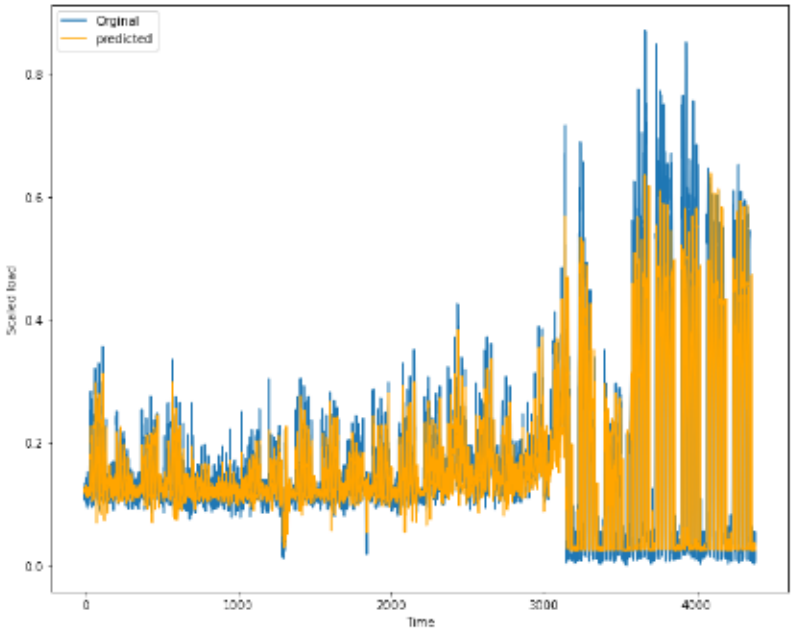

Figure 3.14: The results of first half of French data

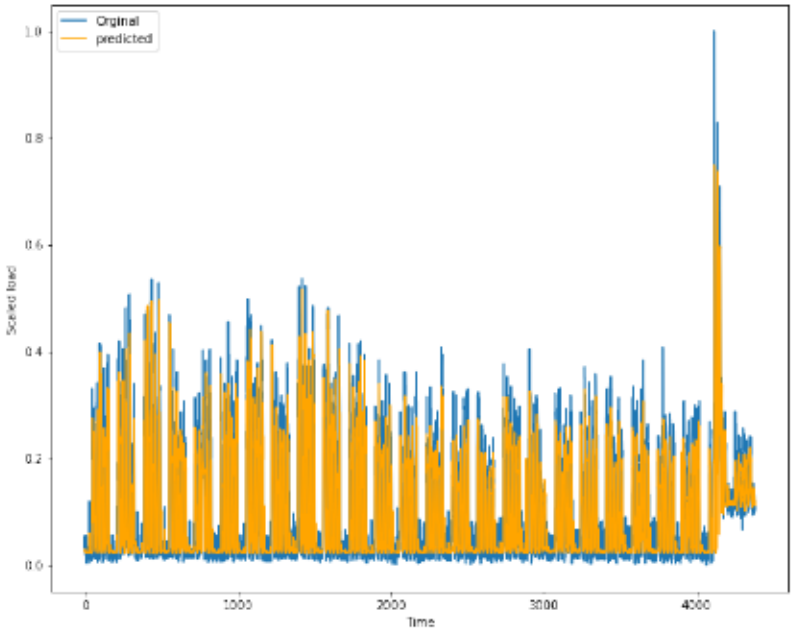

Figure 3.15: The results of second half of French data

3.2.3 9-Month approach

Same as 6-Month approach data is divided into two parts with a subtle difference in this section. The training set includes first 9 months (Jan-Sep) load data and test set is rest of data set (Oct-Dec) once, and in the second try, the training set consists of Apr-Dec and the model is tested by Jan-Mar data. This approach helps the model to understand the behaviour of the data set to have a better prediction. Fig. 3.16 and 3.17 show the illustrative results of this approach.

