Statistical and data mining approach for the prediction of solar radiation

Wu, Ji

2013

Wu, J. (2013). Statistical and data mining approach for the prediction of solar radiation.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54851>

<https://doi.org/10.32657/10356/54851>

Statistical and Data Mining Approach for the Prediction of Solar Radiation

Wu Ji

School of Electrical & Electronic Engineering

**A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy**

2013

Acknowledgements

The first person whom I would like to express gratitude to is my supervisor, Dr Chan Chee Keong, from whom I learned a wealth of knowledge and skills during the length of my research. His patient guidance and warm encouragement have kept me motivated when I was facing difficulties.

Second, I would like to express special thanks to my senior, Dr. Li Ding, Dr. Tang Yi and Dr. Zhang Lei for their knowledge sharing, guidance and support, while we were formulating new research ideas. My appreciation then goes to Associate Professor Xiang Cheng, National University of Singapore, for the many insightful comments that he has shared with me. My appreciation also goes to my dear friend Mr. Wang Tao, for being there listen and console and for the suggestion you offered.

In the laboratory, I am equally grateful to Mr. Benny Chia and Miss Christina Wong for their help in solving problems related to the many experimental setups that I have built. They certainly have created a friendly environment for research students to work in.

I would like to thank my girlfriend Ms. Su Lin in particular. Without her love, encouragement and support, I would not have finished this thesis.

Last but not least, I would like to thank my parents for their everlasting love and patience, and all my friends who have shared joy with me during this period of struggle.

Table of Contents

Summary.....	vi
List of Figures	viii
List of Tables.....	xi
Chapter 1.Introduction.....	1
1.1 Background and Motivation	1
1.2 Contribution of Thesis	4
1.3 Organization of Thesis	5
Chapter 2.Literature Review.....	7
2.1 Empirical Solar Radiation Models.....	7
2.2 Time Series Analysis Algorithms.....	9
2.2.1 ARIMA models	9
2.2.2 Artificial Neural Networks	12
2.2.3 Support Vector Machine.....	19
2.2.4 Clustering Approaches of Time Series.....	25
2.3 Summary	31
Chapter 3.Prediction of Solar Radiation Time Series with TDNN and Statistical Approaches	34
3.1 Introduction.....	34
3.2 ARIMA Model	35
3.2.1 Autoregressive	36
3.2.2 Moving Average.....	36
3.2.3 Autoregressive and Moving Average	37
3.2.4 Autoregressive Integrated Moving Average	37
3.2.5 Building Model for Time Series.	37
3.3 TDNN Model	39
3.3.1 Basic back propagation algorithm	40
3.3.2 Quasi-Newton algorithm.....	41
3.3.3 Levenberg-Marquardt algorithm.....	42

3.3.4	Conjugate Gradient algorithm.....	42
3.4	Simulation Result.....	43
3.4.1	Data Source	43
3.4.2	Stationarity Test	43
3.4.3	Model Identification and Order Determination	45
3.4.4	Training of TDNN.....	46
3.4.5	The assessment of prediction performance and discussion	47
3.5	Summary	49
Chapter 4. Hybrid Model of ARMA and TDNN for Solar Radiation Time Series Prediction		51
4.1	Introduction.....	51
4.2	Detrending Models.....	52
4.2.1	Jain's model	52
4.2.2	Baig's model	53
4.2.3	S. Kaplanis' model	53
4.2.4	Al-Sadah's model.....	53
4.3	Simulation of Detrending Models.....	54
4.4	Prediction Models	60
4.4.1	ARMA Model	60
4.4.2	Time Delay Neural Network.....	63
4.4.3	Hybrid Model.....	65
4.5	Simulation of Prediction Models	65
4.6	Comparative Analysis	68
4.6.1	Comparative Analysis of Different Algorithms	68
4.6.2	Comparative Analysis of Prediction Performance of Different Daily Integral Solar Radiation	72
4.7	Summary	74
Chapter 5. A Novel Multi Model Framework for Solar Radiation Time Series Prediction		76
5.1	Introduction.....	76

5.2	The Proposed Methodology	77
5.2.1	Clustering	78
5.2.2	Prediction Phase	82
5.3	Simulation	84
5.3.1	Segmentation and Clustering	85
5.3.2	Training Prediction Models	85
5.3.3	Pattern Identification Procedure	86
5.3.4	Prediction Result	87
5.4	Comparative Analysis	89
5.4.1	Comparative Analysis of Different Algorithms	89
5.4.2	Comparative Analysis of Prediction Performance of Different Daily Integral Solar Radiation	91
5.5	Summary	93
Chapter 6. Genetic Algorithm with Multi Model Framework for Solar Radiation Time Series Prediction		95
6.1	Introduction	95
6.2	The Proposed Methodology	96
6.2.1	Segmentation and Clustering Methodology	96
6.2.2	Prediction Methodology	99
6.3	Simulation	102
6.3.1	Segment of Time Series	103
6.3.2	Training the Prediction Model	107
6.3.3	Experimental Result	109
6.4	Comparative Analysis	111
6.4.1	Comparative Analysis of Different Algorithms	111
6.4.2	Comparative Analysis of Prediction Performance of Different Daily Integral Solar Radiation	113
6.5	Summary	115
Chapter 7. Conclusion and Future Work		117

7.1	Conclusion	117
7.2	Future Work	118
Author's Publications.....		121
Bibliography		122

Summary

This thesis was initially proposed as part of Singapore National Research Foundation-Competitive Research Program (NRF-CRP) project entitled: “Combined Cycle Solar Energy Self-sustaining Membrane Distillation (MD) and Membrane Distillation Bioreactor (MDBR) Water Production and Recycling System” in 2009. As the proposed bioreactor is very sensitive to temperature changes, it is crucial to keep its temperature stable. As most of the energy is provided by a set of solar panels, ability to predict the solar radiation is therefore crucial in maintaining a stable temperature. Hence, there is a need to develop a system for accurate and consistent prediction of solar radiation.

We first present our research work on some fundamental issues encountered in solar radiation time series prediction. We examine how statistical models and data mining approaches can be used to conduct the prediction of solar time series. Our purpose is to find an approach which can model the complex nonlinear relationship lying in the time series data, effectively remove the noises and outliers and, on the other hand, provide us accurate and consistent prediction.

Auto regressive integrate moving average model (ARIMA), is a widely studied time series prediction approach. It has a very strong foundation in statistics. It gains great popularity because it can be used to model different kinds of time series with properly identified order. Besides, there is a widely accepted methodology to develop a model for a specific time series. However, the disadvantage of ARIMA is its inability to model time series that are nonlinear. Time delay neural network (TDNN) which is developed based on artificial neural network (ANN), is also studied in this thesis. TDNN is capable to capture the nonlinear relationship in the data set. But just like other data driven algorithms, it has the over-fitting problem. And when there are lots of noise data or outliers in the training data set, TDNN may also yield gross mistake.

As ARIMA and TDNN both have their own advantages, we propose a hybrid model which tries to combine them. This model uses ARIMA to model the linear component of time

series and TDNN is used to model the nonlinear component. We also use a novel detrending method to generate stationary series for ARIMA rather than the traditional differencing method. As we use the solar radiation time series in our experiment, several meteorology models are used as detrending model. Experimental result shows that our proposed hybrid model outperforms either ARIMA or TDNN.

To better improve the prediction performance of time series, we propose a novel multi-model framework, or MMF. In this framework, we assume that there are several different patterns occur repeatedly in the time series. Our purpose is to develop prediction model for every pattern and using proper model to predict the future value during the prediction phase. It is therefore necessary to segment the time series and then group the subsequences into different clusters. Initially we adopted a fixed length segmentation schema and find the optimal length for the subsequence through cross validation experiment. As TDNN is proved to be able to model nonlinear relationship of solar radiation, it is adopted as the prediction model. When predicting the future value of the time series, the pattern that the current time series belongs to is firstly identified. After that, the testing data is fed to the chosen model to conduct the prediction. The experimental result shows that the proposed MMF presents better prediction performance than other models.

Next, we sought to improve the clustering. Genetic algorithm and multi model framework or GAMMF is developed by combining genetic algorithm with MMF. We use a dynamic segmentation schema in GAMMF instead of the fixed length segmentation in MMF. To find the optimal segmentation schema, genetic algorithm is used to combine with K-means clustering algorithm. This segmentation schema is supposed to be able to achieve better clustering performance. Then TDNN is used to model different patterns. Support vector regression (SVR) is also used along with TDNN and serves as an additional prediction model. When none of the pattern is appropriate to describe the current time series, the SVR model will be used to conduct the prediction. The experiment result proves that GAMMF outperforms other prediction algorithms in both accuracy and consistency.

List of Figures

Figure 1.1 A daily solar radiation series of Singapore	2
Figure 2.1 The Structure of MLP	15
Figure 3.1 The structure of TDNN.....	41
Figure 3.2 monthly mean of solar radiation of Jan 2009	44
Figure 3.3 first order difference of monthly mean of solar radiation	44
Figure 3.4 the AIC of ARIMA with different order	45
Figure 3.5 the comparison of actual data and prediction of ARIMA.....	46
Figure 3.6 The comparison of actual data and prediction of TDNN	48
Figure 4.1 the monthly mean series of February 2009	54
Figure 4.2 the comparison of Jain's model and actual series of February 2009	55
Figure 4.3 the comparison of Baig's model and actual series of February 2009	55
Figure 4.4 the comparison of S. Kaplanis' model and actual series of February 2009	56
Figure 4.5 the comparison between Al-Sadah's model and the actual series of February 2009	57
Figure 4.6 the residual series after detrending by the 4 different models	58
Figure 4.7 The partial correlation of the residual series	61
Figure 4.8 The auto correlation of the residual series	62
Figure 4.9 The AIC of ARMA with different order	63
Figure 4.10 The architecture of TDNN.....	64
Figure 4.11 The solar radiation of February 2010	66
Figure 4.12 The actual and predicted series by ARMA (1,1)	67

Figure 4.13 The actual and predicted series by TDNN.....	67
Figure 4.14 The actual and predicted series by hybrid model	68
Figure 4.15. The RMSE of the prediction of different models on first half year's solar radiation	71
Figure 4.16. The RMSE of the prediction of different models on second half year's solar radiation.....	71
Figure 4.17 The comparison of prediction of ARMA & hybrid model in different level of daily integral solar radiation	73
Figure 4.18 The comparison of prediction of TDNN & hybrid model in different level of daily integral solar radiation	74
Figure 5.1 the flow of Multi-Model Framework	77
Figure 5.2 The prediction of MMF on July 28 th	88
Figure 5.3The prediction of MMF on Feb 2 nd	88
Figure 5.4 The comparison of prediction of MMF&ARMA in different level of daily integral solar radiation	92
Figure 5.5 The comparison of prediction of MMF&TDNN in different level of daily integral solar radiation	92
Figure 5.6 The comparison of prediction of MMF & hybrid model in different level of daily integral solar radiation	93
Figure 6.1 The flow chart of using GA to find the optimal segment	97
Figure 6.2 The silhouette index of different number of clusters.....	104
Figure 6.3 The interpolation of segment of different length.....	105
Figure 6.4 The average and optimal fitness value along with different numbers of generations	107
Figure 6.5 the flow chart of the experiment.....	109

Figure 6.6 The prediction of GAMMF on July 28 th	110
Figure 6.7 The prediction of GAMMF on Feb 2nd	110
Figure 6.8 The comparison of prediction of GAMMF & ARMA model in different level of daily integral solar radiation	114
Figure 6.9 The comparison of prediction of GAMMF & TDNN model in different level of daily integral solar radiation	114
Figure 6.10 the comparison of prediction of GAMMF & hybrid model in different level of daily integral solar radiation	115
Figure 6.11 the comparison of prediction of GAMMF & MMF model in different level of daily integral solar radiation	115

List of Tables

Table 3.1 The ADF test of monthly mean of daily solar radiation.....	44
Table 3.2 The ADF test of first order difference of monthly mean of daily solar radiation	45
Table 3.3 SMAPE of ARIMA and TDNN	48
Table 3.4 RMSE of ARIMA and TDNN.....	49
Table 4.1 The ADF test for the detrending models	58
Table 4.2 RMSE and NRMSE of different models.....	59
Table 4.3 Mean of SMAPE of different prediction models	70
Table 4.4 Variance of SMAPE of different prediction models	70
Table 4.5 RMSE of different prediction models	70
Table 4.6 Mean value of SMAPE of different level of daily integral solar radiation ..	73
Table 5.1 The best 5 parameter set of segment length and cluster number	85
Table 5.2 The monthly mean value of SMAPE of different window size	87
Table 5.3 Mean and variance of SMAPE of prediction of different month.....	89
Table 5.4 Mean of SMAPE of different prediction models	90
Table 5.5 variance of SMAPE of different prediction models.....	90
Table 5.6 RMSE of different prediction models	91
Table 5.7 Mean value of SMAPE of different level of daily integral solar radiation ..	92
.Table 6.1 Silhouette index value of different number of clusters	103
Table 6.2 The parameters used in the segment experiment	106
Table 6.3 MAPE of prediction of different month.....	110

Table 6.4 Mean of SMAPE of different prediction models	111
Table 6.5 Variance of SMAPE of different prediction models	112
Table 6.6 RMSE of different prediction models	112
Table 6.7 Mean value of SMAPE of different level of daily integral solar radiation	114

Chapter 1. Introduction

1.1 Background and Motivation

Because of the ever increasing energy demands and the danger of exhausting fossil energy supply, the renewable energy system is gathering increasing wider interest. The components of the renewable system are quite expensive. Thus, determining the size of such kind of renewable energy system according to the renewable energy load is necessary. Before constructing renewable energy systems, the renewable source must be analyzed in detail. Also, an accurate estimation of the energy source will largely improve the efficiency of whole system.

Solar radiation is the energy emitted by the sun. It affects almost all the living creatures on the earth and is a major source of clean energy. It is arguably one of the most widely used renewable energy source. Many applications for industrial or scientific usage are developed based on it. Solar radiation data record is one of the key input parameter for solar energy applications such as solar-thermal and photovoltaic systems.

The Singapore national research foundation-competitive research program (NRF-CRP) project: “Combined Cycle Solar Energy Self-sustaining Membrane Distillation (MD) and Membrane Distillation Bioreactor (MDBR) Water Production and Recycling System” is one such solar radiation based project. It aims to use bioreactor for sea water desalination. All the necessary energy required for the operation is derived from the sun as solar heat and electric. Through the use of solar energy, a 'zero-energy' MD/MDBR water production and recycling system will be built and it will have a tremendous impact on water production and recycling. This system requires less 'energy' to operate and is pollution-free. The use of solar thermal and electric energy allows us to desalinate sea water and recycle wastewater for 'free' with zero carbon emission. As the bioreactor is very sensitive to the solar radiation, an accurate prediction of solar radiation is required. However, none of the existed technologies could guarantee an accurate prediction. Thus,

finding an approach which could provide accurate and consistent prediction of solar radiation is of great concern.

Figure 1.1 A daily solar radiation series of Singapore

There are mainly two kinds of solar radiation prediction methodologies. One of them is using empirical model. It tries to use model which takes meteorological parameters as input to estimate the future value of solar radiation. There is a long development history on it and a lot of research work of this topic has been proposed. However, there are many disadvantages for the empirical model. First, most empirical models are developed to conduct long term prediction for solar radiation. Very few work of this kind focused on the short term solar radiation prediction which is important for solar radiation based project. Second, for some locations, the meteorological data is not available. Thus it is unable to utilize the empirical model to do prediction. Third, the empirical model is unable to detect the abnormal data and sudden change of the solar radiation data.

Another way to predict solar radiation is regarding it as time series generated by stochastic process. Figure 1.1 is the example of a daily solar radiation time series. Then a lot of sophisticated time series analysis approaches could be used to do the prediction. And this is the topic we will explore in this thesis.

Time series typically is a sequence of observations measured in successive instant. In most

cases, the order goes along time axis with equally spaced time interval. It can be represented by equation 1.1:

$$X = \{x_t, t = 1, \dots, N\} \quad 1.1$$

t is the index and N is number of observations. Time series analysis is fundamental to scientific, engineer and business endeavors. Researchers study the underlying principle of systems evolves through time and try to develop model which is capable in predicting or controlling them.

There are already a great number of methodologies developed for time series prediction. Traditional statistical approach on time series analysis has been developed for more than 50 years. The time series must be stationary before statistical approach is used to process it. However, this is not the case for most real world time series. Thus it is necessary to stationarilise the time series during the preprocess phase. Identifying proper model for a specific time series is another challenge in using statistical approach. Some information criteria can be used to address this problem.

Compared to the mature linear time series methodology, there are comparatively less research about non-linear time series for it is difficult to identify the statistical properties of complex nonlinear stochastic process. Some recently developed data mining approaches such as artificial neural network (ANN) could be used to model the underlying non-linear relationship of a time series without explicitly knowing its intrinsic nature. However, as the parameters of data mining algorithm are generated through the so called training process, they are very sensitive to the noise in training data set.

Pattern recognition and clustering of time series is another challenging topic of time series analysis. Like the clustering of other data, time series clustering is a procedure which group unlabeled data into several different clusters. The goal is also to maximize the similarity within certain cluster and the dissimilarity between different clusters. Many existed data mining approaches fail to provide satisfactory performance because time series data is dynamic and usually of high dimension. Thus, some dimension reduction approaches are used as preprocess strategy before clustering. After that, traditional

clustering algorithm could be used to do clustering.

There are also other interesting and challenging topics in time series analysis. In this thesis, we would like to use the approaches mentioned above in the prediction of solar radiation time series.

1.2 Contribution of Thesis

In this thesis, we have made some contributions to the prediction of solar radiation time series. They are as follows:

The comparison experiment of solar radiation time series prediction is conducted with both traditional statistical approach and computational intelligence approach. In the experiment, we find both approaches have unique characteristic and their advantages and disadvantages are further discussed.

To take the advantages of both statistical approach and computational intelligence approach, a combination schema is proposed. We work out a hybrid model which uses statistical model to predict the linear component of time series and use time-delay neural network (TDNN) to capture the non-linear relationship. Its performance is compared with individual ARMA model and TDNN model.