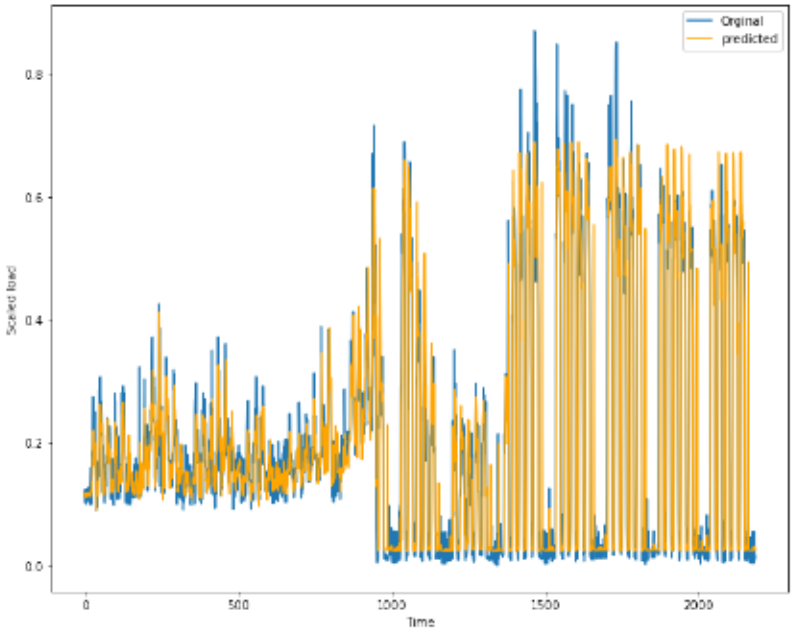

Figure 3.16: The results of first half of French data

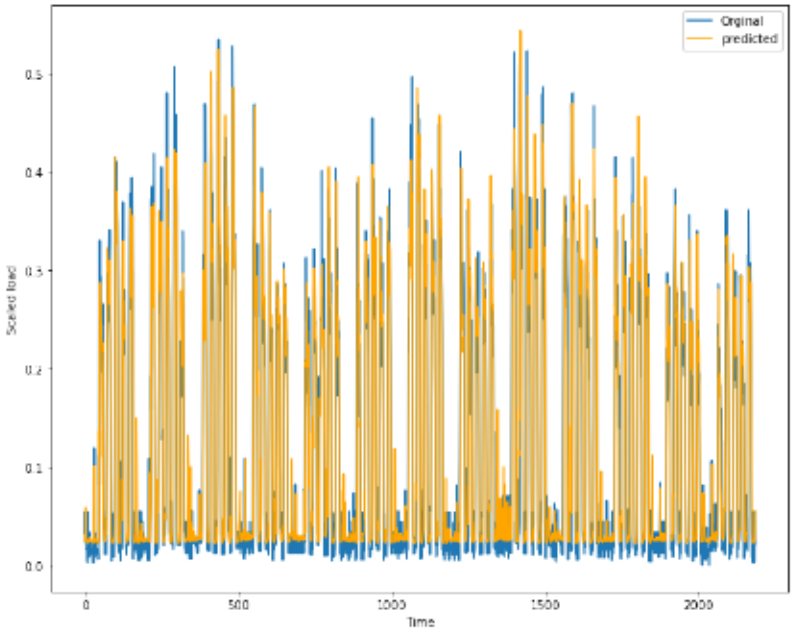

Figure 3.17: The results of second half of French data

3.2.4 Results

By taking look at Fig. 3.10-3.15 it can be concluded that the 6-Month approach has a better performance. Table 3.7 provides the numerical results to carry out a comparison between those approaches. According to the table, the average accuracy of 6-Month approach is 77.53% and this value from 3-Month approach is 68.46%. Likewise, RMSE of 6-Month approach and 3-Month approach are 0.0645 and 0.0713 in which regarding the nature of error a value closer to zero is a better result. Therefore, 6-Month approach has a better performance than 3-Month approach because in 6-Month approach the model has more information to be trained. However, when the results of 6-Month approach is compared to 9-Month approach it can be concluded that 6-Month approach is not good enough because the average RMSE and accuracy of 9-Month approach are 0.0547 and 84.84% which are far from better than to other approaches results. In total, since this data is just one year data and also there are a lot of unpredictable behaviours in this data set, the results show that the model is an appropriate tool for working with these complex data in STLF tasks.

Time	RMSE	R^2 Score
3-Month approach		
Jan-Mar	0.0496	80.9%
Apr-Jun	0.117	57.87%
Jul-Sep	0.1277	56.31%
Oct-Dec	0.063	78.76%
Average	0.0713	68.46%
6-Month approach		
Jan-Jun	0.0679	78.71%
Jul-Dec	0.0611	76.35%
Average	0.0645	77.53%
9-Month approach		
Jan-Sep	0.0585	83.3%
Apr-Dec	0.0509	86.39%
Average	0.0547	84.84%

Table 3.7: The experimental results of French data

3.3 Weather Data

According to [62] exogenous variables can affect the results of prediction in different ways. Usually, adding more variables to a model leads to increase the accuracy of the model, on the other hand if these variables are not added appropriately they may confuse the mode and therefore reduce the accuracy of the model. Regarding our cases, this section studies the impact of the weather information on two discussed data sets, Malaysian and French data. As it is discussed before, Malaysian is a highly aggregated data while the French one is a single building data. Adding temperature to these two different data sets and evaluating proposed model with both of them, is a good way to answer these questions; Does weather information improve the accuracy for aggregated as well as single building data? Is proposed model able to carry out an accurate prediction by adding temperature data? Adding temperature to a deep learning model is important or deep learning model can predict well without using it?

Before getting started to add temperature data to the model, it must be considered that the model needs to have some modification in its architecture to be able to process the weather data. Proposed model's architecture is

shown in 2.10, but it needs to be modified in order to use the weather data as input data. Thus, a fully connected path including dense and reshape layers are added to the model to process the weather data. In order to add temperature as an external variable to the model, some pre-processing is needed. Same as electricity data, temperature data should be scaled first and then are divided into batches with same length and size of electricity batches. For example, if the model is looking back to 72 previous time steps electricity data, for the weather data at the same time the model should look back previous 72 time steps and process the weather and electrical data simultaneously to forecast future load consumption. So the added path is called weather path, and model has another path in addition to LSTM and CNN paths. The processed data in LSTM, CNN and weather path are concatenated in a layer, and then these merged data are processed by two other dense layers to predict future load data. Fig. 3.18 illustrates the new architecture of the model for using weather data as an exogenous variable.

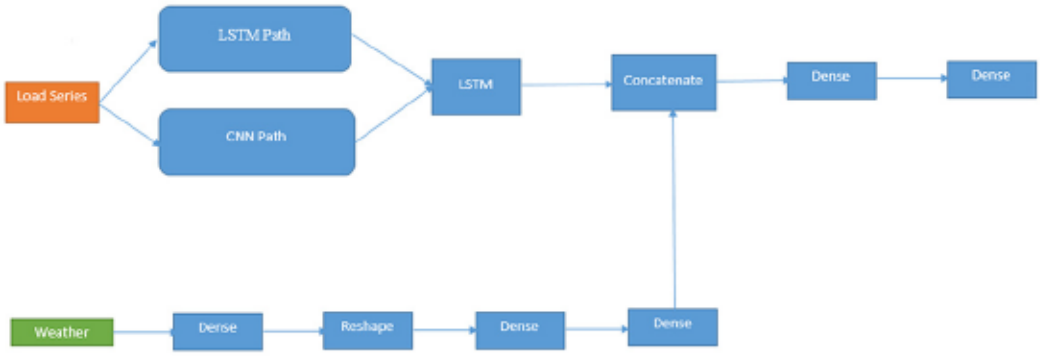

Figure 3.18: The new architecture of PLCNet model

3.3.1 Malaysian Case

Malaysian data includes load series and weather data, in which previous sections discussed only load data. In this section, weather data is added to the model to evaluate the accuracy of prediction in the existence of weather and find out can this information have positive effect or not. Same as load series, hourly weather samples for years 2009 and 2010 are available. In addition, Fig. 3.19 shows the boxplot of weather data for two years.

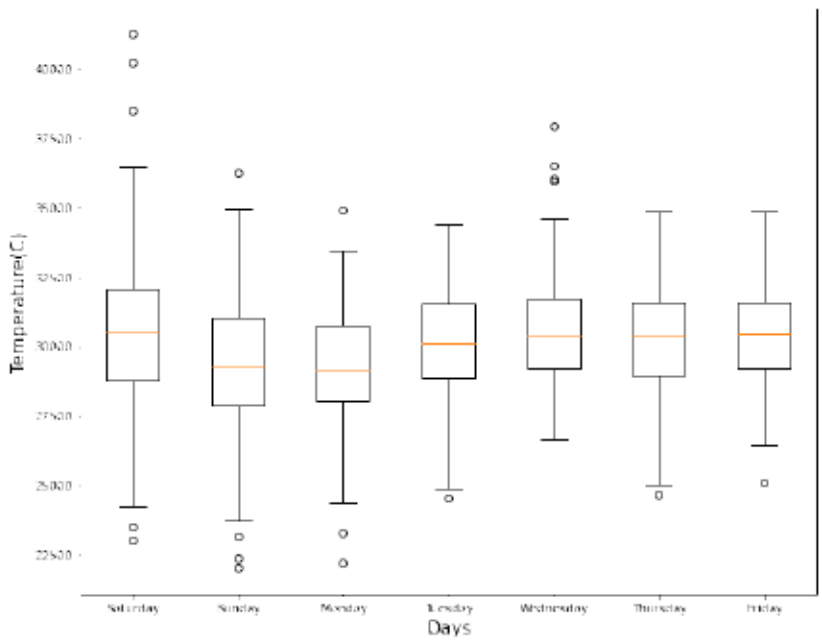

Figure 3.19: Weather boxplot, Malaysian data

In term of data preparation, same as the load series, weather data through equation (2.22) is scaled between 0 and 1 to be used as an exogenous variable. Fig. 3.20-3.23 show the actual and predicted results with using temperature and without using it in different time horizons.