To further improve the prediction performance, we propose a novel multi-model framework (MMF). This framework starts with the theory that there are several different patterns lying in the stochastic time series. To extract these patterns, the solar radiation time series is firstly segmented into subsequences with fixed length. Clustering algorithm is used to group the subsequence into different clusters. And a unique prediction model is trained for each group to represent a specific pattern. When conducting prediction, the appropriate pattern is firstly identified and then according prediction model is used.

The accuracy of clustering is of great importance for MMF. Segmentation with fixed length is over-simplified. We use genetic algorithm combined with k-means to obtain the

optimal segmentation schema. Support vector regression (SVR) is also included in our prediction framework to further improve the prediction performance. The performance of the genetic algorithm combined with multi-model framework (GAMMF) is compared with previously studied model.

1.3 Organization of Thesis

The thesis is organized as following:

Chapter 1 introduces the background and motivation of the research, and summarizes the original contributions documented in the thesis.

Chapter 2 is literature review of the recent development of the approaches which can be used to predict solar radiation.

Chapter 3 introduces ARIMA and TDNN and uses them in the solar radiation prediction. Their performance is evaluated. The process of identifying appropriate ARIMA model for solar radiation time series is elaborated in detail.

Chapter 4 introduces several empirical daily solar radiation models. They are used to detrend the daily solar radiation series to generate a stationary series. A hybrid schema is used to take the advantage of both statistical algorithm and data mining approach. The performance of proposed hybrid model and other models are compared.

Chapter 5 proposes a novel multi-model framework (MMF). The multi-model framework is developed based on the theory that the solar radiation time series contains several different patterns. To explore the pattern lying in the solar radiation time series, segmentation and clustering approach are used. The performance of MMF and other models are compared.

Chapter 6 proposes a genetic algorithm combined with multi-model framework (GAMMF) which is developed based on previously discussed MMF. It adopts a dynamic

segmentation schema instead of segmentation of fixed length. Support vector regression is used combine with TDNN to further improve the prediction performance. Comparison experiment of GAMMF and previously discussed models is also presented in this chapter.

Chapter 7 summarizes all our findings. Some research topics for future investigation are suggested.

Chapter 2. Literature Review

As discussed in last chapter, there are basically two kinds of solar radiation prediction method. One is empirical model which estimate the solar radiation by using various meteorological parameters. These meteorological parameters contain extraterrestrial radiation, sunshine hour, maximum and mean temperature, relative humidity, altitude, latitude, total rain fall, cloudiness and evaporation. We also can regard the solar radiation data as random time series generated by stochastic process. Thus a lot of time series analysis approaches could be used to conduct prediction. Here we will present a review of the work of both categories.

2.1 Empirical Solar Radiation Models

The Angstrom-Prescott-Page model is one of the widely used empirical models. It is defined by:

$$\frac{H}{H_0} = a + b\left(\frac{S}{S_0}\right) \quad 2.1$$

In Equation 2.16, H is the monthly average daily global radiation while H_0 represents the average daily extraterrestrial radiation. a and b are empirical coefficients decided by the user. S is the day length while S_0 is the maximum possible sunshine duration [1].

The monthly average daily extraterrestrial solar radiation can be computed by Equation 2.17:

$$H_0 = \frac{24}{\pi} I_{gs} f [\cos \gamma \cos \delta \sin w_s + \frac{\pi}{180} w_s \sin \gamma \cos \delta] \quad 2.2$$

Where I_{gs} is solar constant of 1367 w/m^2 , f is eccentricity correction factor, γ is the latitude of the site while δ is the solar declination and w_s is the mean sunrise hour angle

for the given month. f , δ , w_s can be computed by Equation 2.18-2.20 [2]:

$$f = 1 + 0.33[\cos(\frac{360n}{365})] \quad 2.3$$

$$\delta = 23.45\sin[\frac{360(284+n)}{365}] \quad 2.4$$

$$w_s = \cos^{-1}(-\tan\gamma\tan\delta) \quad 2.5$$

n is the number of the day has pasted of the year. The maximum sunshine duration S_0 could be computed by:

$$S_0 = \frac{2}{15} w_s \quad 2.6$$

A lot of work has been done based on Angstrom-Prescott-Page model. Loche et al. provided empirical coefficients for the Angstrom-Prescott-Page model and make it suitable for France[3]. It is given as Equation 2.22:

$$\frac{H}{H_0} = 0.18 + 0.62(\frac{S}{S_0}) \quad 2.7$$

Dogniaux and Lemoine proposed a group of equations which use different equation for specific month [4]. Gopinathan suggested use the local altitude to calculate appropriate coefficient for the model [5]. Zabara correlated monthly a and b values of the model with monthly relative sunshine duration [6]. It is defined as follows:

$$a = 0.395 - 1.247\left(\frac{S}{S_0}\right) + 2.680\left(\frac{S}{S_0}\right)^2 - 1.674\left(\frac{S}{S_0}\right)^3 \quad 2.8$$

$$b = 0.395 + 1.384\left(\frac{S}{S_0}\right) - 3.249\left(\frac{S}{S_0}\right)^2 + 2.055\left(\frac{S}{S_0}\right)^3 \quad 2.9$$

There are also other works based on Angstrom-Prescott-Page model [7]. Jain proposed another useful model to estimate the solar radiation [8]. Baig improved Jain's model by bring about better fitting accuracy at the beginning and ending of the day [9]. S.Kaplanis also proposed a simple but effective and reliable model to estimate the hourly solar radiation [10].

2.2 Time Series Analysis Algorithms

Although a lot of work which use empirical model to predict solar radiation time series has been done, there are some fundamental disadvantages of this kind of models. The meteorological data which is required by empirical model sometimes is not available for developing countries. Another disadvantage is that its prediction target value is usually several hours ahead or even longer. However, short term and real-time prediction of the solar radiation is necessary for modern renewable energy system. The empirical solar radiation model is also unable to detect the abnormal or sudden change of solar radiation which is crucial for the control of solar radiation based system. To overcome these disadvantages, a lot of researchers take the solar radiation as time series and use time series analysis approaches to process it. Here we will review some of them.

2.2.1 ARIMA models

It was not until 1927, the notion of stochastic process was proposed by Yule [11]. It regarded any nature time series as one realization of a stochastic process. Based on this idea, the Auto-regressive (AR) and Moving Average (MA) were developed. Since then, a bunch of stochastic processes have been invented and were applied in time series analysis. Box and Jenkins integrated the existed stochastic processes and developed ARIMA model. Moreover, they also provided a step by step way of building up proper ARIMA for specific time series. This made ARIMA the most popular time series prediction model [11, 12].

ARIMA model is one of the most popular univariate time series model. Its popularity is based on the famous Box-Jenkins methodology which proves ARIMA is competent in modeling different types of time series. ARIMA is proved to be able to represent time series with different statistic property long time ago. However, there is not a method to determine the appropriate order for ARIMA model until Box & Jenkins used auto correlation coefficient and partial auto-correlation coefficient to study the order. They found there is certain statistic relationship between these coefficients and the order of AR and MV process. Some mathematic criterion is also used to identify the proper ARIMA. Akaike's information criterion (AIC) and Bayes information criterion (BIC) are two mostly used criteria. They are both based on the idea of minimize the prediction error one step ahead and put some penalty on over fitting. Akaike's final prediction error (FPE) was developed based on AIC for multi-step prediction. It takes the prediction error of several steps ahead and the over-fitting penalty into consideration. Other than the auto-correlation plot and statistic criterion, some research show sample trial and error experiment could sometime present better identification accuracy. In that process, the training data set is split into several groups and cross validation is performed on ARIMA model with different orders. The one gives best performance is regarded as the most appropriate one.

Another problem in the model building process is parameter estimation. Maximum likelihood estimation is the most popular estimation algorithm for its simplicity and efficiency. By regarding the model parameters as random variable, Zellner applied Bayesian analysis to obtain the distribution of them. Some recent finding supports that bias-corrected parameter estimation method is better than the least square error method [13].

A novel method which uses short-memory filter instead of long-memory filter to process time series is presented as ARARMA methodology by Parzen [14]. It requires one more parameter identification step than traditional ARIMA. It presents better performance than ARIMA especially in long term prediction. Another strategy to conduct the multi-steps prediction is using different AR model to conduct the prediction of time series with different horizon. But this would increase the difficulty of model identification and parameter estimation.

Another development based on ARIMA which aims to improve its one step prediction performance is automatic univariate ARIMA. It uses an expert system to generate and select appropriate ARIMA model and use it to conduct the prediction.

When the input of ARIMA model contains more than one dimension, we must adopt a vector form ARIMA model. The ARIMA with a vector input is called Multivariate ARIMA model or VARIMA model. Although this idea was firstly proposed during 1950s, there was little research on it until 1980s and 1990s when computer is used in the statistic model developing. VARIMA model is based on the assumptions of exogeneity and contemporaneous relationship, this brought about new challenges to the researchers [15]. Some researchers proved that smooth filters could improve the performance of VARIMA model. Because the smoothing filters could decrease the irregular fluctuations lying in the series. The VARIMA model could be used to conduct both long term and short term prediction theoretically. To explore the relation of prediction accuracy and prediction horizon, some researcher developed new statistic properties based on the cumulated-multi-step-ahead prediction and cumulated-multi-step-ahead error. However, only very few literature concern about how to decide the order of VARIMA model.

An auto regression model which takes vector as input parameter is VAR model. VAR could also be regarded as VARIMA without moving average process. VAR is mainly used in economic series analysis, and there are many different form of it. However, one disadvantage of VAR is its sensitivity to over-fitting. Although its within-sample-fitting is very good, its out-of-sample prediction performance sometimes is quite poor sometimes. To overcome this shortcoming, Litterman and other researchers imposed prior distribution on the parameters. They believed the parameters of VAR model are in line with certain behavior. For example they had a belief that the parameters of economic model are like random walk. This algorithm is called BVAR model [16].

Based on the previous algorithms, Engle and Granger developed a new concept which is called error correction models (ECMs). Many experiments showed the performance of ECMs is better than VAR and BVAR [17]. Especially in long term prediction, ECMs could outperform VAR and BVAR model. After introducing Bayesian approach in the parameter

estimation of ECMs, it could present even more accurate result than before.

Hokoi et al. used ARMA to develop stochastic model of hourly solar radiation time series for the summer months [18] After the transformation phase, it is found that an ARMA (3, 3) model presented the best performance. The autocorrelation function of the actual and the simulated data coincided at small time lags. However, at the large time lag, the proposed model cannot catch up with the actual data. Chowdhury and Rahman proposed an algorithm to use sub-hourly data for the forecasting of solar radiation [19]. They separate the data set into solar radiation of clear sky and cloudy transmissivities. For the clear sky the author assumed the existed empirical model is sufficient to conduct the prediction. They use ARMA model to conduct the prediction of solar radiation under cloudy day. They declared that these models were quite accurate except in the case of transitional changes in the cloud cover across the sun. A similar work is proposed by Mora-Lopez and Sidrach-de-Cardona [20]. Their work uses ARIMA (0, 1, 1) for the modeling of the global solar radiation as a percentage of the maximum available value.

2.2.2 Artificial Neural Networks

Artificial neural network (ANN) is a recently developed intelligence paradigm. The research on ANN is aggressively ongoing. ANN is inspired by the human brain system. It has been proved to be competent in many industrial applications [21]. One important application of ANN is time series prediction [22]. Compared to the traditional time series prediction algorithm, ANN provides more flexibility. It is driven by the training data and is able to capture the complex relationship lying in the data. It is especially suitable in dealing with problems which lack prior knowledge or theoretical model. That is the problem encountered by many researchers. Most of the time when dealing with time series; we have plenty of data of a time series process but no theoretical model. Thus, with ANN, we could model the relationship of a system without exploring its underlying law which generates the data. Another advantage of ANN is its ability in extracting unseen part of the training data. This is very useful in prediction because the basic principle of prediction is deducing the future behavior of a time series by learning its past behavior. However, the

performance of ANN largely relies upon the training data. Training data corrupted by noise might cause huge error of the ANN model. Thus, when adopting ANN to model time series, the quality of training data is of great importance.

It has been proved that ANN could approximate continuous function of any kinds to any desired accuracy. ANN could provide more flexibility than traditional statistical methodology. Traditional statistic model like ARIMA tend to use types of stochastic process to represent time series. It has limitation in the systematic complexity because it merely contains linear processes. Linear process is easy to understand and implement. But it is not capable in modeling non-linear process. However, most of the real world time series are non-linear. Thus non-linear model is needed to address these problems. Another difficulty of modeling non-linear mechanism is that we have little knowledge of its underlying law. Some model-driven non-linear methodologies have the problem that their predefined model is not general enough to capture all the features lying in the series. ANN is proved to be flexible enough to address this problem.

In this section, we bring about a brief introduction of ANN. Multi-layer feed forward network is the most popular type of ANN. It is composed by a number of nodes which just have basic processing ability. And these nodes are called neurons. The neurons take input from outside or from other neuron. After processing it through transfer function or simply activation, the processed data is passed to other neuron or output as final result. The Multi-layer feed forward artificial neuron network contains several layers of neurons. Although an individual neuron can process simple task only, the ANN can provide amazing modeling capability and is able to solve many kinds of problems.

There are ANNs with different kinds of structure. The most widely used models are multi-layer perceptron (MLP), Hopfield networks and Kohonen's self-organizing networks [23]. Hopfield network has quite unique structure without extra layer. Its nodes are completely interconnected. Its output may not be the result of transfer function, but rather, that stable states of iterative process. This unique structure enable it has associative memory. That means Hopfield network could recognize an example even it is distorted. Kohonen's self-organizing network is inspired by the self-organizing behavior of human

brain. The multi-layer perceptron (MLP) is the most popular used in prediction. Thus we will focus on the structure and usage of MLP in the rest of this section.

MLP contains several layers of neurons. The first layer is called input layer, which would take the input from outside. The last layer is the output layer which output the final result. Between the input and output layer, there is one or more intermediate layers which is often referred as hidden layers. The neurons of hidden layer are often interconnected with neurons of input layer. A basic structure of MLP is presented in Figure 2.1.

When dealing with prediction problem, the input data is usually the independent variable. The output is the dependent variable. The ANN estimates the relationship between the input and output. It can be given as:

$$y = f(x_1, x_2, \dots, x_p) \quad 2.10$$

In the equation x is the independent variable and y is the dependent variable. The ANN serves as traditional statistic model and tries to capture the relationship between input and output. Specifically for time series analysis, the input of the model is the past value of the series and the output is the future value. Thus, the equation could be rewrite as:

$$y_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-p}) \quad 2.11$$

The form of the equation is quite similar as autoregressive model. But ANN is more capable in modeling the underlying law lying in the data.

Figure 2.1 The Structure of MLP

Just as other learning algorithms, the ANN must learn from the training data set. It is called training process of ANN model. Training is actually determining the weight of the arcs which connect different neurons. The weight of arc is very important for ANN for it stores the knowledge learned from training data. ANN maps the nonlinear relationship from the input neurons to the output neurons through the connected arcs. During the training process, there is a desired output for each input data set. The desired output is called target value and the training process is called supervised training.

Before training the ANN model, the dimension of input layer should be firstly determined. The dimension of input layer is the dimension of vector which is defined by the user. Specifically in time series prediction, it is the number of past observation which is used as independent variable. Determining the number of input dimension is a challenging work. It is largely relied upon the user's experience. The available data is usually divided into training data set and validation data set. The training data set is used to generate the weight of arc and the validation data set is used to check the prediction capability of the trained ANN and check the generalization ability.

During the training process, the training data is input to ANN through input layer. It is activated and weighted by the hidden layer, then pass through a transfer function and the transformed value becomes the input value for the next layer. After transformation and processing of the hidden layer, the final value is output by the output layer. The output value is compared with the target value and error is computed. To minimize the error, we adopt certain strategy to change and update the weight of arc. The strategy is called training algorithm. Thus now the training process is converted to an optimizations problem. Many classical optimization algorithms could be used as training algorithm.

Assuming we have N observations of a time series in the training data set and we adopt input dimension of n to conduct one-step-ahead prediction. There are $N-n$ training patterns in the training data set. Using a_i to represent the output of the ANN and using sum of squared error as the error measure, we have the equation:

$$E = \frac{1}{2} \sum_{i=n+1}^N (y_i - a_i)^2 \quad 2.12$$

After training, E is minimized and weight is changed accordingly.

ANN has been used to address prediction task in many area and it provides encouraging performance. Lapedes and Farber used ANN to predict the time series which is generated by Glass-Mackey equation[24]. Their experiment result proves that the ANN is competent in modeling and predicting the deterministic chaotic time series. The traditional benchmark data set of time series prediction, sunspot series, has been used to study the performance of many traditional statistic models. The experiment conducted by former researcher proved that compare to the traditional prediction models, ANN could provide more accurate prediction. ANN has also been used to study the future trend of financial time series such as the business failure, bankruptcy, foreign exchange rate, stock prices [25]. Power electric load consumption prediction is another important application. An accurate prediction of electric load of the power grid will largely improve the efficiency of power system. Experiment shows even simple ANN could provide better prediction result than the traditional statistic model. A novel cascaded sub-network was proposed by Bacha and Meyer to predict the load of power system and they proved it could provide better

result than naive ANN [26]. Four-layer MLP was proposed by Srinivasan et al. To predict the hourly load of power system and its performance is encouraging [27]. ANN's capability in time series prediction was compared with other prediction model in the famous M-competition [28]. The data comes from different disciplines such as physics, business, economics and finance. Various experiments on that data set justified ANN's capability in time series prediction [29]. Other prediction problems like environmental temperature, commodity prices, macroeconomic indices, helicopter component loads, river flow, ozone level, wind pressure are also addressed with ANN model [30].

There are many researches of using ANN to find more accurate prediction than conventional ones. For example, Mohandes et al. [31], Sfetsos and Coonick [32], Dorylo et al. [31] have developed ANN of various kinds of structure for the solar radiation prediction. Soares et al. also proposed ANN based model to prediction hourly diffuse solar radiation [33]. Some new methods developed based on ANN is also proposed. Cao and Cao proposed wavelet transform preprocess approach combined with ANN to conduct the prediction [34]. Mellit et al. also proposed a wavelet network to predict solar radiation [35]. Cao and Lin proposed a novel model diagonal recurrent wavelet neural network (DRWNN) model which combines a recurrent neural network and a wavelet network with fuzzy technology [36]. It aims to forecast the hourly and daily global solar radiation. In this work, the wavelet basis is used as the transfer function instead of conventional nonlinear activation function. Their experiment result proved the DRWNN could provide better prediction performance.