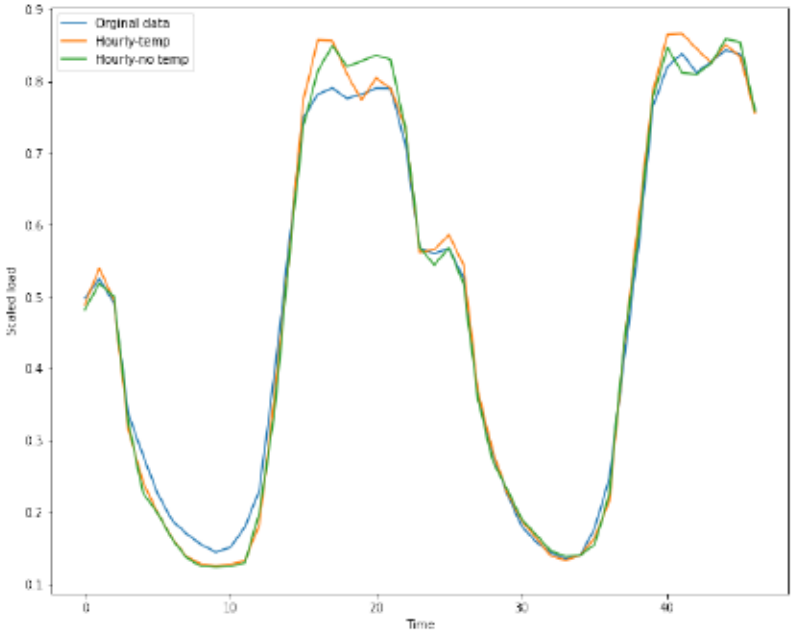

Figure 3.20: Predicted results for one hour ahead prediction, Malaysian data

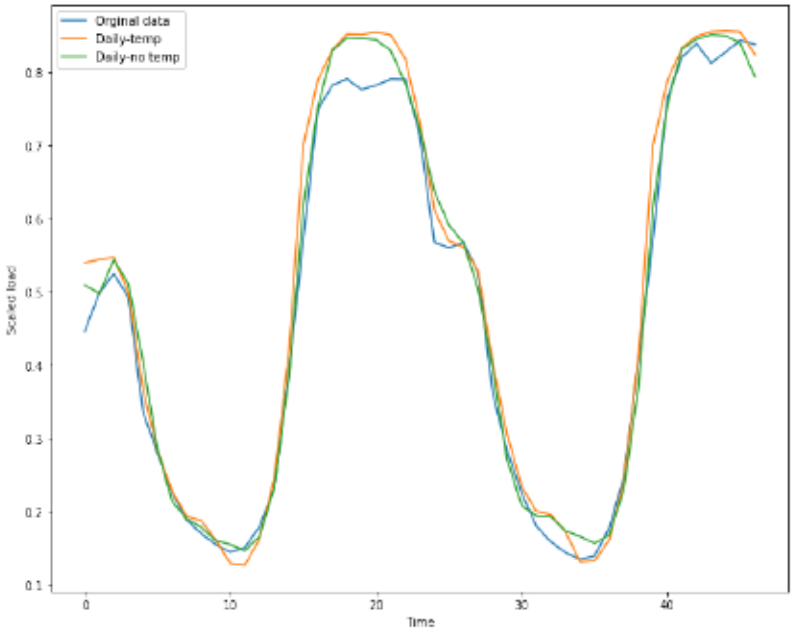

Figure 3.21: Predicted results for one day ahead prediction, Malaysian data

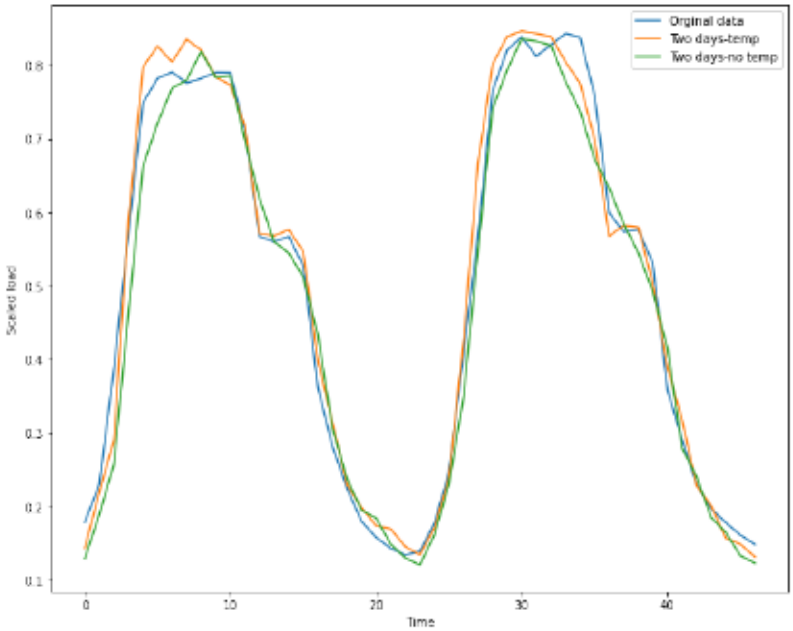

Figure 3.22: Predicted results for two days ahead prediction, Malaysian data

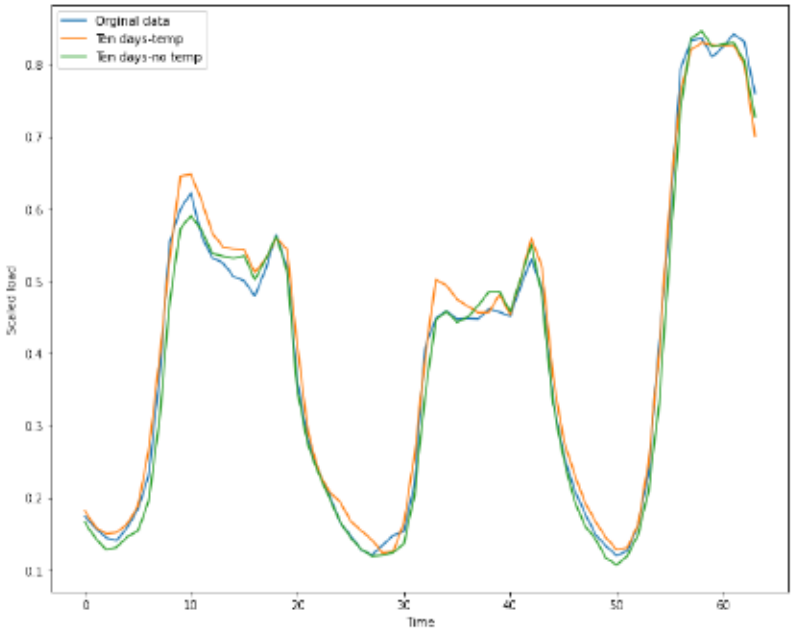

Figure 3.23: Predicted results for ten days ahead prediction, Malaysian data

3.3.2 French Case

French data is another hourly data which includes load and weather data of year 2016. This section focuses on discussing the weather information of French data. Fig. 3.24 shows the boxplot of weather data for whole year 2016. Scaled Weather data is used as an auxiliary data to improve the accuracy of proposed model (Fig. 3.18). Since in section 3.2, it has been concluded that the 9-month approach is better than 3-month approach, the model predicts future load using historical load and weather data through 9-month approach. Fig. 3.25 and 3.26 illustrate the prediction results for first and second cycles of the year, respectively.

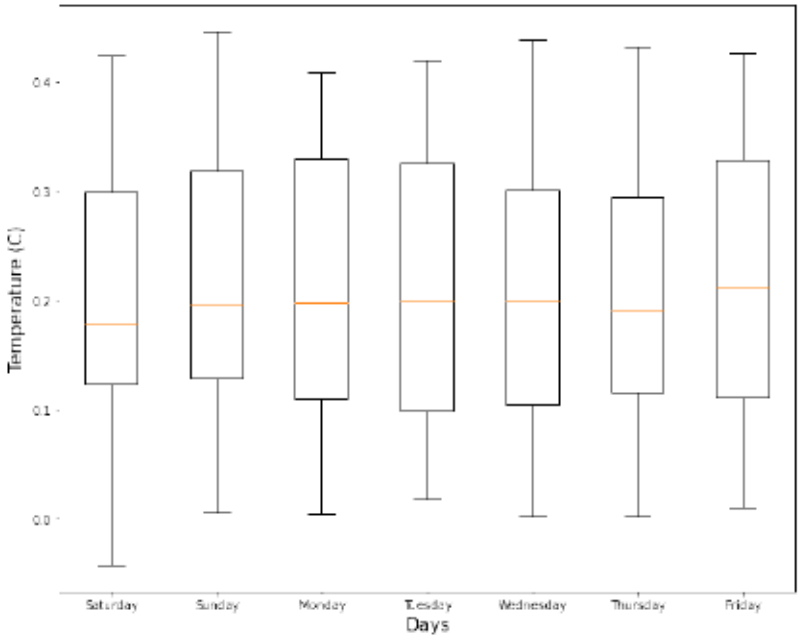

Figure 3.24: Weather boxplot, French data

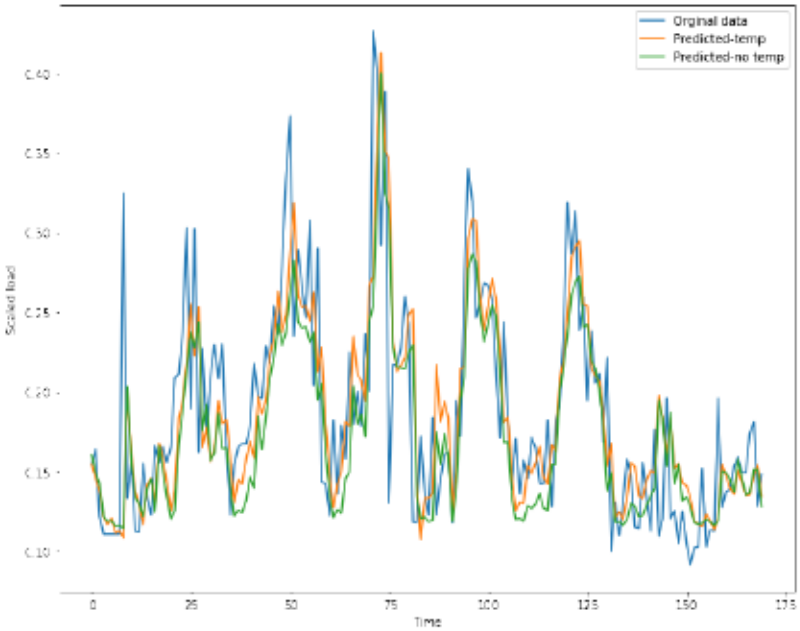

Figure 3.25: First cycle of the year prediction, French data

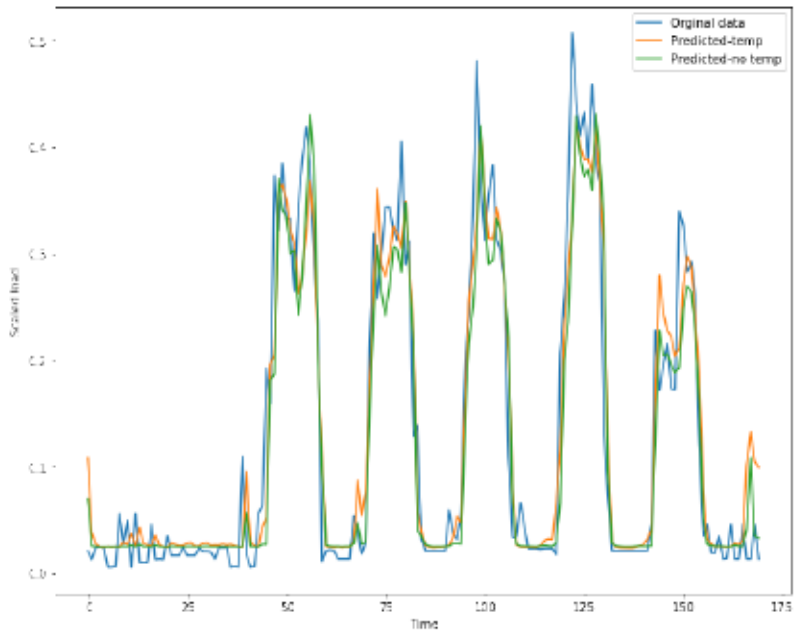

Figure 3.26: Second cycle of the year prediction, French data

3.3.3 Results

This section provides the numerical results of using temperature as an external variable, and compares the results before using them and after that. Correlation between historical load and weather data is an important concept which should be studied. The correlation between the information in Malaysian data is 0.56 while it is -0.08 in French data which explains well why the weather data has a better impact on the results in Malaysian data compared to French data. In fact, negative correlation is not the problem, the main problem is that the correlation is not high enough in French data. Table 3.8 and 3.9 indicates the Malaysian and French results in terms of RMSE and R^2 score as well as runtimes. By taking look at these numbers, it can be figured out that weather data improved accuracy and reduced the error which are the two primary goals of using temperature data.

Time	RMSE	R^2 Score	Runtime(s)
Next hour	0.0336	98.14%	92.47
Next hour with temp data	0.0289	99.15%	210.62
Next day	0.0379	97.55%	289.13
Next day with temp data	0.0302	98.2%	383.9
Next 2 days	0.0401	96.64%	602.11
Next 2 days with temp data	0.0385	97.26%	795.26
Next 10 days	0.0575	94.16%	726.54
Next 10 days with temp data	0.0492	96.05%	2821.08

Table 3.8: The experimental results of Malaysian data after adding temperature

As it can be assumed the weather data helps the model to be trained better, where it improved the model's performance in all the time horizons almost 1%, as well as it reduces all the $RMSE$ values remarkably. Whereas, adding temperature has some bad effect on training process runtime. According to the runtime column in table 3.8, runtime has been increased significantly, especially in 10 days ahead prediction where it launched from 726.54(s) to 2821.08(s) which is a big difference.