Sometimes, other meteorological parameters are used combined with past solar radiation data in ANN to predict the future value. Alawi and Hinai proposed methodology which uses ANN to predict solar radiation in areas that are not covered by direct measurement instrumentation [37]. The input parameters of the ANN contain the location, the monthly average pressure, the temperature, the relative humidity, the wind speed and the sunshine duration. The prediction result is compared to the actual solar radiation data and the accuracy is very high.

Mohandes et al. used radial basis function (RBF) networks to model monthly mean daily

values of the global solar radiation and conduct the prediction [38]. The performance of RBF is compared with other algorithms like ANN and traditional regression models. According to the author, the RBF outperforms other algorithms in the experiment.

Dorvio et al. also used RBF and multi-layer perception (MLP) methods to estimate the solar radiation data [39]. The result shows both RBF and MLP performed well but the RBF consume less computation power and time.

Reddy and Ranjan proposed an ANN based model for estimating the monthly, daily and hourly values of the solar radiation [40]. The solar radiation data from 13 different stations spared over India is used to evaluate the performance of ANN. The experiment result shows ANN is a promising model for solar radiation prediction.

Sozen et al. used ANN to predict the solar potential of certain location with the geographical coordinate parameters [41]. The data from stations spared over Turkey were used in the experiment. The result shows ANN could present very accurate prediction which enable scientists to locate and design proper solar-energy systems and determine the appropriate solar technology to use.

Mellit et al. also proposed a hybrid model which combines ANN and Markov chain [42]. It is called ANN-MTM (Markov Transition Matrix). This model is used to conduct long term prediction for Algeria and it present very good performance.

Martin et al. used computational intelligence techniques contain ANN to estimate the solar radiation data from satellite images [43]. The models have been fitted to 15 different terrestrial stations in Span. The result of ANN is compared to a multivariate regression based model.

Rehman and Mohandes used ANN to predict daily solar radiation from air temperature and relative humidity at Saudi Abha [44]. Four years' data is used to train the ANN and one year data is used for testing and validating. Benghanem et al. compared the performance of different linear, polynomial regression models and MLP of estimating solar energy at Al-Madinah city [39]. The data used in this work are solar radiation, air temperature, sunshine duration and air temperature. ANN gives very good performance in

the experiment.

2.2.3 Support Vector Machine

Support vector machine (SVM) is another famous data mining algorithm. It is developed base on statistical learning theory which was proposed by Vapnik and Chervonenkis [45]. Like other data mining approaches, it is firstly used in pattern recognition and classification. The regression version of SVM, support vector regression (SVR) is used in time series prediction and regression analysis. Similar to ANNs, SVR requires no prior knowledge of the time series. It adopts the existed observation as training data to train a proper SVR model. In another words, user need not pre-define the model for a specific time series. The SVR will learn from the train data and adjust its own structure to accommodate it. In another words, it is a data driven algorithm.

Assume a time series $X(t)$ has n observations: $\{x_1, \dots, x_n\}$. $x_{n+\Delta}$ is the time series value to be predicted. $f(x)$ is prediction algorithm which tries to model the relationship between the past and future value of the time series.

$$f(x) = (w \cdot x) + b \quad 2.13$$

$$f(x) = (w \cdot \phi(x)) + b \quad 2.14$$

The above equations define the functions for both linear and non-linear relationship respectively.

As SVM tries to find an optimal plane to separate the input, there must be criterion to evaluate the fitness of the plane. First criterion is the error generated during the training process. The lower the error, the better the plane is. Second criterion is the “flatness” of the plane. Euclidean norm ($\|w\|^2$) is used to measure the flatness. Thus the overall goal is converted to an optimization process which could be defined as:

$$R_{reg}(f) = R_{emp}(f) + \frac{\gamma}{2} ||w||^2 \quad 2.15$$

$R_{reg}(f)$ is the regularized risk to be minimized. γ is the scale factor and used as penalty parameter of flatness. It controls the flatness to further solve the “over-fitting” problem.

A more commonly used loss function which is defined by Vapnik is:

$$\text{minimize } \frac{1}{2} ||w||^2 + C \sum_{i=1}^n L(y(i), f(x(i), w)) \quad 2.16$$

Where

$$L(y(i), f(x(i), w)) = \begin{cases} |y(i) - f(x(i), w)| - \varepsilon & \text{if } |y(i) - f(x(i), w)| > \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad 2.17$$

This equation is also referred as ε -insensitive loss function. ε in this equation is called tube size and is used to control the extend of approximation. C is the constant used to adjust the weight of estimation error in the loss function.

To solve the optimal weight and bias is and convex optimization problem. This problem is addressed by using Lagrange multipliers method. It could be converted to:

Maximize:

$$-\frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < x(i), x(j) > - \varepsilon \sum_{i=1}^N (\alpha_i - \alpha_i^*) + \sum_{i=1}^N \gamma(i)(\alpha_i - \alpha_i^*) \quad 2.18$$

$$\text{Subject to: } \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0; \alpha_i, \alpha_i^* \in (0, C) \quad 2.19$$

The so called support vector is the points on or outside the ε tube.

Support vector regression (SVR) is the regression of input parameter on high dimension

space. The SVR use the dot product of input data and the kernel function which satisfies Mercer's conditions could be expressed as:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \quad 2.20$$

There are many kernel functions satisfy Mercer's conditions such as polynomial and the well-known Gaussian. The selection of kernel function is of great importance for the performance of SVR. The most popular kernel is Gaussian kernel. However, there is no commonly accepted way of selecting kernel function. The identification of proper kernel function is a topic worth further studying.

The training algorithm is also very important for SVR. A lot of Quadratic Programming methods have been used to train SVR and the most popular one is the Sequential Minimization Optimization method. The detail of these training algorithms could be found in [46] [47].

SVR has been used in the prediction of different types of time series. Here we will present some of them. Solar radiation prediction is one of the most important applications.

Zeng and Qiao used SVM to conduct short term prediction of solar radiation [48]. They proposed a least-square support vector machine based model for short-term solar radiation prediction. The input for their model is the historical data and a novel 2D form is used for the input data. The experiment result shows SVM outperforms AR and ANN significantly.

Chen et al. also used SVM to estimate the solar radiation of Chongqing, China [49]. They used the data from the observatory station in Chongqing to train SVM and predict the monthly solar radiation data. The performance of SVM is compared to some empirical temperature-based models. Result shows SVM could present a better prediction performance.

SVR has also been used to predict other environmental parameters which related to solar radiation. For example, Lu et al. proposed their work which uses SVR to forecasting air quality [50]. The input data for their model was respirable suspend particles (RSP) and

nitrogen oxides, etc. The SVR model is used to predict the value one week ahead. SMO training strategy and Gaussian kernel function is adopted for their model. The experiment result shows SVR outperforms RBF network and other regression models.

Wang et al. extended Lu's work and compare an SVR approach to an adaptive radial basis function (ARBF) network which uses principle component analysis (PCA) [51]. They tried to predict respirable suspended particulate (RSP) concentrations three days ahead. The parameters for SVR were determined by cross-validation experiment which adopted MAE and RMSE as measurement. According to their experiment, SVR outperforms ARBF.

Trafalis et al. used both SVR and LS-SVR to predict rainfall and the presence of rain using WSR-88D radar data [52]. During the prediction, the LS-SVR uses a polynomial kernel and the SVR uses a Gaussian RBF kernel. They found SVR outperforms LS-SVR slightly in detecting the rainfall in a geographic grid. But LS-SVR outperforms SVR in prediction. The author stated that using LS-SVR instead of SVM will lose the sparseness quality of the representation of the solution. Using LS-SVR to solve an optimization problem is essentially a matrix calculation.

Prem and Srinivasa Raghavan also used SVR in the Network Weather Services which is a grid of computational nodes used for weather prediction [53]. Their goal is to optimize the parameters of the network to improve the performance of the whole system. By accurately predicting the need for the resource needed, the system could operate more efficiently and provide better prediction performance. As compared to other methods, the SVR provided better performance especially in multi-step prediction of CPU time and network bandwidth.

There are also many works on using SVR in the prediction of other kinds of time series. Financial time series is the most studied time series data. Many researchers have explored the usage of SVR in financial time series prediction [53, 54]. Because of the non-linear nature and inherent noise of financial time series, it is suitable to be used in evaluating the performance of non-traditional time series algorithms such as SVR and ANNs. Many different variation of SVR have been applied to it.

The comparison of SVR and ANNs with different structure was conducted by Trafalis and Ince [55]. They adopted ϵ -loss function and different training algorithm in the experiment and the result shows SVR is superior over other ANN based models. SVR was used in the prediction of S&P and other bond index by Tay and Cao [56]. They aimed to predict the data five steps ahead. The data set is divided into three sets: training data, validation data and test data. The data is preprocessed by exponential moving-average windows and outlier is replaced by the closest value. The SVR presents better performance compare to MLP and other models. They concluded this result is due to SVR's capability in extracting non-linear relationship and dealing with over-fitting issue. A "mixed expert system" was proposed in [57]. This system is formed by two phase. In the first phase, the data is processed by self-organizing feature map (SOM) to cluster data with similar statistical property. In the second phase, several trained SVR "expert" is included in the prediction. The experiment result shows that although the mixed expert system doesn't outperform single SVR, it presents much faster converging speed. C-ascending SVM which bases on SVR was developed by Tay and Cao[58]. It puts more weight of the ϵ -loss function on the more recent value and less on the distant value. It provides a better overall performance in prediction. LS-SVM combined with Bayesian evidence framework was developed by Van Gestel [59]. It presents better performance than traditional regression models when being applied on US short term T-bill and DAX30 market data.

Abraham compares the one-step-ahead prediction performance of SVM with traditional ANN, difference boosting neural network (DBNN) and Takagi-Sugeno neuro-fuzzy model [60]. The data source is Nasdaq-100 index and the NIFTY index. The experiment result of SVM is slightly better. He further developed a hybrid system which combines the four above algorithms with genetic algorithm (GA). It shows the hybrid system outperform individual algorithm. Ongsritrakul and Soonthronphisaj used gold price as the data source to evaluate the performance of MLP, decision tree and SVR [61]. SVR presents better performance than the others. Liang and Sun used Gaussian RBF as the kernel function of SVR. The model's prediction performance on Shanghai Stock Exchange (CISSE) data is compare with traditional SVR. A fuzzy support vector machines regression (FSVMR) method was proposed by Bao et al. [62]. They adopted RBF kernel function and find the optimal parameter of the model with cross-validation. The model is used to predict the

future value of Shanghai stock exchange and it provides encouraging performance.

Sansom et al. applied SVR in the prediction of electricity price prediction and compared its performance with MLP [63]. These algorithms are used to predict the price one week ahead. The experiment result shows SVR has faster converging speed than MLP. But it does not outperform MLP in prediction accuracy. Their experiment result is contradicted to other people's experiment result. They concluded the reason might be the way how they choose and use training data set. Huang et al. used SVR in the prediction of corporate credit rating [64]. They used the corporate credit rating data of Taiwan and US as the data source. They mainly used SVR and ANN to conduct the comparison experiment. For SVR they adopted 21 input variables and for ANN they used back propagation structure. And finally, they state that SVR present better result. Boase and Pal used SVM to predict the business failure and success of Dotcom Company [65]. They used selected financial feature as the input parameter. It is defined by expert of the investment area. The experiment result shows the accuracy of identifying a survived company is better than identifying a failed one. SVM was also used to predict the "turn ratio" of customer in the insurance industry by Hur and Lim [66]. The "turn ratio" here means customer change from one insurance company to another company. They used a SVM model with 15 input variable and the experiment result shows SVM outperform other algorithms. Using SVR to predict the product value of machinery industry was discussed by Pai and Lin [67]. They conducted the comparison experiment of SVR and seasonal auto-regressive integrated moving average (SARIMA) method and a general regression neural network (GRNN). Different evaluation index was used to compare their performance and the result shows SVR outperform others.

Chang et al. proposed a SVR based model to predict the daily maximum electrical load with the temperature data and previous electrical load data [68]. Their study shows there is clear seasonal component in the data set. The electrical load tends to be less than usual in holidays and this is referred as "holiday effect". The local weather is also related to the electrical load. Thus these data is used as input parameter to find the electrical load of a specific time. Their model is named as EUNITE network competition. The comparison of short term prediction on electrical load between SVR and traditional regression model was

conducted by Mohandes [69]. They eliminated the cycling and trending component of the data set in the preprocess phase. Their result shows that SVR is better than traditional AR model. Their experiment also shows the accuracy of SVR is increased as more data is added into the training set. Tian and Noore proposed SVM based model to conduct multi-step ahead prediction of electrical load data [70]. Their model uses weather factor such as the temperature, humidity as the input to do the prediction. The input data used in their experiment is normalized to a range of (0, 1). In their experiment, SVM outperforms ANN and radial basis function neural network. Bao et al. uses self-organizing map (SOM) along with SVM to conduct short term prediction of EUNITE competition data [71]. SOM is used to group data of similar weather condition into the same cluster. Their experiment result shows the combined model outperform individual SVM model. Another conclusion stated by them is that smoothing the data set will bring worse performance.

As there are many factors which correlate to the electrical load, Ji et al. adopted mutual information to select the proper input variable for SVM [72]. They chose the input variable which maximizes the computation of Shannon's entropy. They used LS-SVM to predict the data of six steps ahead. Poland electricity data set was used as the data source. And this method presents very encouraging performance. Li et al. proposed a method which firstly identifies a day with similar data to the next day [73]. Then the similar day's data is used to train the SVR model and trained model is applied to do one-day-ahead prediction. He et al. proposed a hybrid model of ARIMA and SVM. They assumed there are two components in the electrical load data. One is linear and the other one is non-linear. They used ARIMA to predict the linear component and use SVM to model the non-linear component. Using MAPE as the performance criterion, the result shows hybrid model is better than others [74].

2.2.4 Clustering Approaches of Time Series

Clustering is a technique which aims to classify unlabeled data with similar structure into the same group. Its purpose is to find proper partition of a data set and minimize the dissimilarity within the same group and maximize the dissimilarity between different

groups. A lot of clustering algorithms have been developed to process time series data [75]. They are mainly classified into five categories: model-based methods, grid-based methods, density-based method, hierarchical methods and partitioning methods. Han and Kamber have introduced them in [76] and here we will present a brief introduction.

A partition clustering method constructs k partitions of the unlabeled data. A crisp partition means each object just belongs to one partition while a fuzzy partition means an object may belong to more than one partition. K-means and K-centroid algorithm are two renowned heuristic methods. K-means represents certain cluster with the mean value of the objects in that cluster and K-centroid represents a cluster with the mostly centrally located object. There are also fuzzy version of these two algorithms which are fuzzy c-means algorithm and fuzzy c-centroids algorithm[77].

A hierarchical clustering algorithm clusters the data by grouping them into tree shape clusters. There are two kinds of hierarchical clustering methodologies which are agglomerative and divisive. Agglomerative cluster algorithm works by initially take each object as a single cluster and then merge the clusters with similar feature to generate a new cluster. The iterative process goes on until certain criterion is satisfied such as desired number of clusters has been generated. Divisive cluster algorithm works in the opposite way.

The general idea of density based clustering is to grow a cluster until its density exceeds the pre-defined threshold [78]. OPTICS was proposed and it is a method which computes an augmented clustering order for further analysis [79]. The order contains information could be used in determine the parameter setting which is a major challenge of using density based clustering method.

The general idea of grid-based clustering method is to transfer the object space into a feature grid and further group them into different clusters. STING is a well-known grid based clustering approach [80]. It uses different size of rectangle cell to represent different level of resolution. The statistic attribute for each cell is pre-defined. After the grid established, a query process to attach the object to proper grid is executed. In that process, the confidence interval which reflects its relevance to current query is computed.

Inappropriate cell is removed from the current query and the query is executed from top to bottom.

The model-based clustering method assumes there is certain model lying in each cluster and attempt to use a model to extract that underlying relationship. Major clustering algorithms of this kind contain AutoClass and self-organizing feature maps [81]. AutoClass attempts to use Bayesian statistical analysis to estimate the proper number of clusters. Self-Organizing feature map is an ANN based clustering algorithms.

As the data ranging from various areas like engineering, finance, business, energy application and climate could be regarded as time series, the research on time series is of interest. However, using clustering algorithm to process solar radiation time series has not been thoroughly explored by any researcher. Classifying the unlabeled time series data into different clusters according to their statistic properties is often useful for further analysis. The rest of this chapter will introduce works on time series clustering which have been done so far. The existed clustering algorithms on time series are grouped into three categories: clustering algorithm which works directly on raw data, clustering algorithm which works on extracted feature and clustering algorithm with model built on the raw data.

a) Raw Data Based Algorithms

This kind of clustering algorithm may work on either time domain or frequency domain. Kosmelj and Batagelj modified traditional relocation clustering procedure which is originally developed for static data and used it to cluster time series data [82]. They introduced a cross-sectional based general model to measure the dissimilarity between two different time series. To find the optimal number of clustering, they adopted an index which is called generalized ward criterion. Liao et al. also used cross-section approach combined with other clustering algorithms like K-means and fuzzy c-means [83]. They used this method to cluster time series which is not evenly sampled. Golay et al. used fuzzy c-means to cluster MRI data to assist further medical diagnose [84]. Different distance measurements like Euclidean distance and cross-correlation-based distance are used in their algorithm. The cross-correlation-based distance provides best performance.

They also investigated the effect of other factors like preprocessing approach and the number of clusters. Wijk and Selow clustered the daily power consumption data with agglomerative hierarchical clustering method [85]. They adopted root mean square distance to measure the dissimilarity. How the power consumption distributes over the year is studied to enhance the efficiency of power system. Kumar et al. proposed a hierarchical clustering method to group seasonality sequence into certain number of clusters [86]. They used independent Gaussian model to simulate the data error. The experiment result shows their method outperforms traditional cluster method like K-means. Moller-Levet et al. proposed a novel short time series distance (STS) to measure the similarity in their study of DNA microarray data [87]. The DNA data is sampled with the same interval. Adopting their novel distance measurement in fuzzy c-means clustering algorithm, the experiment result is better than before. Kakizawa et al. applied hierarchical clustering method combined with K-means cluster to group the earth quake data into different clusters [88]. Shumway used Kullback-Leibler discrimination information measure to cluster non-stationary time series based on the dissimilarity in frequency domain [89]. They applied agglomerative hierarchical cluster analysis to the data until a final set of two clusters are obtained. Policker and Geva used fuzzy clustering procedure to cluster non-stationary time series [90]. The optimal number of clusters should be determined by a temporal cluster validation criterion.