Time	RMSE	R^2 Score	Runtime(s)
First 9 months	0.0585	83.3%	181.2
First 9 months with temp data	0.0579	83.44%	260.18
Second 9 months	0.0509	86.39%	120.06
Second 9 months with temp data	0.0501	86.62%	165.73

Table 3.9: The experimental results of French data after adding temperature

According to the table, the accuracy is improved after adding temperature data where for both halves the accuracy is more than 80%. However, same as Malaysian data, adding temperature causes higher runtime. Finally, since the 9-Month approach and adding weather data have proven that they can achieve better results, these methods are used in French case study to predict one day ahead load data. To carry out this prediction, the model looks back to 3 days ago load and weather data and predicts next 24 hours load. The model is trained by Jan-Sep data and is tested by Oct-Dec information. The achieved results are as following; RMSE=0.0613 and Accuracy=

79.87% which are completely reasonable regarding the low number of training samples and the complexity within data. Fig. 3.27 shows the illustrative results.

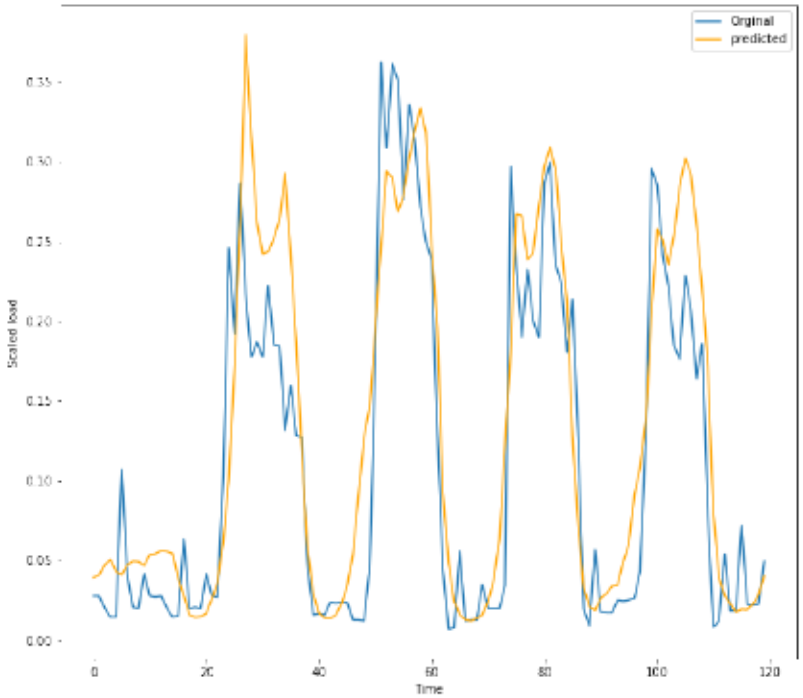

Figure 3.27: Part of next day prediction results, French data

In summary, after studying the results of both data sets it is obvious that the advantages of adding temperature to the model outweights its disadvantages, because higher accuracy is more important than speed of the model in load forecasting.

Chapter 4

Conclusion

With smart grids on the rise, the importance of short-term load forecasting highly increases. To predict the future load consumption, some factors such as weather can affect the results. The lack of future weather information is a challenging problem for load forecasting. In this paper, the previous consumption was used as a parameter to predict the load one step ahead. Some non-deep learning approaches like linear regression or ARIMA have proven that they are powerful tools for accurate load forecasting. However, regression-based approaches come with some disadvantages. In order to use these models, such as SVR and linear regression, lags are used as parameters through auto-correlation (AC) values. As the threshold value is subjective, the number of lags as parameters for regression-based models can be different. Fully connected networks also use the same approach as regression-based approach. Because of the reason that there is not a constant threshold to find those lags which are suitable to be used as variables, this procedure (finding lags through AC plot) may lead to higher errors. ARIMA and ETS also are two well-known time series analyses approaches. However, some parameters need to be tuned to work with these methods. This procedure needs numerous trials to find the best values for them. Furthermore, in time series methods data must be analyzed to find out if they are stationary or not. In contrast, LSTM can achieve reasonable results whether data is stationary or not. CNN-LSTM also is a hybrid model, which is used in various load forecasting studies. This proposed model achieves the best results between all the discussed models where the accuracy increase from 83.17% to 91.18% for load data a German case study. Likewise, for a Malaysian data set, the

obtained accuracy from the model is 98.23% which is very high for time series results and the RMSE is very low at 0.031. In summary, the proposed model improves the results remarkably for the German data set. Besides, while all the models have an acceptable performance in Malaysian data, the most accurate results come from the proposed model. In terms of runtime, the proposed model is faster than other deep learning models to train both German and Malaysian data. This improvement and highly accurate results as well as fast training process prove that this novel hybrid model is a good choice for STLF tasks. The proposed model was also evaluated in different horizons, and it performed better than other deep learning models. The accuracy of the proposed model in Malaysian experiment for different horizons is between 94.16% and 98.14%, and in German data it is between 82.49% and 91.31%, which are acceptable results compared to the other deep learning model's results. This model was also challenged to work with a complex data set from a research building in France including hourly load and weather data. Before adding temperature to the model and only using previous load series the accuracy of next hour load has achieved almost 77%, but after using weather data this accuracy was raised up to around 80%. Likewise, for Malaysian data case study this approach was studied, and the results were completely improved where for one hour ahead the achieved accuracy is more than 99%. All of aforementioned results are proofs that the proposed model is a powerful model for load forecasting.

Nowadays, the interest of using artificial neural networks for electric load forecasting is winning ground in research and industries, especially when deployed in IoT applications. According to the discussed results, deep learning models can be a good choice for IoT compared to other techniques, thus further work could be devoted to use deep learning models such as the proposed model in this paper for online load forecasting tasks.

List of References

[1] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid the new and improved power grid: A survey. IEEE Communications Surveys Tutorials, 14(4):944–980, Fourth 2012.

[2] H. Daki, A. El Hannani, A. Aqqal et al. Big Data management in smartgrid: concepts, requirements and implementation.J Big Data4, 13 (2017).https://doi.org/10.1186/s40537-017-0070-y

[3] S. Acharya, Y. Wi, and J. Lee. Short-term load forecasting for a single household based on convolution neural networks using data augmentation.Energies, 12:3560, 2019.

[4] S.P Schnaars and N. Meade. Book Reviews. International Journal of Forecasting, 2:241–242, 1986.

[5] V. Gupta. An Overview of Different Types of Load Forecasting Methods and the Factors Affecting the Factors Affecting the Load Forecasting.International Journal for Research in Applied Science and Engineering Technology, 5:729–733, 2017.

[6] D. Bunn, E.D Farmer. Comparative Models for Electrical Load Forecasting; Wiley: New York, NY, USA, 1986; p. 232.

[7] T. Hong, J. Wilson and J. Xie, "Long Term Probabilistic Load Forecasting and Normalization With Hourly Information," in IEEE Transactions on Smart Grid, vol. 5, no. 1, pp. 456-462, Jan. 2014, doi: 10.1109/TSG.2013.2274373.

[8] B. Yildiz, J. I. Bilbao, and A. B. Sproul. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renewable and Sustainable Energy Reviews, 73(March 2016):1104–1122, 2017.

[9] G. Dudek. Pattern-based local linear regression models for short-term load forecasting. Electric Power Systems Research, 130:139–147, 2016.

[10] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," in IEEE Transactions on Power Systems, vol. 16, no. 4, pp. 798-805, Nov. 2001, doi:10.1109/59.962429.

[11] A. Gasparin and C. Alippi. Deep Learning for Time Series Forecasting: The Electric Load Case. ArXiv, abs/1907.09207, 2019.

[12] W. He. Load Forecasting via Deep Neural Neural Networks. Procedia Computer Science, 122:308–314, 2017.

[13] C. Tian, J. Ma, C. Zhang, and P. Zhan. A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network. Energies, MDPI, Open Access Journal, 11, 2018.

[14] C. Gallicchio, A. Micheli. Deep Echo State Network (DeepESN): A Brief Survey. ArXiv abs/1712.04323 (2017).

[15] P.H Kuo, C.J Huang. A High Precision Artificial Neural Networks Model for Short-Term Energy Load Forecasting. Energies. 11. 213. 10.3390/en11010213.

[16] A. Liaw, M. Wiener. Classification and Regression by randomForest. R News 2002, 2, 18–22.

[17] S. Humeau, T.K Wijaya; M. Vasirani, K. Aberer. Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. In Proceedings of the 2013 Sustainable Internet and ICT for Sustainability, Palermo, Italy, 30–31 October 2013.

[18] C. Xia, J. Wang, K. McMenemy. Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks. Int. J. Electr. Power Energy Syst. 2010, 32, 743–750.

[19] F. Javed, N. Arshad, F. Wallin, I. Vassileva, E. Dahlquist. Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting. Applied Energy. 96. 150–160. 10.1016/j.apenergy.2012.02.027.

[20] L. Ekonomou, C.A. Christodoulou, V. Mladenov. A short-term load forecasting method using artificial neural networks and wavelet analysis. Int. J. Power Syst. 2016, 1, 64–68.

[21] G. Zhang, B.E. Patuwo, M.Y Hu. Forecasting with artificial neural networks: The state of the art. Int. J. Forecast. 1998, 14, 35–62.

[22] M.Q. Raza, A. Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renew. Sustain. Energy Rev. 2015, 50, 1352–1372.

[23] C. Gerwig. Short termload forecasting for residential buildings—An extensive literature review. Smart Innov. Syst. 2015, 39, 181–193.

[24] IEA (2017), World Energy Outlook 2017, IEA, Paris https://www.iea.org/reports/world-energy-outlook-2017

[25] H. Javedani, M. Hisyam, F. Gadelha, and P. Cândido de Lima e Silva. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. Energy, Elsevier, 175:365–377, 2019

[26] R. Sevlian, R. Rajagopal. Short Term Electricity Load Forecasting on Varying Levels of Aggregation. Trans. Power Syst..

[27] Y. Iwafune, Y. Yagita, T. Ikegami, K. Ogimoto. Short-term forecasting of residential building load for distributed energy management. In Proceedings of the 2014 IEEE International EnergynConference, Cavtat, Croatia, 13–16 May 2014; pp. 1197–1204.

[28] E.G. Kardakos, M.C. Alexiadis, S.I. Vagropoulos, C.K. Simoglou, P.N. Biskas, A.G Bakirtzis. Application of time series and artificial neural network models in short-term forecasting of PV power generation. In Proceedings of the 2013 48th International Universities' Power Engineering Conference, Dublin, Ireland, 2–5 September 2013; pp. 1–6.