Liao et al. proposed a two-step algorithm which is capable to cluster multivariate time series with equal or unequal interval [91]. The first step is to convert the multivariate time series into univariate discrete-valued time series. The converted series is regarded as process of varied states. The second step is using traditional clustering algorithm like K-means to group the converted series into different clusters. The optimal number of clusters is determined experimentally. Euclidean distance and Kullback-Liebler distance are both used in different step of the experiment.

b) Feature Based Algorithms

The raw time series data usually appears with high dimension. This raises the computation difficulty for clustering task. Another difficult of processing the raw data directly is that

there is a lot of noise lying in it. Hence, feature extraction method is proposed to address these problems. It is hard to find a universal solution which is applicable for different types of time series. One method works well on one application may present bad performance on another.

Wilpon and Rabiner proposed an algorithm which is based on K-means for the recognition of isolated words [92]. First, it forms template with the user's knowledge about the series to be processed. Each template represents a specific pattern of the series. Then the series is clustered based on these templates. This modified K-means algorithm also addresses the problems of how to find cluster center and how to split cluster to obtain proper number of clusters. Itakura distance is used to measure the dissimilarity of two different word patterns. The modified K-means algorithm outperforms other clustering algorithms in the experiment. Shaw and King used hierarchical algorithm to cluster the time series indirectly [93]. The original time series is preprocessed and normalized by the amplitude of the largest peak to generate so called spectra. Principal component analysis (PCA) is also applied to reduce the dimension. Goutte et al. used K-means and hierarchical clustering algorithms to group the fMRI series into different clusters [94]. They used cross-correlation function as feature space. They further discuss the potential capability of their method in [95]. They also used well-known information criterion like AIC, BIC to determine the optimal number of clusters. It shows that the proposed feature based clustering algorithm provides encouraging result. Fu et al. proposed a method which uses SOM combined with sliding window to group the time series segment with similar characteristic into clusters [96]. For dimension reduction consideration, they introduced a new concept: perceptually important point (PIP). Thus, the original time series is converted to series contain PIP only. Mean squared distance is adopted as the dissimilarity measurement. Owsley et al. proposed an algorithm called sequence cluster refinement algorithm (SCRA) [97]. This algorithm contains discrete hidden Markov models (HMM) and SOM. It uses SOM to cluster the time series through template matching approach. The different cluster is represented by a quantized vector in the transient matrix. During their experiment, they modified the structure of SOM to improve its generalization ability. Vlachos et al. proposed a clustering algorithm based on Haar wavelet transform [98]. The Haar wavelet transform is firstly used to transform the time series and then K-means is

applied to cluster the converted time series in various resolution levels. The clustering error for all levels is computed to measure the accuracy of clustering.

c) **Model Based Algorithms**

This class of approaches assumes that the time series is generated by certain model or by certain distributions. Piccolo introduced an algorithm which clusters time series by matching them to a set of ARIMA models with different order [99]. A distance matrix is computed to record the distance between these models. A dendrogram is constructed by linkage clustering method based on the distance matrix. Baragona proposed an approach to cluster time series based on the cross-correlation absolute value (CCAV) [100]. They clustered with aim that maxim CCAV within cluster and minimize the CCVA between clusters. Fitting the time series with existed model, the residual series is computed. The residual series is used to compute the cross-correlation absolute value. They evaluate several algorithms in their experiment. The experiment result shows Tabu search provides the best performance. Maharaj proposed a method using hierarchical clustering approach based on the hypothesis test result [101]. They firstly used AR (k) model with different input parameter to fit the input time series. Chi-square test is conduct on two input time series with the hypothesis that they are generated by the same AR process. If the hypothesis is accepted, the two time series will be grouped into the same cluster. Ramoni et al. proposed Bayesian algorithm dynamic clustering approach (BCD) [102]. BCD converts the time series set into a set of Markov chain (MC) first. Then they mined the most probable generating process from the MC set. The algorithm is an example of Bayesian models selection problem. Kullback-Liebler distance is adopted to construct the MC transition probability tables. The performance is assessed by the loss of information during the clustering process. Xiong and Yeung proposed an ARIMA model based clustering algorithm [103]. They assumed that the input time series set is generated by several different ARIMA model and each model represents a specific cluster. They used an improved expectation-maximization algorithm to estimate the parameters of the algorithm. The optimal number of cluster is determined automatically during this process. Tran and Wagner proposed a fuzzy c-means clustering and Gaussian mixture model based algorithm to process the speech signal [104]. Biernacki et al. proposed a method use integrated

completed likelihood (ICL) to select proper Gaussian mixture model and appropriate cluster number [105]. ICL is developed based on BIC. Experiment result shows ICL solve disadvantage of overestimating number of cluster lying in BIC. Oates et al. propose a method based on hidden Markov model (HMM) to cluster time series data [106]. They adopted dynamic time warping (DTW) as the distance measurement. Then they applied hierarchical agglomerative clustering algorithm to obtain the initial parameter of HMM model. The HMM model for each cluster go through an iterative training process to obtain proper parameter finally. Li and Biswas proposed a HMM based approach for the clustering of temporal data [107]. They assumed the temporal data set has Markov property and can be expressed as probabilistic walk with fixed set of states. Their approach works well on model generated artificial data set. Li et al. further improved the approach by using Bayesian HMM instead of normal HMM and using BIC as the criterion to generate proper states for this approach [108]. They generated an initial model and increase its model size until optimal BIC value is obtained. Their experiment uses both artificial data and actual ecology data and the result is very encouraging.

2.3 Summary

In this section we have reviewed different methodologies which could be used to predict solar radiation. There are basically two categories of them. One is empirical solar radiation model. It tries to model the solar radiation with only meteorological parameters. A lot of work is done on this topic. But empirical solar radiation is incompetent in detecting abnormal change of solar radiation. Besides, most of the empirical models aim to conduct long term prediction while short term solar radiation prediction is of great importance for many renewable energy projects.

As solar radiation can be regarded as time series generated by stochastic process, a lot of time series analysis approaches can also be applied on it and conduct the prediction. We reviewed the development of the some popular time series prediction algorithms in this section. The usage of these algorithms in the prediction of solar radiation and other time

series is also discussed.

We presented a review of the development of ARIMA model in this section. The study of ARIMA has been started since 1940s. Box and Jenkins' solid methodology of building and using ARIMA model makes it the most popular model for time series prediction. We also introduced a lot of application and variation of ARIMA model in this section. It has achieved success in many different kinds of time series applications. However, ARIMA model requires the input time series to be stationary. Time series in real world is usually non-stationary and contains a lot of noise. Thus, some preprocess technologies is needed when using ARIMA model. Determining the optimal order is another challenge for the usage of ARIMA model. Although many researchers have addressed these problems, there is still no perfect solution.

The development of ANN and its application in time series prediction is also reviewed in this section. Various structure of ANN besides MLP is introduced. Many experiments have justified ANN's capability in predicting time series.

However, there are also some challenges in using ANN. Avoiding over fitting during the training process is one of them. Because the prediction capability of ANN is "learned" from the training data set, it is likely the trained model is affected too much by the training data and lost the generalization ability. The relationship of generalization capability and the size of training data were discussed. It is found that different problems require different size of ANN structure. Thus finding ANN with proper size for a specific time series becomes a challenge for all researchers in this area. Information theory is applied to handle this problem and several stepwise methods are developed.

Combination approach of ANNs with different structure was also addressed. Experiment result shows that the combined model has better prediction capability than individual ANN. Its superior performance may due to that the hybrid model uses ANN with different structure to capture different components of time series. Hybrid models which combine ANN with other algorithm are discussed in this section as well. Comparison experiment of sunspot data shows the hybrid model could improve the accuracy considerably. The usage of various kinds of ANN in solar radiation prediction is reviewed in this section too.

We also reviewed a lot of research work on SVM/SVR in this section. All these work proved that SVM/SVR is powerful prediction algorithm. Especially for those non-linear time series which cannot be processed by traditional statistic time series model, SVM presents its capability in extracting and modeling the non-linear relationship. However, there are challenges lying in designing the SVM model. First of all, the selection of kernel function is of great importance for the final performance. By now Gaussian kernel is the most popular kernel but this is not the universal solution for other time series. The estimation of parameter is also a topic needs further studying. Many traditional computational intelligence training algorithms are adopted in the training process of SVM. But some of them may cause problems such as slow convergence rate. Choosing appropriate input dimension is another challenge of SVM model. Some researchers' work shows improper input dimension may cause the predicted value seriously deviate from actual value.

Time series clustering is a comparatively new but important topic for time series prediction. Proper clustering of the time series would be helpful for understanding its nature and for further improving the prediction accuracy. Lots of research work has been proposed on this topic. These studies are grouped into three different categories based on how they work on data. They may work directly on the raw data set and this kind of algorithm has an advantage of preserving the integrity of data set and can be implement easily. However, huge data set will bring about slow converging rate. Noise lying in the data source may cause clustering error. Thus another kind of clustering algorithm which works on extracted feature of the raw data set is developed. This largely improves the efficiency of the clustering and reduces the impact of noise. Finding proper feature set to represent the raw data is of great importance. Some feature extraction based clustering algorithm doesn't work well because the loss of information during the feature extraction process. Another kind of clustering algorithm which assuming the data of each cluster satisfy certain distribution or model is proposed recently. Gaussian distribution is the mostly widely used process to model the data from different cluster. Experiment result shows they outperform many traditional algorithms in addressing many clustering tasks.

Chapter 3. Prediction of Solar Radiation Time Series with TDNN and Statistical Approaches

3.1 Introduction

As mentioned previously, the traditional statistical time series analysis has been proven to be competent in the linear time series prediction. The most well-known statistical time series analysis methodology is auto regression integrate moving average (ARIMA) which is developed by Box and Jenkins [109]. The methodology contains time series model of auto regression (AR), moving average (MA) and autoregressive moving average (ARMA) process. It provides a method to model a stationary time series with AR and MA procedure. Another advantage of it is that it requires only a few parameters. However, only very few actual time series process is stationary. Most of the real world time series is non-stationary. Before ARMA could be used to model them, they must be preprocessed. Differencing is a widely accepted preprocesses strategy to generate stationary series. Combined both ARMA and differencing, the whole process is called ARIMA. A challenge when applying ARIMA model to conduct time series analysis is to determine its order. One strategy is to use a trial and error method. That means using ARIMA of different order to fit the training data iteratively until an optimal solution is found. Another way is using auto-correlation and partial auto-correlation function. Box has proved there are certain relation between the auto-correlation function, partial auto-correlation function and the order of ARIMA model. Recently developed information theory provides another way to find out the optimal order. The most well-known examples of them are Akaike's Information Criterion (AIC) and Bayesian information criterion (BIC).

Other than the traditional statistic method, some recently developed computational intelligence algorithms were also proven to be competent in time series prediction. One widely used algorithm is time-delay neural networks (TDNN). TDNN is developed based on artificial neural networks (ANNs) and is used for time series prediction. The study of using TDNN and other form of ANNs to conduct time series prediction is aggressively ongoing. TDNN is a data driven model. This means it requires no pre-knowledge during

the model building process. It is capable to learn from training data to capture the underlying relation without explicitly knowing it. It could even be used as a simplification of the underlying pattern of time series. This characteristic of TDNN made it extremely useful to solve many practical industrial problems. Very often, it is much easier to get actual input/output of a system than finding out the law behind it. However, one disadvantage of TDNN is it is quite sensitive to the error data of training data set. If the training data is not reliable or corrupted by noise, the output might seriously deviate from actual value. Another disadvantage of this kind of data driven model is that its performance is not as stable as traditional statistical methodology. Selecting proper structure is also important to the performance of TDNN and sometimes it depends on the user's experience. Choosing appropriate training data set, using proper transfer function and building optimal hidden layer structural are all the challenges in the research of TDNN.

In this chapter, both ARIMA and TDNN are applied to conduct the prediction of monthly average solar radiation series. And their performance is compared.

The rest of the chapter is organized as follows: Section 3.2 introduces the ARIMA model. Section 3.3 introduces the detail of TDNN model. We have used both models to conduct prediction of solar radiation series. The experiment result is presented in section 3.4. Section 3.5 is the summary.

3.2 ARIMA Model

One popular classical statistical algorithm used for time series prediction is the autoregressive moving integrate average models which is also known as ARIMA[110]. The popularity of the ARIMA model is its ability to extract useful statistical properties and the adoption of the well-known Box-Jenkins methodology [111].

However, before applying ARIMA model, it must be ensured the time series is stationary. A time series is stationary if its statistical feature does not change with time. A time series

$\{y_t\}$ is stationary if it meets following requirements:

- a) Its expectation $u(y_t) = E(y_t)$ is constant.
- b) Its variance $\sigma_y^2 = E((y_{t+m} - u_y)^2)$ is constant, so $E((y_t - u_y)^2) = E((y_{t+m} - u_y)^2)$
- c) Its auto covariance $\gamma_y = cov(y_t, y) = E((y_t - u_y)(y_{t+m} - u_y))$ is constant, so for any k, $cov(y_t, y_{t+k}) = cov(y_{t+m}, y_{t+m+k})$

3.2.1 Autoregressive

If the value of a stationary time series y_t can be represented by the linear combination of p past value: $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ plus the residual value at time t , and the residual series is white noise. It can be expressed as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad 3.1$$

This is the autoregressive model (AR). AR with order p can be expressed as AR (p).

3.2.2 Moving Average

If the value of a stationary time series y_t can also be represented by the weighted average of q past residual series $\{\varepsilon_t\}$ plus the residual value at time t , and the residual series $\{\varepsilon_t\}$ is white noise, it can be defined as:

$$y_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad 3.2$$

This is the moving average model (MA), with order q it can be recorded as MA (q).

3.2.3 Autoregressive and Moving Average

Autoregressive and Moving Average Model (ARMA) is the combination of Autoregressive model and Moving average model. It can be expressed as:

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad 3.3$$

This can also be recorded as ARMA (p, q) .

3.2.4 Autoregressive Integrated Moving Average

If a time series is non-stationary, it must firstly be differenced to get a stationary series before applying model to extract its statistic characteristics. Then ARMA (p, q) could be used on it. If the stationary series is obtain after d times of differencing. The model applied on the originally series can be recorded as ARIMA (p, d, q) . And it is a autoregressive integrated moving average model.

3.2.5 Building Model for Time Series.

Since the models introduced before are based on stationary time series, the stationarity of time series must first be checked.

a) Test of Stationarity

The Augmented Dickey-Fuller (ADF) test [112, 113] is a popular method to test the stationarity of time series. It tests whether there exists unit root in a time series. If there is a unit root in time series, the time series is not stationary; otherwise, it should be stationary.

The model of ADF test is:

$$\partial Y_t = \mu + \beta t + \rho Y_{t-1} + \partial_1 Y_{t-1} + \cdots + \partial_p Y_{t-p} + e_t \quad 3.4$$

The μ in this equation is a constant, β represent the trend. And p is the order of autoregressive process. e_t is a sequence of independent normal random variables with mean zero and variance $\sigma = 1$.

ADF test applies equation 3.4 to construct a statistics to check whether the series has unit root (or $\rho = 1$).

b) Model Identification

The orders p and q for the autoregressive and moving average operators can be determined by studying the theoretical autocorrelation and partial autocorrelation function.

Briefly, if the autocorrelation of the time series tails off and the partial autocorrelation cutoff, it can be modeled by AR. On the contrary, if its partial autocorrelation trials off and autocorrelation cutoff, it can be modeled by MV. If both the autocorrelations and partial autocorrelations tail off, a mixed process is suggested. In general, autoregressive (moving average) behavior, as measure by the auto-correlation function, tends to mimic moving average (autoregressive) behavior as measured by the partial autocorrelation function.

c) Determining the Order of ARIMA

As discussed above, once the lag we use in the mixed process is appropriate, the estimation of the prediction model should be very close to the actual data. We use the following sum of square error (SSE) measurement:

$$SSE = \sum_{t=1}^T (Y_t - \hat{Y}_t)^2 \quad 3.5$$

As the number of parameters increase, SSE will decrease accordingly. However, too many parameters may cause the problem of over fitting. To take both the goodness of fitting and the number of parameter into consideration, we adopt AIC (Akaike's Information

Criterion) [112] to find the optimal model. AIC can be obtained from the following equation:

$$AIC = \exp \frac{2k \sum_1^T e_t^2}{T} \quad 3.6$$

T is the sample size, $k=p+q$ is the number of parameters in the model. As the fitting accuracy of prediction model increase, the value of AIC will decrease.

3.3 TDNN Model

Time delay neural network (TDNN) is developed based on traditional neural network (NN). NN has been widely applied in many areas such as: process control, prediction, pattern recognition. It shows great competency in solving nonlinear problem. And one advantage of NN is that it can be trained to solve problems when it's difficult to establish an analytic model. In a traditional NN, the neurons in the input layer take input data and multiply it with a connection weight. The result is summed together and then passed through a transfer function. The output of the transfer function will be compared with a threshold value and then how to update the connection weight of NN will be decided. The training and updating are an iterative process and many epochs of updating are needed to obtain the optimal structure.

Time Delay Neural Network (TDNN) is developed from general feed forward neural network. It is used to obtain the nonlinear relationship between the input and output position in time series [114]. The conventional neurons of a neural network provide their response to the weighted sum of the inputs. However, for TDNN, it uses the sum of a finite number of past inputs. In this way, the output provided by a given layer depends on the output of the previous layers computed based on the temporal domain of input values. Since TDNN and the traditional NN have similar structure, back-propagation and some other algorithms can therefore be applied to train the TDNN. As in the traditional NN, the

TDNN also has a training phase. The updating of weight can be expressed with the following equation:

$$W_{ij,n} = U_n + \alpha W_{ij,n-1} \quad 3.7$$

$W_{ij,n}$ is the current weight connection, α is the speed of updating. U_n is the updating function. There are several methods to update the weights. For example, using the conjugate gradient-based minimization algorithm, update is computed along the conjugate direction so that the error function decreases fastest. Other algorithms, which can produce faster convergence, are the Newton's method and the Levenberg - Marquard algorithm, which uses the second derivative of the error function. There are many methods that can be used to train NN , we introduced four most popular algorithms: (1) Basic back Propagation algorithm (BP), (2) Quasi-Newton algorithm (QN), (3) Levenberg - Marquard algorithm (LM), (4) Conjugate Gradient algorithm(CG). Here is some brief introduction of these algorithms [114].