[29] A. Veit, C. Goebel, R. Tidke, C. Doblander, H. Jacobsen. Household electricity demand forecasting: Benchmarking state-of-the-art method. In Proceedings of the 5th International Confrerence Future Energy Systems, Cambridge, UK, 11–13 June 2014; pp. 233–234.

[30] P. Ji, D. Xiong, P. Wang, and J. Chen. A Study on Exponential Smoothing Model For Load Forecasting.2012 Asia-Pacific Power and Energy Engineering Conference,(2):1-4, 2012

[31] T. Soubdhan, J. Ndong, H. Ould-Baba, M.T Do. A robust forecasting framework based on the Kalman filtering approach with a twofold parameter tuning procedure: Application to solar and photovoltaic prediction. Sol. Energy 2016, 131, 246–259.

[32] M.D. Reddy, N Vishali. Load Forecasting using Linear RegressionAnalysis in Time series model for RGUKT, R.K. Valley Campus HTFeeder.International Journal of Engineering ResearchTechnology(IJERT), 6(05):624–625, 2017

[33] S. Humeau, T.K. Wijaya, M. Vasirani, K. Aberer. Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. In Proceedings of the 2013 Sustainable Internet and ICT for Sustainability, Palermo, Italy, 30–31 October 2013.

[34] J.A.K. Suykens, J. Vandewalle. Least squares support vector machine classifiers. Neural Process. Lett. 1999, 9, 293–300.

[35] E. Ceperic, V. Ceperic, A. Baric. A Strategy for Short-Term Load Forecasting by Support Vector Regression Machines. IEEE Transactions on Power Systems, 28(4):1–9, 2013.

[36] B.J Chen, M.W Chang, C.J Lin. Load forecasting using support vector machines: a study on eunite competition 2001. IEEE Transactions on Power Systems, (4):1821–1830, 2004.

[37] B.W. White, F. Rosenblatt. Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. Am. J. Psychol. 1963, 76, 705.

[38] X. Ying. (2019). An Overview of Overfitting and its Solutions. Journal of Physics: Conference Series. 1168. 022022. 10.1088/1742-6596/1168/2/022022.

[39] D. P Kingma, J.L Ba. Adam : A method for Stochastic Optimization. CoRR, abs/1412.6980, 2015.

[40] S. Hochreiter, J. Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[41] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values. Sci. Rep. 2018, 8, 6085.

[42] S. Muzaffar, A. Afshari. Short-Term Load Forecasts Using LSTM Networks, Energy Procedia, Volume 158, 2019, Pages 2922-2927.

[43] B. Kwon, R. Park, K. Song. Short-Term Load Forecasting Based on Deep Neural Networks Using LSTM Layer. J. Electr. Eng. Technol. 15, 1501–1509 (2020).

[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In Proceedings of the IEEE, pages 2278–2324, 1998.

[45] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778, 2016.

[46] A. Krizhevsky, I. Sutskever, G.E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[47] R. Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[48] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N Dauphin. Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122, 2017.

[49] A. van den Oord, S. Dieleman, H. Zen, K.Simonyan, O.Vinyals, A.Graves, N.Kalchbrenner, A.Senior, K.Kavukcuoglu. Wavenet: A generative model for raw audio. In Arxiv, 2016.

[50] R. Yamashita, M. Nishio, R.K.G Do et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[51] B. Stephen, X. Tang, P.R Harvey, S. Galloway, K.I Jennett. Incorporating Practice Theory in Sub-Profile Models for Short term aggregated Residential Load Forecasting. IEEE Trans. Smart Grid 2017, 8, 1591–1598.

[52] J. Li, X. Li and D. He. A Directed Acyclic Graph Network Combined With CNN and LSTM for Remaining Useful Life Prediction, in IEEE Access, vol. 7, pp. 75464-75475, 2019, doi: 10.1109/ACCESS.2019.2919566.

[53] A. Z. Hinchi and M. Tkiouat, "Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network", Procedia Comput. Sci., vol. 127, pp. 123132, 2018.

[54] J. Sola, J. Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. Nuclear Science, IEEE Transactions on. 44. 1464 - 1468. 10.1109/23.589532.

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. Dropout: A SimpleWay to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 2014, 15, 1929–1958.

[56] H.S. Hippert, C.E. Pedreira, R.C. Souza. Neural networks for shorttermload forecasting: Areviewand evaluation. IEEE Trans. Power Syst. 2001, 16, 44–55.

[57] W. Kong, Z.Y. Dong, Y. Jia, D. Hill, Y. Xu, Y. Zhang. (2017). Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network. IEEE Transactions on Smart Grid. PP. 1-1. 10.1109/TSG.2017.2753802.

[58] S. Ruder. An overview of gradient descent optimization algorithms. (cite arxiv:1609.04747Comment: Added derivations of AdaMax and Nadam)

[59] A Novel Hybrid Deep Neural Network Model for Short-Term Electricity Price Forecasting, International Journal of Energy Research, pp.1-22, 2020.

[60] Multiple-Input Deep Convolutional Neural Network Model for Short-Term Photovoltaic Power Forecasting, IEEE Access, vol. 7, pp. 74822-74834, Jun. 2019.

[61] M.P Raju, A. Laxmi. IOT based Online Load Forecasting using Machine Learning Algorithms. Procedia Computer Science. 171. 551-560. 10.1016/j.procs.2020.04.059.

[62] J. Marcin. Methods of weather variables introduction into short-term electric load forecasting models - a review. PRZEGLAD ELEKTROTECHNICZNY. 1. 72-75. 10.15199/48.2017.04.18.