3.3.1 Basic back propagation algorithm

This algorithm updates the value of weight with a fixed length in the direction of the negative gradient direction to minimize the error function. The error function is as following:

$$E = \frac{1}{2} \sum_{t=1}^n (t_i - o_i)^2 \quad 3.8$$

t_i is the target data set and o_i is the output of NN. Then we can get:

$$g_n = \rho \frac{\partial E}{\partial W_{ij,n}} \quad 3.9$$

$W_{ij,n}$ is the weight between neuron i and j in the n th iteration.

And then we could define the U_n by:

$$U_n = -g_n \quad 3.10$$

The main disadvantage of BP is its slow convergence rate and false minimal.

3.3.2 Quasi-Newton algorithm

Newton method can converge very fast to the optimal value. The update equation is :

$$W_{i,j,n} = W_{i,j,n-1} - A_n^{-1} g_n \quad 3.11$$

Figure 3.1 The structure of TDNN

A_n^{-1} is Hessian matrix which is formed by second derivatives. And that's why Newton method converges faster than other algorithms. But constructing the Hessian matrix incurs more computation burden.

Quasi-Newton method uses an iterative method to obtain the approximation of Hessian matrix instead of computing it. Hence Quasi-Newton method is more computation efficient than Newton method.

3.3.3 Levenberg-Marquardt algorithm

Levenberg-Marquardt algorithm (LM) is the most popular training algorithm. It also uses the second derivative of error function to find the optimal value for the weight. The updating function is as following equation:

$$U_n = -(J^T \times J + \mu I)^{-1} \times J^T \times e \quad 3.12$$

J is Jacobian matrix which is formed by the first derivatives of the error function of the NN. e is the error term, I is the identical matrix and μ is the scalar parameter used to update the NN. The μ is decided by the error function. When the error is small, the μ would be decreased and when error increase the μ would also be increased accordingly.

3.3.4 Conjugate Gradient algorithm

Conjugate Gradient algorithm is a well-known optimization algorithm. It converges to the optimal along the gradient direction of the error function. It is widely used as training algorithm for NN due to its simplicity and fast convergence. The basic form of its definition equation could be:

$$U_0 = -g_0$$

$$U_n = -g_n + \beta_n \times U_{n-1} \quad 3.13$$

g_n has been discussed above. β_n is a constant determined by the algorithm. For example, Fletcher Reeves update's β_n is defined by:

$$\beta_n = \frac{g_n^T \times g_n}{g_{n-1}^T \times g_{n-1}} \quad 3.14$$

During the simulation, LM shows great converging speed and good prediction accuracy compare to other training algorithms. Thus it is chosen as the training algorithm in our simulation.

3.4 Simulation Result

3.4.1 Data Source

The solar radiation data obtained in year 2009 and 2010 in Singapore is used in the simulation. The data is captured and recorded by the observation station located in Nanyang Technological University (NTU) and can be easily accessed online. For the reason that daily solar radiation data is highly affected by noise caused by weather, monthly average of solar radiation is used in the simulation. The data of 2009 is used as training data and the data of 2010 is used as testing data. The monthly average of daily solar radiation of January 2009 is shown in Figure 3.2.

3.4.2 Stationarity Test

Figure 3.2 shows that the mean of the solar radiation time series is not constant. Thus it may not be stationary. To confirm our assumption, we conduct the unit root test.

As shown in Table 3.1, the statistical test value is -0.8118, which is above the 10% critical value. That implies that we cannot reject the null hypothesis. There is unit root lying in the series. It is therefore non-stationary.

To achieve stationary series, first order difference is performed on it.

As shown in Table 3.2, the statistical test value of first order difference is -10.2992. It is below the 1% level critical value, which implies that we could reject the null hypothesis with 99% confidence level. Thus there is no unit root lying in the series and it is therefore stationary.

Figure 3.2 monthly mean of solar radiation of Jan 2009

Figure 3.3 first order difference of monthly mean of solar radiation

Table 3.1 The ADF test of monthly mean of daily solar radiation

ADF test	Test result	
	<i>Significant level</i>	
Test statistic		-0.8118
Critical value	1%	-3.5190
	5%	-2.8994
	10%	-2.5870

Table 3.2 The ADF test of first order difference of monthly mean of daily solar radiation

ADF test	Test result	
	<i>Significant level</i>	
Test statistic		-9.9740
Critical value	1%	-3.5190
	5%	-2.8994
	10%	-2.5870

3.4.3 Model Identification and Order Determination

Figure 3.4 the AIC of ARIMA with different order

The result shows that ARIMA (9, 1, 9) is the optimal model for the data. Figure 3.5 is the comparison of prediction and actual data.

Figure 3.5 the comparison of actual data and prediction of ARIMA

3.4.4 Training of TDNN

During the simulation of TDNN, LM shows great converging speed and good prediction accuracy. Thus it is chosen as the training algorithm in our simulation.

The number of hidden neuron affects the prediction accuracy significantly. To find the optimal number of hidden neuron, a validation experiment is performed on various TDNN structure. Symmetric Mean Absolute Percentage Error (SMAPE) is used to evaluate the performance of TDNN with different number of neurons in the hidden layer.

As shown in table 3.3, TDNN with five hidden neurons present the best prediction performance. Hence, it is used in the comparison experiment.

After using the data from January 2009 to train the TDNN, the trained model is applied to predict the monthly average series of January 2010. The comparison of prediction and actual data is shown in Figure 3.6.

Table 3.3 SMAPE of TDNN with different number of hidden layer neurons

Neurons Month	1	2	3	4	5	6	7	8	9	10
Jan	3.24	5.81	1.12	4.25	3.89	5.36	3.42	8.10	6.21	8.10
Feb	8.71	49.6	3.17	4.88	2.87	6.61	2.89	7.25	4.62	5.94
Mar	3.57	3.44	3.60	6.52	6.24	7.52	4.28	5.76	20.0	5.89
Apr	4.39	4.57	1.93	12.2	2.32	5.21	6.60	7.08	10.4	9.76
May	5.53	59.1	11.0	6.19	4.13	5.92	6.37	8.11	5.24	10.7
June	11.2	5.69	4.36	7.17	10.0	5.61	10.7	6.74	10.2	8.65
July	3.65	4.16	43.8	5.30	3.53	4.60	4.88	6.31	3.92	6.49
Aug	63.4	2.64	4.05	3.65	8.11	4.88	3.23	4.81	6.03	9.40
Sep	4.12	5.64	5.34	6.91	3.07	5.19	14.8	7.95	3.95	4.87
Oct	4.74	2.60	14.2	5.72	4.32	15.2	4.60	6.70	3.07	7.76
Nov	2.94	57.0	3.60	6.06	4.27	6.13	6.89	7.26	6.79	7.03
Dec	1.42	1.77	2.41	4.37	13.0	3.83	2.75	4.66	3.19	3.81
Mean	9.75	16.84	8.23	6.11	5.54	6.34	5.95	6.73	6.98	7.37

3.4.5 The assessment of prediction performance and discussion

To evaluate the prediction performance, the well-known measure symmetric mean absolute percentage error (SMAPE) is used in the simulation; it is given by equation 3.15.

This measurement is proposed by Makridakish and Hibon. It accounts for a different number of observations in the individual data sub-samples of training and test set, and the different scale between individual series. It is also established best-practice in industry and in previous time series prediction competitions such as NN3 time series forecasting competition [115].

$$SMAPE = \frac{1}{n} \sum_{t=1}^n \frac{|y_t - \hat{y}_t|}{(y_t + \hat{y}_t)^2} \times 100 \quad 3.15$$

y_t is the actual value of time t and $\hat{y}_t$ is the estimated value, n is the sample size.

The widely accepted measurement root mean square error (RMSE) is also use to evaluate the prediction performance, it can be given as:

$$RMSE = \left(\frac{1}{N} \sum_{i=1}^N (e_i - m_i)^2 \right)^{\frac{1}{2}} \quad 3.16$$

Table 3.3 is the SMAPE of the prediction of different monthly mean solar radiation.

Figure 3.6 The comparison of actual data and prediction of TDNN

The SMAPE of ARIMA and TDNN is presented in Table 3.3.

Table 3.4 SMAPE of ARIMA and TDNN

	SMAPE of prediction of ARIMA	SMAPE of prediction of TDNN
Jan	10.98	3.89
Feb	23.93	2.87
Mar	19.25	6.24
Apr	23.99	2.32
May	27.04	4.13
June	45.74	10.00
July	42.32	3.53
Aug	35.60	8.11
Sep	14.75	3.07
Oct	12.82	4.32
Nov	28.64	4.27
Dec	23.05	13.09

Table 3.5 RMSE of ARIMA and TDNN

	RMSE of prediction of ARIMA	RMSE of prediction of TDNN
Jan	50.20	15.58
Feb	85.27	13.96
Mar	94.59	34.06
Apr	62.71	11.50
May	82.32	4.65
June	98.92	25.39
July	53.04	8.93
Aug	94.87	31.77
Sep	89.77	11.28
Oct	85.68	22.30
Nov	31.85	10.64
Dec	43.89	12.28

Table 3.4 shows that the accuracy of prediction of monthly average solar radiation of TDNN is much better than ARIMA. The prediction accuracy has been improved a lot in every month. The biggest improvement is in July. The SMAPE of TDNN is 91.64% lower than ARIMA. In December, the SMAPE of TDNN is 43.18% lower than ARIMA, which is the least improvement. The result shows that TDNN is competent in predicting the monthly average of solar radiation.

To further study the performance of ARIMA and TDNN in the prediction of monthly average solar radiation, we also use another index: root mean square error (RMSE) to assess the accuracy. The result is presented in Table 3.5. We can see the result of RMSE is similar to SMAPE. TDNN outperforms ARIMA in every month.

3.5 Summary

In this chapter, two well-known algorithms have been used to predict the short term future trend of solar radiation. In the experiment, the monthly average solar radiation series is found to be non-stationary. First order difference is performed on it to obtain stationary

series. After model identification phase, a proper ARIMA model is used to conduct the prediction. TDNN is also used to predict the short term future trend of solar radiation. The monthly average solar radiation of 2009 and 2010 of Singapore is used to assess the performance of different model. Finally, TDNN presents better prediction performance than ARIMA in every month. However, in the simulation it is also noted that TDNN is very sensitive to the training data. Once it is used to predict fast fluctuate data like the daily solar radiation, TDNN will present huge error. On the contrast, ARIMA tend to present a stable prediction performance even when the data source changing ups and downs frequently.

In the experiment, both ARIMA and TDNN have shown their advantages. The challenge now is whether we could find a way to combine their advantages together. This problem will be addressed in the following chapters.

Chapter 4. Hybrid Model of ARMA and TDNN for Solar Radiation Time Series Prediction

4.1 Introduction

In the previous discussion, we found that both statistic model and computational intelligence model have their own advantages in prediction. Traditional statistic model such as ARIMA is able to present consistent prediction performance while the computational intelligence model is more capable in capturing the non-linear relationship intrinsic of the time series. In this chapter, we aim to combine them to take both advantages.

The empirical finding shows that hybrid model outperforms individual model. It is found that the performance of simple hybrid schema is as good as elaborate ones. There are several reasons for the superior performance of hybrid model. First, any prediction model might bring systemic bias because of the imperfection of its theory. The systemic bias could be well controlled by combine appropriate prediction model. Second, one important assumption of traditional statistical approach is the time series to be processed is stationary. That means the series has consistent statistic property. The relationship between the past and the future data of that time series remain unchanged. However, this assumption is rarely the case in reality, the majority of time series is non-stationary. Although there are lots of methods to transform it to obtain stationary series, their result is just an approximation and could still contain non-stationary component. Artificial intelligent method apparently makes more sense in modeling that. The parameters of a data driven prediction model are decided by the training data. The model is trying to capture certain relationship lying in the data set through the training process. Since the relationship is complex and is quite hard to be represented by only one model. Combine different models together might present a better result. And this covers a wide range of possibility with different outcome.

In this chapter, a new approach that contains two phases is used to predict the hourly solar radiation series. In the detrending phase, several empirical models are applied to remove

the non-stationary trend lying in the solar radiation series. To evaluate the goodness of different detrending models, the Augmented Dickey – Fuller method is applied to test the stationarity of the residual. The optimal model is used to detrend the solar radiation series. In the prediction phase, the Autoregressive and Moving Average (ARMA) model is used to predict the stationary residual series. Furthermore, the Time Delay Neural Network (TDNN) is applied to do the prediction. Because ARMA and TDNN have their own strength respectively, a novel hybrid model that combines both the ARMA and TDNN, is used to produce better prediction.

The rest of the chapter is organized as follows: Section 4.2 introduces some widely used detrending model for solar radiation. Section 4.3 is the simulation result of detrending. Section 4.4 introduces the prediction algorithms and the experimental result is presented in section 4.5. Section 4.6 is the comparative analysis between the proposed hybrid model with other algorithms. Section 4.7 is the summary.

4.2 Detrending Models

As mentioned earlier, the solar radiation is non-stationary and we need to detrend it. Because of the unpredictable noise, it is not easy to find the trend in a specific day's series. Instead, we explored the trend in a much more stable series namely the monthly average series. Below are some models that model the general distribution of hourly solar series:

4.2.1 Jain's model

Jain's model [9] tries to fit the solar radiation series with a Gaussian function. That is:

$$r_h = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(h-m)^2}{2\sigma^2}\right) \quad 4.1$$

r_h is the solar radiation of different time. h is time, σ is the standard deviation of the Gaussian curve. It is a parameter that should be decided by the actual data. The parameter m represents the peak hour of a day.

4.2.2 Baig's model

Baig's model [116] is developed based on the Jain's model. Baig modified Jain's model to better fit the record data at the start and end of series. The model is:

$$r_h = \frac{1}{\sigma\sqrt{2\pi}} \left\{ \exp\left(-\frac{(h-m)^2}{2\sigma^2}\right) + \cos\left[180 \frac{(h-m)^2}{S_0-1}\right] \right\} \quad 4.2$$

S_0 is the sunshine hour of the day at a site with latitude φ and sun's declination could be calculated by:

$$S_0 = \frac{2}{15} \cos^{-1}(-\tan\varphi \tan\delta) \quad 4.3$$

δ is the angle between the rays of the Sun and the plane of the Earth's equator.

4.2.3 S. Kaplanis' model

Another interesting model is proposed by S.Kaplanis [10]. That is:

$$r_h = a + b \cos\left(\frac{2\pi(h-m)}{24}\right) \quad 4.4$$

The a and b in the equation should be decided according to the actual situation of different area. m is the peak hour of solar radiation in this area.

4.2.4 Al-Sadah's model

Al-Sadah found that high order polynomial model is quite good in fitting hourly global radiation on a horizontal surface [117]. This model is:

$$r_h = a_1 + a_2 h + a_3 h^2 \quad 4.5$$

4.3 Simulation of Detrending Models

In this section, these detrending models are simulated with actual data. The data collected is from the observation station set up at Nanyang Technological University, Singapore. The sampling interval is five minutes. We ignored the data between 10:00 pm to 7:00 am as there is consistently no solar energy received during this period. The unit of solar radiation energy is watt-hour per square meter (WH/m²). The monthly mean series of February 2009 is shown in Figure 4.1. Different models are applied successively to fit the monthly mean series.

Figure 4.1 the monthly mean series of February 2009

Using Jain's model to fit the monthly mean series, to estimate the parameter σ and m , the method of least square method was applied to train the model to find those parameters. The comparison of Jain's model and actual monthly mean series is presented in Figure 4.2.

Figure 4.2 the comparison of Jain's model and actual series of February 2009

The solid line is the actual series and dot line is the estimated series of Jain's model.

We can see that Jain's model generally fits the mid-range of the monthly mean series. But it fails badly at the beginning and ending of the model.

Baig's model aims to improve the accuracy at the beginning and ending of the actual series, as shown in Figure 4.3.

Figure 4.3 the comparison of Baig's model and actual series of February 2009

The solid line is the actual series and the dot line is Baig's model. Since the Baig's model is based on Jain's model, the general profile is the same. However, the model provides a much better fit at the beginning and ending stage of the actual series.

S. Kaplanis' model is different from the previous two models. This model is based on cosine function. The experimental data provides the estimate for the parameter a and b in the function,

Figure 4.4 the comparison of S. Kaplanis' model and actual series of February 2009

The solid line the actual series and dot line is the S. Kaplanis' model.

Using Al-Sadah's model also requires us using actual data to train the model to get the value of these unknown coefficients. We used the least square method and the actual data as input. Figure 4.5 is the result achieved after training.

Figure 4.5 the comparison between Al-Sadah's model and the actual series of February 2009

To evaluate the stationarity of the solar radiation after detrending by these models, we again use the Augmented Dickey-Fuller (ADF) test mentioned in section 3.4.2.

Figure 4.6 shows the residual of monthly mean solar radiation series after detrending by the models introduced above. The test is carried out on the residual series under the null hypothesis that the true underlying process contains a unit root. If the test result is above the critical value, that means we should accept the null hypothesis and the time series as not stationary; otherwise we should reject the null hypothesis and the time series as stationary. After this testing procedure, we can attain the statistical power, which is the probability that the test rejects a false null hypothesis.

Figure 4.6 the residual series after detrending by the 4 different models

In the test procedure, we use the same significance level 0.05 for all the 4 models.

Table 4.1 The ADF test for the detrending models

Detrending model	Statistical power	Significant level	Test result	Critical value
Jain's model	0.9961	0.05	-3.9002	-2.8994
Baig's model	0.9941	0.05	-3.7176	-2.8994
S. Kaplanis' model	0.9961	0.05	-3.8936	-2.8994
Al-Sadah's model	0.9990	0.05	-6.3815	-2.8994

From the table we can see that the test results for all these models are below the critical value. Hence, we can surmise that the residual series of all these models can be regarded as stationary series. The statistical power of Al-Sadah's model is the highest one, which means the residual series detrended by the Al-Sadah's model has the lowest probability to contain a unit root.

Another key factor that should be taken into consideration is the accuracy of fitting the actual series. To evaluate this accuracy, we use the indicator: root mean square error (RMSE) and the normalized root mean square error (NRMSE):

$$RMSE = \left(\frac{1}{N} \sum_{i=1}^N (e_i - m_i)^2 \right)^{\frac{1}{2}} \quad 4.6$$

$$NRMSE = \frac{\left(\frac{1}{N} \sum_{i=1}^N (e_i - m_i)^2 \right)^{\frac{1}{2}}}{\frac{1}{N} \sum_{i=1}^N m_i} \quad 4.7$$

In the above two equations, $\{e_i\}$ is the model fitted series and $\{m_i\}$ is the actual series.

The RMSE and NRMSE of different models are shown in the following table:

Table 4.2 RMSE and NRMSE of different models

Model	Error (RMSE)	Error (NRMSE)
Jain's model	0.0459	0.1407
Baig's model	0.0434	0.1330
S. Kaplanis' model	0.0444	0.1360
Al-Sadah's model	0.0231	0.0706

From Table 4.2 we see that Al-Sadah's model yields the best result in fitting accuracy.

Since Al-Sadah's model renders the best performance in both detrending and fitting, we chose it to detrend the solar radiation series for further prediction.

4.4 Prediction Models

4.4.1 ARMA Model

The concept of ARMA is introduced in section 3.2. Here we present a brief introduction again. The Autoregressive Moving Average (ARMA) model is usually applied to auto correlated time series data [110]. This model is a great tool for understanding and predicting the future value of a specified time series. ARMA is based on two parts: autoregressive (AR) part and moving average (MA) part. Also, this model is usually referred as ARMA (p, q). In which p and q are the order of AR and MA respectively.

Autoregressive (AR) models are based on the idea that the current value of the series, x_t can be explained as a function of p past values: $x_{t-1}, x_{t-2} \dots x_{t-p}$ where p determines the number of steps into the past needed to predict the current value. An autoregressive model of order p , abbreviated AR (p), is of the form:

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + \dots + \beta_p x_{t-p} + w_t \quad 4.8$$

Where x_t is stationary. $\beta_1 \dots \beta_p$ are constants ($q \neq 0$). w_t is a Gaussian white noise series with mean zero.

The moving average (MA) model is an alternative to the autoregressive representation in which the x_t on the left-hand side of the equation are assumed to be combined linearly, while the moving average model of order q , abbreviated as MA (q), assumes the white noise w_t on the right-hand side of the defining equation are combined linearly to form the observed data.

The moving average model of order q , or MA (q) model, is defined to be:

$$x_t = \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q} + w_t \quad 4.9$$

Now we mix the autoregressive and moving average to get the ARMA model. The definition for the model is as follow:

A time series $\{x_t; t = 0, \pm 1, \pm 2 \dots\}$ is ARMA (p, q) if it is stationary and:

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + \cdots + \beta_p x_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q} \quad 4.10$$

The parameters p and q are called the autoregressive and the moving average orders, respectively. $\{w_t; t = 0, \pm 1, \pm 2 \dots\}$ is a Gaussian white noise sequence.

To determine the order of ARMA model, we need to calculate partial correlation and auto correlation of the series. The plot for partial correlation and auto correlation with different lags are as follows:

Figure 4.7 The partial correlation of the residual series

Figure 4.8 The auto correlation of the residual series

The figures above apparently show that both the partial correlation and auto correlation decrease sharply after 1 lag. Therefore, the p and q order for ARMA should both be 1.

The Akaike information criterion (AIC) can also be applied to decide the order of ARMA model. AIC is a measure of the goodness of fitting an estimated model. It is based on the concept of entropy. Entropy is a measure of the information lost when a mathematical model is used to describe the actual data. AIC is a powerful tool for model selection. The model with the lowest AIC has the best performance.

The AIC is defined by the following equation:

$$AIC = \log V + \frac{2d}{N} \quad 4.11$$

V is the loss function, d is the number of estimated parameters, and N is the number of values in the estimation data set.

The loss function V is:

$$V = \det\left(\frac{1}{N} \sum_{t=1}^N \varepsilon(t, \theta_N) (\varepsilon(t, \theta_N))^T\right) \quad \text{(Here } T \text{ means transpose of the matrix)} \quad 4.12$$

θ_N represents the estimated parameters.

Figure 4.9 is the AIC of ARMA model with different order.

Figure 4.9 The AIC of ARMA with different order

From the figure, we see that the result is the same as what was found in the partial correlation and auto correlation plot. ARMA (1, 1) has the lowest AIC. Thus, that is the best order for ARMA model.

4.4.2 Time Delay Neural Network

Time Delay Neural Network (TDNN) has been introduced in section 3.3. It is developed from general feed forward neural network to obtain the relationship between the input and output position in time series. The conventional neurons of a neural network provide their response to the weighted sum of the current inputs. But for TDNN, it extends the sum to a finite number of past inputs. In this way, the output provided by a given layer depends on the output of the previous layers computed based on the temporal domain of input values. Because of the very similar structure of the TDNN and the general MLP, back-propagation

with some modifications can be applied to train the TDNN. The structure of TDNN is shown in Figure 4.10.

As in the classical NN, the TDNN also has a training phase. There are basically two different methods of training and updating the weights. In the first method, weights are updated for each of the input patterns using an iteration method. In the second method, an overall error for all the input output patterns of training sets is calculated once the weights and biases are obtained.

There are several methods to update the weights. Four most popular methods are: (1) Basic back Propagation algorithm (BP), (2) Quasi-Newton algorithm (QN), (3) Levenberg-Marquard algorithm (LM), (4) Conjugate Gradient algorithm (CG). The detail of these algorithms is introduced in section 3.3.

Figure 4.10 The architecture of TDNN

4.4.3 Hybrid Model

Both ARMA and TDNN can be used in predicting time series. But none of them is a universal model that addresses both linear and nonlinear problems. ARMA model has been proven to be suitable for linear problems, but be inadequate for complex nonlinear problems. On the other hand, using TDNN to solve linear problem may yield mixed results. Some cases showed that if there are outliers in the data, the networks could easily out-perform linear regression models. Some other cases also found that the performance of TDNN for linear regression problems depends on the sample size and noise level. Therefore, it is not wise to use TDNN blindly to any type of data, since it is difficult to know all the characteristics of the data in an actual problem. A hybrid model that combines both these models would then be able to capture both the linear and nonlinear aspects.

We assume that the daily solar radiation series is composed by linear and nonlinear component [118].

Then, it should be:

$$Y_t = L_t + N_t \quad 4.13$$

L_t denotes linear component and N_t denotes nonlinear component.

First of all, we used ARMA model to fit the linear component. This means the residual series should contain nonlinear component only. Then we used TDNN to model the nonlinear component lying in the residual. This hybrid model has the potential to harness the unique feature and strength of both models.

4.5 Simulation of Prediction Models

The solar radiation data we use in the simulation is obtained from the observation station in Nanyang Technological University. Interested readers can obtain the data from the website:

http://nwsp.ntu.edu.sg/weather/new_portal_intro.htm. We have used the data of 2010 to evaluate the performance of different prediction model. A sample is taken every 5 minutes.

Figure 4.11 The solar radiation of February 2010

During our investigations, we found that the solar radiation received from 22:00 to 07:00 is constantly zero in Singapore. Thus we only took the sample between 07:00 and 20:00 into consideration.

Our goal in the simulation is to explore the optimal model to predict the hourly solar radiation of a specified day. Hence, we used all the prediction models to predict the solar radiation 5 minute ahead in a specified day.

First, the ARMA model is applied to do the prediction and the Al-Sarah's model to detrend the solar radiation series. Then, the data of 1st Feb 2010 is used to train the ARMA (I, I) model. After training, the ARMA model is used to do prediction of the data of 2nd Feb 2010. The prediction figure is in Figure 4.12.

In the figure, the dot line is the predicted series; the solid line is the actual series. We can see that the ARMA model can generally predict the trend of the solar series. But there is some "lag" in the prediction.

Next, the Time Delay Neural Network is used in the simulation. In all the TDNN structure, the sigmoid function was finally adopted and all the popular training methods were tested

with various time delay and neuron sizes. It was experimentally found that the fastest convergence with the smallest prediction error was obtained by using the LM method in the training phase 3-step time delay and 5 neurons in the hidden layer.

The comparison of prediction by TDNN and the actual data is shown in Figure 4.13. The training data is 1st Feb 2010, and the prediction data is 2nd Feb 2010.

Figure 4.12 The actual and predicted series by ARMA (1,1)

Figure 4.13 The actual and predicted series by TDNN

In Figure 4.13, the dot line is predicted series and solid line is actual series. The figure shows that TDNN doesn't have obvious lag as ARMA. But it cannot catch the peak of the solar radiation when it fluctuates.

Finally, we used the hybrid model of ARMA and TDNN to do the prediction, using the same set of data of 1st Feb 2010 to train the model and the data of 2nd Feb 2010 to verify the prediction performance. The comparison of actual and predicted series is presented in Figure 4.14. The dot line is the predicted series and the solid line is the actual series.

The prediction of hybrid model is more accurate than using the ARMA or TDNN model separately.

Figure 4.14 The actual and predicted series by hybrid model

4.6 Comparative Analysis

4.6.1 Comparative Analysis of Different Algorithms

The result of the experiment with data from just a day is not enough. To better verify the performance of these models, we applied 12 months (Jan, 2010- Dec, 2010)' data to do the

prediction and to confirm the result. These data are first detrended by Al-Sadah's model, then 3 different prediction models are applied the residual data. ARMA (1, 1) is chosen as the proper ARMA model. The data of the previous day was used to train the ARMA (1, 1) model. The trained model was used to predict the next day's data. For TDNN and hybrid model, we also use the same approach; the previous day's solar energy data was used to train the model, and then applied the trained model to predict the series of the day after.

RMSE are used to evaluate the performance of prediction. The RMSE of ARMA, TDNN and Hybrid model are calculated. To compare the RMSE of different models more intuitively, we plot the result of 12 months (To facilitate visualize, we take the interval of 2 days) which spread over a year in the same coordinate system. The RMSE of these models are as shown from Figure 4.15 to Figure 4.16.

It is easily observed that although the overall performance of TDNN is better than ARMA, however, it is not stable. Sometimes, the prediction is close to the actual situation but at other times a huge error can occur. The reason for this phenomenon might be attributed to the sensitivity of the TDNN model to the training data. This would explain why TDNN is able to capture the trend of time series without lagging, yet when the training data set is not properly pre-processed, it would affect the prediction performance. On the other hand, the ARMA model is a comparatively stable algorithm. The prediction accuracy of ARMA for everyday is very consistent. However, in most cases, the ARMA model cannot present performance as good as TDNN. In comparison, the hybrid model's performance is the best in most cases. It may not be as stable as the ARMA model, but the consistency of hybrid model is much better than the TDNN. Thus, we can see that the hybrid model has harnessed the advantages of both the TDNN and the ARMA model. Although the hybrid model doesn't always yield the best performance in prediction, it can maintain the consistency and accuracy of its performance at a high level.

Table 4.3 Mean of SMAPE of different prediction models

	Hybrid model	TDNN	ARMA
Jan	28.1	30.0	41.4
Feb	37.0	33.7	45.3
Mar	52.4	55.4	63.5
Apr	33.3	37.7	45.5
May	24.6	24.7	38.3
June	25.8	37.4	47.3
July	34.5	34.2	46.1
Aug	31.5	33.7	38.4
Sep	34.1	37.1	40.4
Oct	28.4	25.7	29.2
Nov	27.4	37.3	41.1
Dec	27.2	29.4	39.5

Table 4.4 Variance of SMAPE of different prediction models

	Hybrid model	TDNN	ARMA
Jan	79.2	416	290
Feb	206	302	363
Mar	677	804	643
Apr	118	684	734
May	68.4	156	135
June	230	575	230
July	101	542	587
Aug	105	524	276
Sep	125	509	425
Oct	54.9	448	130
Nov	313	540	585
Dec	128	216	135

Table 4.5 RMSE of different prediction models

	Hybrid model	TDNN	ARMA
Jan	130	137	137
Feb	172	174	206
Mar	150	177	174
Apr	124	141	170
May	118	119	152
June	105	104	146
July	106	135	136
Aug	103	153	131
Sep	113	180	150
Oct	88.1	128	118
Nov	94.8	118	118
Dec	125	191	146

Symmetric mean absolute percentage error (SMAPE) is again used along with RMSE to assess the performance of different algorithms. The mean value of SMAPE for different months is presented in Table 4.3 and the variance of SMAPE is presented in Table 4.4. It shows that the proposed hybrid model presents better accuracy than the other two in most months. Furthermore, the variance of hybrid model is lower than TDNN and ARMA in all months. That means the performance of hybrid model is more stable and consistency than the others.

The RMSE of different month is presented in Table 4.5. The result of RMSE is very similar to SMAPE. The performance of hybrid model is better than others in most of the time.

Figure 4.15. The RMSE of the prediction of different models on first half year's solar radiation

Figure 4.16. The RMSE of the prediction of different models on second half year's solar radiation

4.6.2 Comparative Analysis of Prediction Performance of Different Daily Integral Solar Radiation

To explore the effect of weather condition on the prediction models, we plot the SMAPE of different models along with different level of daily integral solar radiation. The day with high energy level can be regarded as a clear day. Day with low energy level usually indicates bad weather condition like a cloudy or rainy day. There is clearly a down trend of the SMAPE as the daily integral solar energy increases. The SMAPE is extremely high when the daily integral solar energy falls below 60 KW/m^2 . Thus we come to the conclusion that the prediction models perform very well in clear days, but when weather condition is not ideal, the accuracy of prediction models decrease. Typically, for a clear day, the daily integral falls in the range of 150 KW/m^2 to 300 KW/m^2 with acceptable SMAPE.

Figure 4.17 is the comparison of ARMA and hybrid model in different daily integral solar radiation. It clearly shows that the hybrid model outperforms ARMA in different level of daily integral. This is true especially in the range from 150 KW/m^2 to 300 KW/m^2 .

Figure 4.18 is the comparison of TDNN and hybrid model. From this figure we can see that although TDNN present better performance than hybrid model occasionally, its prediction sometimes yields very high SMAPE. On the contrary, the hybrid model presents very good consistency compare to TDNN.

Table 4.6 presents the mean value of SMAPE of different prediction models in different level of daily integral. Its result is in accordance with Figure 4.17 and Figure 4.18. Hybrid model present better accuracy and consistency compare to other algorithms.

Improving the accuracy of prediction models under bad weather condition is an important problem which requires further studying.

Table 4.6 Mean value of SMAPE of different level of daily integral solar radiation

Daily integral solar radiation (KW/m ²)	Hybrid model	TDNN	ARMA
0-50	69.0	103.9	106.8
5-100	38.0	64.4	65.1
100-150	30.9	42.2	49.9
150-200	29.5	29.3	43.1
200-250	31.6	32.1	42.7
250-300	28.5	29.1	39.2
300-350	27.5	30.3	33.1
350-400	24.7	24.6	16.1

Figure 4.17 The comparison of prediction of ARMA & hybrid model in different level of daily integral solar radiation

Figure 4.18 The comparison of prediction of TDNN & hybrid model in different level of daily integral solar radiation

4.7 Summary

In this chapter, we compared the detrending performance of several empirical hourly solar radiation models. To access the performance of detrending, ADF test is used to measure the stationarity of the detrended series. Also we have used RMSE and NRMSE to measure the goodness of curve fitting of these different models. Coincidentally, Al-Sadah's model's performance is the best in both aspects.

After detrending phase, ARMA is applied to conduct prediction on the stationary series. To decide the order of the ARMA model, auto correlation and partial correlation of the residual series is studied. The result shows the best order is ARMA (1, 1). An inspection with AIC provides the same result. Therefore, ARMA (1, 1) is selected as prediction model. The TDNN model was also used in this experiment. Just as discussed previously, it is capable to capture the fast changes in the series, but sometime it also produces huge error. To take the advantages of ARMA and TDNN, we used a hybrid schema to combine both. The hybrid model uses ARMA model to predict the linear component of the series and the TDNN model to predict the nonlinear component.

The same data set is applied to the hybrid model to evaluate its performance. Our conclusion is that the prediction of the hybrid model is much better than individual ARMA model or TDNN model since it is stable and accurate. This is due to the fact that the solar radiation series contain both linear and nonlinear component.

Chapter 5. A Novel Multi Model Framework for Solar Radiation Time Series Prediction

5.1 Introduction

In last chapter, we have used ARIMA, TDNN and hybrid model in the solar radiation time series prediction. Their performances were evaluated and the hybrid model is the best.

In this chapter, a novel framework is presented to conduct short term prediction of solar radiation time series. Solar radiation data is the only input parameter needed. The solar radiation is regarded as stochastic time series. We assume that there exists several different patterns embedded in it. The value of the time series relates not only to time, but also to the pattern it belongs to. It is then segmented into subsequence. The subsequence is grouped into different clusters. Training and building prediction models are performed with the data of each cluster. Hence, when predicting the future trend of time series, the label of cluster that the current time series belongs to is firstly determined and then corresponding model is applied to do the prediction.

Daily solar radiation data of Singapore is used in the experiment. The data is collected by an observation station located in Nanyang Technological University, Singapore and can be obtained online. To ensure the accuracy and adaptability of the framework, 24 months' data is used in the whole process. The data of 2009 is used to train and tune the framework. Then the trained framework is used to predict solar radiation data of 2010.

The rest of the chapter is organized as follow: Section 5.2 details the building of the MMF (multi-model framework) and the related algorithm. Section 5.3 discusses and analyzes the result of experiments and section 5.4 is the comparison with other algorithms. Section 5.5 is the conclusion.

5.2 The Proposed Methodology

The proposed multi-model prediction framework contains two different phases: the clustering phase and the prediction phase.

In the clustering phase, the solar radiation series is segmented into subsequence of fixed length. K-means algorithm is utilized to cluster the segmented data. To improve the accuracy of clustering, dynamic time warp (DTW) is adopted as the distance measurement. Then TDNN is used as prediction model for further processing. The data set of each group is applied to train TDNN separately to get their own prediction model. In the predicting phase, the pattern for current time is identified first, and then the actual data is fed to the appropriate model to predict the future trend of solar radiation series.

The basic flow of the methodology is shown in Figure 5.1.

Figure 5.1 the flow of Multi-Model Framework

In this chapter, the problem of short term instant prediction is addressed. The exploration of using MMF for long term predicted will be left for future work.

5.2.1 Clustering

The clustering phase in this proposed prediction framework is of great importance. The prediction performance highly related to the accuracy of clustering. Since the solar radiation time series will frequently switch from one pattern to another due to changes in the weather, it must be segmented first. Ideally, each subsequence should belong to a specific pattern.

a) Segmentation

The ideal segment point is when the pattern of solar radiation switches. However, finding every pattern changing point for the series is very difficult. Several researchers have addressed this problem. They have introduced segment-based approach for sub-sequence searches (SBASS) [119]. Feature point segmentation (FPS) is that segments the time series based on so called feature point [120]. However, none of these schemas presents satisfactory performance in our experiment. And subsequences of varied length also bring about difficulty for clustering. Thus, we adopted fixed length segmentation in our framework. The problem is thus converted to finding the optimal length L for segmentation. Another factor which will affect the clustering performance is the number of clusters. We will discuss how to find the optimal L in the experiment section.

b) Clustering Algorithm

The performance and applicability of different cluster algorithms are widely discussed in literature [121]. Apparently, there is no universal solution for all large scale dataset. Researchers have claimed that the well-known cluster algorithm K-means is competent in clustering energy time series data set [122]. Therefore, K-means is utilized to cluster the solar radiation time series.

K-means algorithm requires user to define the number of clusters in advance. And this number will largely affect the goodness of clustering. Finding optimal number of cluster K will be discussed in the experiment section.

c) Dynamic Time Warp

Finding proper distance measurement is another challenge of the K-means algorithm. K-means algorithms aims to minimize the distance between elements in the same group while maximizing it between different groups. Adopting different measurement of distance will bring about different clustering result. Most cluster algorithms adopt the Euclidean distance. However, Euclidean distance has its limitation in identifying time series. Another disadvantage of using Euclidean distance as distance measure is that the result of clusters lack robustness, especially when the data is corrupted by noise data.

Dynamic time warping (DTW) was firstly used as similarity measurement in text data matching and image pattern recognition and showed its usefulness in pattern identification and recognition of distorted data sequence. Researchers also have introduced this algorithm in time series analysis and obtained good experiment result [123]. Here DTW is used to measure the similarity in K-means in this experiment.

Below is a brief introduction of DTW. Assume there are two time series X and Y , of length of n and m respectively. Then it can be represented as:

$$X = x_1, x_2, \dots x_n$$

$$Y = y_1, y_2, \dots y_m$$

A distance matrix D is first defined to compare these two series. $D(x_i, y_j)$ is the distance between x_i and y_j , which can be regarded as a measure of dissimilarity between these two points.

$$D = \begin{bmatrix} d(x_1, y_m) & \cdots & d(x_n, y_m) \\ \vdots & \ddots & \vdots \\ d(x_1, y_1) & \cdots & d(x_n, y_1) \end{bmatrix} \quad 5.1$$

The element $d(x_i, y_i)$ in matrix D can be regarded as the measurement of dissimilarity of two points in these two time series.

The warp path is a set of independent elements which forms a consecutive path in the

dissimilarity matrix.

$$W = w_1, w_2, \dots, w_n \quad 5.2$$

w_i means the i th steps of the path. From the definition, the path must meet the following condition:

Boundedness: $\max(m, n) \leq K \leq m + n - 1$

Boundary condition: $w_1 = D(x_1, y_1)$ and $w_k = D(x_n, y_m)$

Continuity condition: For any $w_i = D(x_a, y_b)$ and $w_{i+1} = D(x_{a1}, y_{b1})$ must have $a1 - a \leq 1$ & $b1 - b \leq 1$

Monotonic condition: For any $w_i = D(x_a, y_b)$ and $w_{i+1} = D(x_{a1}, y_{b1})$ must have $a1 - a \geq 0$ & $b1 - b \geq 0$

With these conditions above, the number of paths is restricted. For instance, with the monotonic constraint, if a path has already moved one element up in the distance matrix D then its next move must be either diagonally or to the right to meet the rules.

To improve the efficiency and exclude any meaningless result of the DTW algorithm, instead of finding all possible routes through the matrix, the DTW algorithm works by keeping track of the cost of the best path to each point in the grid. During the matching process the path of the lowest cost cannot be known beforehand; but it can be traced back after reached the end point.

As there are many different warp paths in the matrix, the one with minimal length is used as the measurement of distance of the input time series. This can be represented as:

$$DTW(X, Y) = \min_k \left(\frac{1}{k} \sum_{i=1}^k w_i \right) \quad 5.3$$

d) Evaluation Index

To properly select length L for subsequence and number K for cluster, three well-known validity indices: Silhouette index [124], Dunn index [125],

Davies-Bouldin index [126] which have been proved to be capable in the evaluation of time series clustering are applied to various numbers of clusters.

The Silhouette index is a measure for the accuracy of clustering. $a(i)$ represents the average distance of certain object i belong to cluster A to all the objects in A and $d(i, C)$ represents the average distance of that object to all the cluster which $C \neq A$, with $d(i, C)$ we can compute:

$$b(i) = \min d(i, C) \text{ with } i \in A \quad 5.4$$

The value $b(i)$ measures the distance of the object i to its nearest neighbor cluster. Thus, the index can be computed by the following equation:

$$\text{Silhouette}(i) = \frac{a(i) - b(i)}{\max\{a(i), b(i)\}} \quad 5.5$$

The index should range from -1 to $+1$, where $+1$ means objet i is attached to an adequate cluster and -1 means that object is attached to an inadequate cluster. The optimal number of cluster is the one which maximize the index. In our simulation, the mean value of silhouette index of all training data is computed to find the optimal number of cluster.

Dunn index is another index can serve as a validation tool. The index for K clusters C with $i=1, \dots, K$ can be defined by:

$$DU = \min_i \min_{i \neq j} f_{i,j} \quad 5.6$$

Where

$$f_{i,j} = \frac{d(C_i, C_j)}{\max_m \text{diam}(C_m)} \quad 5.7$$

$d(C_i, C_j)$ is the dissimilarity of clusters C_i and C_j , it can be defined as:

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} ||x - y|| \quad 5.8$$

And $diam$ is the diameter of the cluster, it is defined as:

$$diam(C_m) = \max_{x,y \in C} ||x - y|| \quad 5.9$$

$|| \cdot ||$ Means norm. The optimal combination is when the index is maximized.

The Davies-Bouldin index can also be used to evaluate the performance of clustering. For K clusters, the index is defined as:

$$DB_k = \frac{1}{K} \sum_1^K \max_{j \neq i} f_{i,j} \quad 5.10$$

Where:

$$f_{i,j} = \frac{diam(C_i) + diam(C_j)}{d(C_i, C_j)} \quad 5.11$$

And diameter of a cluster is defined as:

$$diam(C_i) = (\frac{1}{n_i} \sum_{x \in C_i} ||x - z_i||^2)^{\frac{1}{2}} \quad 5.12$$

The optimal combination is achieved when the index is minimized.

5.2.2 Prediction Phase

a) Prediction Model

There is a thorough review of recent development on time series prediction [127]. Instead of using traditional classical statistical prediction method, other data mining paradigms, such as time delay neural network (TDNN) has proved to be effective in capturing the underlying non-linear relationship of time series. Study shows that models with neural network structure provide better performance than other algorithms in non-linear time series prediction [128]. Thus, TDNN is selected as the prediction model for further training.

TDNN is developed from general feed forward neural network to obtain the nonlinear relationship between the input and output in time series. The conventional neurons of neural network respond to the weighted sum of the current inputs. But for TDNN, it responds to a finite number of past inputs. In this way, the output provided by a given layer depends on the output of the previous layers computed based on the temporal domain of input values. Because of the similarity between TDNN and traditional neural network, back-propagation with some modifications can be applied to train the TDNN.

As in the classical artificial neural network (ANN), the TDNN also has a training phase. There are basically two different methods of training and updating the weights. In the first method, weights are updated for each of the input patterns over a fixed number of iterations. In the second method, an overall error for all the input-output patterns of training sets is calculated once the weights and biases are obtained. This training stops once the overall error is below the predefined value.

The data set of each cluster is used to train a unique TDNN model. Therefore, each TDNN model represents a specific pattern of solar radiation time series. When predicting the future data of solar radiation series, which pattern is appropriate for current time should be firstly determined. Therefore, a pattern identification process is needed.

b) Pattern Identification

The pattern identification procedure aims to select proper prediction model for current time. Assume:

$$Y_t = \begin{cases} f(M_1, s) & \text{if it should be model 1} \\ \cdot & \\ \cdot & \\ \cdot & \\ f(M_n, s) & \text{if it should be model } n \end{cases} \quad 5.13$$

s is the actual time series data and M_n means the prediction model we adopt. Y_t is our prediction at time t .

To identify the appropriate prediction model for current time, an error function is constructed:

$$J_t = \sum_{i=t-w}^t (y_i - \hat{y}_i)^2 \quad 5.14$$

y_i is the actual data at time i while $\hat{y}_i$ is the prediction at that time. When selecting appropriate prediction model at time t , the prediction models of all clusters are fed with the data w step back. Then use equation 5.14 to compute the error function J_t . The model with minimal J_t will be regarded as most appropriate for current time.

The length of w , which is called window size of the pattern identification, is then explored. In order to find the optimal length of window size, cross validation experiment is conducted on different w .

5.3 Simulation

To assess the performance of the methodology discussed above, the data of solar radiation obtained in 2009 and 2010 in Singapore is used in the simulation. The data is recorded by the observation station located in Nanyang Technological University (NTU) and can be obtained from the website: <http://nwsp.ntu.edu.sg>. CMP 3 pyranometer is used to record the solar radiation. The original sample rate is one recording per minute. To reduce the effect of noise, outlier and missing data, we take an average preprocess and generate a new series. Each data point of the new series spaced 5 minutes apart. As we focus on short term prediction in this experiment, our task is to predict the value 5 minutes ahead.

Makridakish and Hibon proposed a new performance metric symmetric mean absolute percentage error (SMAPE) to account for a different number of observations in the individual data sub-samples of training and test set, and the different scale between individual series. It is also established best-practice in industry and in previous time series prediction competitions such as NN3 time series forecasting competition [115]. Its definition is given in equation 3.15.

5.3.1 Segmentation and Clustering

The length of subsequence and the number of cluster both contribute to the accuracy of clustering so different combination of them will be explored and evaluated with the indices discussed above. The segment length of time series ranges from 40 minutes to 80 minutes while the number of clusters varies from 2 to 15.

Three different indices were used to evaluate the accuracy of clustering. The procedure to select a proper set of segment and number is under study.

In the tests of different index, the best five combinations were presented in Table 5.1. The parameters in the bracket were length of segment and number of clusters respectively. For example, (45, 2) means during the process of clustering, the length of segment is 45 minutes and number of clusters is 2.

Table 5.1 The best 5 parameter set of segment length and cluster number

Ranking (best to worst)		1	2	3	4	5
Silhouette	Segment length and cluster number	(45,2)	(70,4)	(65,4)	(80,2)	(65,5)
	Index value	0.193	0.185	0.184	0.180	0.176
Davies-Bouldin	Segment length and cluster number	(70,4)	(70,3)	(45,9)	(55,12)	(70,11)
	Index value	0.209	0.306	0.441	0.533	0.629
Dunn	Segment length and cluster number	(80,4)	(80,2)	(70,4)	(70,5)	(80,7)
	Index value	0.045	0.039	0.036	0.310	0.028

Table 5.1 shows that the optimal sets in three tests are different and none of them coincide. However, both DB index and DU indicate cluster of 4 is the optimal number for clustering. The length of segment also varies in different test. It is noted that the length of 70 minutes appears the most often in Table 5.1 (2nd best in Silhouette, 1st 2nd 5th best in DB, 3rd 4th best in DU). Moreover, the set of (70, 4) is the 2nd best in Silhouette, 1st best in DB, 3rd best in DU, respectively. Thus, (70, 4) is selected as the parameter to be used in clustering.

5.3.2 Training Prediction Models

Upon completing the segmentation and clustering process, the next step is training and

building prediction model for each cluster.

TDNN is used for further training because of its great flexibility and capability in capturing the nonlinear relationship inherent in the series. As presented in [114], for the TDNN structure, the sigmoid function has been proved to be optimal. All the popular training methods were tested with various time delay and neuron sizes. Just as discussed in previous chapter, we still adopt Levenberg-Marquard method in the training phase. The structure of 3-step time delay and five neurons in the hidden layer is again used.

The data set is used to train the TDNN separately on each cluster. Four different prediction models on the pattern are obtained. Each model represents a specific pattern of the stochastic component of solar radiation. When predicting the solar radiation, we firstly decide which model is most appropriate for current time then use it to conduct the prediction. To better diversify the candidate models and improve the prediction accuracy, a TDNN model which is trained by the data from the last day (one day before the day to be predicted) is added to the candidate prediction model set. And when none of the four patterns is appropriate for current time, this additional model will be adopted to do the prediction.

5.3.3 Pattern Identification Procedure

The window size w of the pattern identification phase affects this prediction performance significantly. To find the optimal window size, a 12-fold cross-validation experiment is performed with various window sizes.

In the cross validation experiment, solar radiation data of the year 2009 is divided into 12 different groups according to month. For each experiment, a group is chosen as validating data while the rest is used as training data. The experiment is repeated 12 times. Each time a different group is used as validation data to ensure the integrity and robustness of the prediction performance. Combined with a varied w , the cross validation experiment's result is shown in Table 5.2. It shows that the optimal window size varies over data used from different month.

Table 5.2 The monthly mean value of SMAPE of different window size

	1	2	3	4	5	6	7	8	9	10	11	12
Jan	23.0	23.4	20.0	21.5	20.3	21.7	21.8	21.8	24.6	24.6	24.7	23.6
Feb	29.7	26.6	25.8	26.7	24.6	27.3	26.4	26.4	27.2	26.3	26.3	27.0
Mar	37.0	37.1	38.5	36.6	40.6	34.2	39.4	39.1	39.2	40.3	41.9	40.5
Apr	23.6	21.4	21.9	20.8	20.6	21.6	19.8	23.6	22.8	21.6	22.8	22.9
May	22.8	20.5	21.7	19.1	20.1	21.8	19.7	21.7	21.8	22.2	22.7	23.6
June	28.3	25.1	25.3	25.9	25.5	24.9	26.6	27.9	28.6	27.4	28.4	28.1
July	25.4	22.6	21.2	22.3	22.1	23.1	23.6	23.5	25.3	27.1	25.4	27.1
Aug	20.6	19.5	19.0	20.3	19.2	20.9	21.9	23.2	22.6	22.2	22.9	22.0
Sep	24.4	23.2	23.1	23.5	24.2	22.8	24.1	24.5	24.1	24.2	25.8	25.9
Oct	17.9	16.1	16.3	17.4	17.7	16.4	18.5	18.8	17.4	18.6	19.1	20.5
Nov	25.0	28.1	22.2	30.5	28.1	24.2	20.7	23.8	28.2	31.6	19.8	29.4
Dec	22.2	37.0	25.0	24.8	26.9	14.0	18.5	16.4	26.0	33.3	37.4	49.1
Mean	25.0	25.1	23.3	24.1	24.1	22.7	23.4	24.2	25.7	26.6	26.4	28.3

A value of 6 for the window size presents the smallest SMAPE in most of the months. It also gives the best performance from an overall view on the whole year's data. Thus, a window size of 6 is experimentally determined to be optimal and will be used in the prediction.

5.3.4 Prediction Result

In this section, the trained framework is applied to predict the short term trend of daily solar radiation series. The data source is the solar radiation data of 2009 and 2010 in Singapore. One sample for every 5 minutes is taken to form the time series. We have used the data of 2009 to train the framework and obtained four different patterns. When predicting the data at time Y_t , the data of $Y_{t-1}, \dots, Y_{t-6}$ is applied to equation 5.13 to determine which pattern that current data belongs to. Then the corresponding TDNN is used to do prediction. If no pattern is appropriate for current time, the TDNN model which is trained by yesterday's data is used to do prediction.

On the contrary to Figure 5.2, Figure 5.3 presents the prediction of less ideal weather condition.

A prediction of a clear day by MMF is presented in Figure 5.2. Because the weather condition is clear at that day, the solar radiation series goes smoothly. MMF algorithm presents accurate prediction. The SMAPE for MMF is 3.1.

As can be seen in Figure 5.3, the solar radiation in 2nd February fluctuates up and down frequently. This may due to the corruption of noise. The SMAPE for MMF is 55.3. Since the TDNN model aims to capture the relationship lying in time series and using past data to predict future, it display a smoothing effect the time series when it changing frequently. That's why the variability of predicted series is less than the actual series.

Figure 5.2 The prediction of MMF on July 28th

Figure 5.3 The prediction of MMF on Feb 2nd

The prediction performance of solar radiation of year 2010 is presented in Table 5.3. To better evaluate the general performance, we present both mean and variance values of SMAPE of different month.

Table 5.3 Mean and variance of SMAPE of prediction of different month

	mean of SMAPE	variance of SMAPE
Jan	21.5	89.7
Feb	24.8	106.2
Mar	33.5	184.8
Apr	21.2	94.5
May	17.7	57.5
June	23.0	70.5
July	19.2	93.8
Aug	17.2	56.7
Sep	20.8	61.6
Oct	15.4	37.8
Nov	16.6	94.8
Dec	22.5	81.5

Table 5.3 shows that the mean of SMAPE of all the different month are below 34 which is a very accurate prediction result. The best result is achieved in the prediction of solar radiation of October which is 15.4, and the worst is achieved in March which is 33.5. The same thing happened in the variance column, the prediction of October presents the lowest variance 37.8 while the prediction of March presents the highest variance 184.8.

5.4 Comparative Analysis

5.4.1 Comparative Analysis of Different Algorithms

As presented in [129], ARMA and TDNN have been proven to be satisfactory in predicting the solar radiation of Singapore. We therefore compare the prediction result of MMF to ARMA and TDNN to assess its performance.

Hybrid model which combines ARMA and TDNN can provide better performance

than the individual model [130]. Thus it is also incorporated in the comparison experiment.

The mean of SMAPE of prediction of the different months' data was presented in Table 5.4. And the variance was presented in Table 5.5.

Table 5.4 Mean of SMAPE of different prediction models

	MMF	Hybrid model	TDNN	ARMA
Jan	21.5	28.1	30.0	41.4
Feb	24.8	37.0	33.7	45.3
Mar	33.5	52.4	55.4	63.5
Apr	21.2	33.3	37.7	45.5
May	17.7	24.6	24.7	38.3
June	23.0	25.8	37.4	47.3
July	19.2	34.5	34.2	46.1
Aug	17.2	31.5	33.7	38.4
Sep	20.8	34.1	37.1	40.4
Oct	15.4	28.4	25.7	29.2
Nov	16.6	27.4	37.3	41.1
Dec	22.5	27.2	29.4	39.5

Table 5.5 variance of SMAPE of different prediction models

	MMF	Hybrid model	TDNN	ARMA
Jan	89.7	79.2	416	290
Feb	106	206	302	363
Mar	185	677	804	643
Apr	95	118	684	734
May	57.5	68.4	156	135
June	70.5	230	575	230
July	93.8	101	542	587
Aug	56.7	105	524	276
Sep	61.6	125	509	425
Oct	37.8	54.9	448	130
Nov	94.8	313	540	585
Dec	81.5	128	216	135

Table 5.4 shows that the proposed MMF algorithm obtains much better prediction result than all the other algorithms in every month. Compared to the hybrid model which has been proved to be capable in predicting the solar radiation, MMF algorithm achieves greater improvement. Especially in October, the mean of SMAPE improve 45.7%. Even in June, the prediction performance of MMF is very close to hybrid model, but it still achieves 11.0% improvement in prediction. With the data of March, Hybrid model, TDNN, ARMA all present very high mean of SMAPE which is 52.4, 55.4, 63.4 respectively. The MMF algorithm also presents the highest mean of SMAPE

compare to its prediction of other months but maintain it below 34.

Table 5.6 RMSE of different prediction models

	MMF	Hybrid model	TDNN	ARMA
Jan	122	130	137	137
Feb	161	172	174	206
Mar	138	150	177	174
Apr	109	124	141	170
May	111	118	119	152
June	91.0	105	104	146
July	101	106	135	136
Aug	90.4	103	153	131
Sep	99.0	113	180	150
Oct	81.1	88.1	128	118
Nov	85.4	94.8	118	118
Dec	109	125	191	146

Table 5.5 shows that in most cases, the variance of MMF is obviously lower than the other three algorithms. This means that MMF provides a much more consistent prediction performance.

We also use another metric root mean square error (RMSE) to evaluate the performance of different prediction models. The result is shown in Table 5.6. The result also shows MMF outperforms other models in different month.

5.4.2 Comparative Analysis of Prediction Performance of Different Daily Integral Solar Radiation

To further compare the performance of different algorithms, we also explore the prediction accuracy of different level of daily integral solar radiation. High energy level usually occurred on a clear day while days with low energy level may be indication of bad weather condition such as cloudy or rainy day. There is clearly a down trend of the SMAPE as the daily integral solar energy increases. The SMAPE is extremely high when the daily integral solar energy falls below 50 KW/m².

Table 5.7 Mean value of SMAPE of different level of daily integral solar radiation

Daily integral solar radiation (KW/m ²)	MMF	Hybrid model	TDNN	ARMA
0-50	67.0	69.0	103	106
50-100	29.9	38.0	64.4	65.1
100-150	20.4	30.9	42.2	49.9
150-200	20.0	29.5	29.3	43.1
200-250	21.0	31.6	32.1	42.7
250-300	19.7	28.5	29.1	39.2
300-350	21.2	27.5	30.3	33.1
350-400	11.8	24.7	24.6	16.1

Figure 5.4 The comparison of prediction of MMF&ARMA in different level of daily integral solar radiation

Figure 5.5 The comparison of prediction of MMF&TDNN in different level of daily integral solar radiation

Figure 5.6 The comparison of prediction of MMF & hybrid model in different level of daily integral solar radiation

Thus we come to the conclusion that the prediction models perform very well on clear days, but when weather condition are not ideal, the accuracy of prediction models decrease. Typically, when the daily integral is in the range of 150 – 400 KW/m² all predictions present better SMAPE than below 150 KW/m².

Figure 5.4 to Figure5.6 present a thorough view of the comparison of the prediction performance of MMF and other prediction models in different level of daily integral solar radiation. It shows clearly that MMF has better performance than 3 other benchmark models in different levels.

This experiment justifies that the prediction performance of the proposed MMF is superior to the other benchmark models in any weather condition.

5.5 Summary

In this chapter, a novel multi-model prediction framework for solar radiation time series is proposed. As described above, the framework starts with the assumption that there are several patterns in the stochastic component of solar radiation series. To extract the pattern embedded in it, the series is segmented into fixed length subsequences. The subsequences are then grouped into different clusters using K-means algorithm. To determine the optimal number of clusters, a 12 fold cross-validation experiment is performed. Three indices are used to evaluate the fitness

of clustering and it has been shown that using four clusters provides the best clustering result. In each cluster, a TDNN is trained for prediction using the respective data to model the corresponding specific pattern. For prediction, the pattern for the current time is firstly identified. This is followed by selecting the appropriate trained TDNN model. Another TDNN model which is trained by the data of last day serves as an additional model. When none of the four patterns is appropriate for current time, the additional model will then be used to conduct the prediction.

To assess the performance of MMF, the solar radiation data of Singapore was applied to it and it provides encouraging performance. The result is compared with other benchmark prediction model such as ARMA, TDNN and hybrid model.

The simulation result shows that MMF can achieve improvement over other models. That justifies our assumption. There are indeed different patterns which appear repeatedly in solar radiation series. We use segment and clustering method to extract them and model them with TDNN algorithm. The advantage of MMF is that it models non-linear relationship of different pattern with different TDNN separately rather than simply using one TDNN to capture the general trend of whole series. When conducting prediction, the pattern that will appear is firstly identified and corresponding prediction model is used to predict the future value. This is a more refined prediction strategy compare to other method. The experiment result also shows MMF outperforms other methods in both accuracy and consistency.

Future work will focus on further improving the clustering accuracy and on finding better model to extract the relationship of data with certain pattern.

Chapter 6. Genetic Algorithm with Multi Model Framework for Solar Radiation Time Series Prediction

6.1 Introduction

In last chapter, we have proposed a novel multi model framework (MMF) to predict daily solar radiation time series. To extract different patterns lying in solar radiation series, it is segmented and the subsequences are grouped into different clusters. A prediction model is trained for each cluster to represent the specific pattern belonging to that cluster. During the prediction phase, the pattern which is suitable for the current time is identified. Then the appropriate model is used to do the prediction.

In the MMF framework, the accuracy of segmentation and clustering is quite important. However, using fixed length of segmentation is an over-simplified approach because a meaningful pattern might be missed if it is split over different subsequence. A segmentation which generates subsequence with variable length is more suitable to be used in the prediction framework. Therefore, genetic algorithm is applied to find the optimal segmentation schema.

To improve the prediction performance of multi-model framework (MMF), supporting vector regression (SVR) is also added to the framework. When none of the patterns proved to be suitable for the current time, SVR is used to do prediction.

To assess the performance of the framework described above, daily solar radiation data of Singapore is applied to it. The data is collected by an observatory station located in Nanyang Technological University, Singapore and can be obtained online. To ensure the accuracy and adaptability of the framework, 24 month data is used in the whole process. The data of 2009 is used to train and tune the framework. Then the trained framework is applied to predict solar radiation data of 2010.

The rest of the chapter is organized as follow: Section 6.2 details the process of building GAMMF and also discusses the related algorithms. Section 6.3 discusses and analyzes the result of experiments and section 6.4 is the comparison of proposed algorithm with other algorithms. Section 6.5 is the summary of our work in this section.

6.2 The Proposed Methodology

6.2.1 Segmentation and Clustering Methodology

During the multi model framework (MMF) mentioned in last chapter, the solar radiation series are segmented and grouped into different clusters. The data of different cluster is used to train a prediction model. In the prediction phase, the appropriate prediction model is identified for the current time, and then the actual data is fed to the selected model to conduct the prediction. The segmentation in MMF adopts a fixed length. This might be oversimplified because the meaningful pattern in the time series may appear with different lengths. A fix length segment will split that pattern across different subsequence. Thus a segment approach with dynamic length is introduced to improve the clustering accuracy.

Since the time series segment problem is so important, there have been a great number of researches related to it [131-133]. This problem can be easily treated as an optimization problem. Genetic algorithm (GA) is a widely used optimization algorithm. Chung et al. has proposed a time series segmentation approach based on GA which enable the user to search pre-defined pattern in time series [134]. Tseng et al. also have proved GA can be used to address the time series segmentation problem to find meaningful pattern [132].

In this chapter, we segment the solar radiation series into subsequence of variable length instead of fixed length. Combined with cluster algorithm, GA is applied to find the optimal segmentation schema. The proposed segmentation algorithm generates a population of chromosomes firstly. Each chromosome represents a possible way of segmentation of the solar radiation series. After the segmentation, K-means is used to group the subsequences into k different clusters. Then each cluster only contains subsequences of similar feature. Before clustering, polynomial interpolation is applied to normalize subsequences. The Silhouette index is used to measure the quality of clustering. For the reason that GA is an iterative optimal algorithm, the goodness of clustering is used as the fitness function to choose the appropriate chromosome for the further mutation and mating. Finally, the optimal chromosome will be obtained. The whole process of finding optimal segment is shown in Figure 6.1. Some details will be discussed in the following sections.

Figure 6.1 The flow chart of using GA to find the optimal segment

a) Chromosome Representation.

The segment must first be encoded as a chromosome. There are two popular encoding methods for GA. One is real-number method; the other is bit-string method. In this thesis, bit-string encoding method is adopted in our framework because it could simulate the segmentation of time series more intuitively.

A time series can be represented by a vector $\{x_1, x_2, \dots, x_k, \dots, x_n\}$, x_k is the k -th point of time series. The chromosome is a bit string which has the same length as the time series. It could be represented as a vector as $\{c_1, c_2, \dots, c_k, \dots, c_n\}$. c_k is the k -th point of chromosome. If $c_k = 1$, it means x_k is a breaking point. For example, if c_3 and c_7 equals one, that means $\{x_3, x_4, x_5, x_6, x_7\}$ is one subsequence generated by the chromosome. Other segments will also be generated in this way. To ensure it represents a meaningful segmentation, there are two constraints for the chromosome. First, the beginning and ending points of the chromosome must

be segment points. Second, there cannot be two adjacent points in chromosome to be one as this will generate a null subsequence. Each chromosome represents a possible segmentation schema for the time series. Then the genetic algorithm will be used to find the optimal chromosome for the segment of the time series.

b) The Initial Population

The genetic algorithm needs a population of chromosome to start the iteration process. Each chromosome is a feasible segmentation for the residual time series. Just as the experiment which is proposed by Tseng et al. [132], This initial chromosomes are randomly generated which satisfy these constraints as mentioned above.

c) Clustering Algorithm

To derive proper pattern of the residual series of solar radiation, K-means algorithm is used to group these subsequences into different cluster. As the subsequences are of variable length, it is difficult to compare the similarity between them. Polynomial interpolation is used to normalize these subsequences. They are converted to pre-defined length. K-means is applied to the normalized subsequences to group them into different clusters. Each cluster represents a pattern underlying in the residual series. The number of clusters K is pre-defined. To evaluate the goodness of clustering, the silhouette index is used. It is defined in Equation 6.2. $a(i)$ represents the average distance of certain object i belong to cluster A to all the objects in A and $d(i, C)$ represents the average distance of that object to all the cluster which $C \neq A$, with $d(i, C)$ we can compute:

$$b(i) = \min d(i, C) \text{ with } i \in A \quad 6.1$$

The value $b(i)$ illustrates the dissimilarity of the object i to its nearest neighbor cluster. Thus, the index can be computed by the following equation:

$$silh(i) = \frac{a(i) - b(i)}{\max\{a(i), b(i)\}} \quad 6.2$$

The index should be ranged from -1 to $+1$, where $+1$ and -1 mean that the object i belongs to an adequate and inadequate cluster respectively. The optimal chromosome is obtained when the index is maximized.

d) Genetic Algorithm

The genetic algorithm is an iterative optimization algorithm. Each time, the parent chromosomes (which are generated by previous iteration) will go through mating and mutation process. In the mating process, a cross point is randomly selected. Two parent chromosomes will exchange their value from the cross point afterwards. For example, two parent chromosomes $C1 = \{1\ 1\ 0\ 1\ 0\ 0\ 0\}$ and $C2 = \{1\ 0\ 0\ 1\ 0\ 1\ 1\}$ to be crossed. If they will cross from the fourth point, the result should be $C3 = \{1\ 1\ 0\ 1\ 0\ 1\ 1\}$ and $C4 = \{1\ 0\ 0\ 1\ 0\ 0\ 0\}$.

During the mutation process, a mutation bit is randomly selected and mutated with probabilities. For example, the fifth point of $C1$ is the mutation bit; it will change from 0 to 1. The result of mutation is $\{1\ 1\ 0\ 1\ 1\ 0\ 0\}$.

The mating process and the mutation process are also subjected to the two constraints mentioned above. That means the beginning and ending points must be one and two adjacent points could not be 1 at the same time.

6.2.2 Prediction Methodology

After the segmentation and clustering, we train a TDNN for each cluster to model its underlying pattern. SVR also shows its superiority of time series prediction in many researchers' work [135, 136]. The difference between TDNN and SVR is that the former uses empirical risk minimization principle while the latter uses structure risk minimization principle. SVR tries to minimize the classification error from the correct solution while TDNN seeks to minimize the upper bond of the error. Sometimes SVR can offer global optimum while the TDNN is trapped in local optimum. Thus, to take the advantage of SVR in our framework, a SVR model is trained and serves as additional prediction model in our framework. When none of the TDNNs is appropriate to model the time series, it will be used to conduct the prediction.

a) Time Delay Neural Network

J.G.D. Gooijer and R.J. Hyndman have made a thorough review of recent development on time series prediction[127]. Instead of using traditional classical statistical prediction method, data mining paradigm, such as TDNN has been proved to be good in capturing the underlying non-linear relationship of time series. As known in [128] using neural network performs better than most other algorithms. Thus, it is selected as the prediction model for further training. The detail of TDNN has been elaborated in previous section.

b) Support Vector Regression

Support vector machine (SVM) is proposed by Vapnik in 1995[137]. The fundamental concept of SVM is projecting the data to a higher dimensional space and finding an appropriate hyper plane to separate data into different classes. A simple 2D-plane can only separate the data set into two classes. The hyper-plane is formed by part of the training data.

Support vector regression is developed based on SVM theory [138]. It introduces a ε -insensitive parameter to the SVM model. In another word, the SVR tolerates training error to the degree decided by ε parameter. Training data lying within the ε -zone is considered as correct, otherwise considered as incorrect and deemed as supporting vector. The supporting vectors will further determine the regression surface.

The same as SVM, SVR also projects the original data space $\{(x_1, y_1), \dots, (x_l, y_l)\} \in X \times R$ (X denotes the space of input patterns) into space with higher dimension. As linear regression, SVR is to find a surface which deviates no more than ε from the training data in the hyper space and keep the surface as flat as possible at the same time.

The linear functions f can be defined as:

$$f(x) = \langle w, x \rangle + b \text{ with } w \in X, b \in R \quad 6.3$$

$\|w\|$ represents the flatness of the regression surface. Now the problem can be expressed as an optimization problem with subjection:

$$\begin{aligned} & \text{Minimize: } \frac{1}{2} \|w\|^2 \\ & \text{s.t. } \begin{cases} y_i - w^T x_i - b \leq \varepsilon \\ w^T x_i - b + y_i \leq \varepsilon \end{cases} \end{aligned} \quad 6.4$$

To increase flexibility to the optimization, two non-negative slack variables ξ_i, ξ_i^* are added. Now the optimization problem becomes:

$$\begin{aligned} & \text{Minimize: } \frac{1}{2} \|w\|^2 + C \sum_1^l (\xi_i + \xi_i^*) \\ & \text{s.t. } \begin{cases} y_i - w^T x_i - b \leq \varepsilon + \xi_i \\ w^T x_i - b + y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad 6.5$$

The parameter C determines the tradeoff of flatness of the plane and deviation of the target data. Increasing C will tolerate larger deviation and decrease the flatness, vice versa.

Lagrangina multipliers algorithm is used to optimize the cost function. SVR using a kernel function to project the original data into higher dimension space. The process can be expressed as:

$$\begin{aligned} & \Phi: R^n \rightarrow F \\ & X \rightarrow \Phi(X) \end{aligned} \quad 6.6$$

The decision function is computed by $\Phi(X)^T \Phi(X)$ thus kernel function is $K(x, z) = \Phi(X)^T \Phi(X)$.

c) Pattern Identification

To select the proper prediction model, also to combine both TDNN and SVR in our prediction framework, a procedure of pattern identification must be performed before

prediction. As shown below:

$$Y_t = \begin{cases} f(M_1, s) & \text{if it should be model 1} \\ \vdots & \\ f(M_n, s) & \text{if it should be model } n \end{cases} \quad 6.7$$

s is the actual time series data and M_n represents the prediction model we adopt. Y_t is the prediction at time t .

To find the appropriate prediction model for the current time, an error function is constructed:

$$J_t = \sum_{i=t-w}^t (y_i - \hat{y}_i)^2 \quad 6.8$$

y_i is the actual data at time i while $\hat{y}_i$ is the predicted data at that time. To find an appropriate prediction model at time t , all prediction models are fed with the previous data of the time series. Then Equation 6.8 is used to compute the error function J_t . The model with minimal J_t will be regarded as the most appropriate prediction model for the current time.

The length of w , which is called window size of the pattern identification, is predefined by the user.

6.3 Simulation

To assess the performance of the methodology discussed above, the data of solar radiation obtained in 2009 and 2010 in Singapore are used in the simulation. The data is recorded by the observation station located in Nanyang Technological University (NTU) and can be easily accessed online.

Just as mentioned in previous chapters, the index symmetric mean absolute percentage error (SMAPE) is capable in evaluating the performance of prediction algorithms. In this section, we also adopt SMAPE to evaluate the performance of GAMMF to facilitate the comparison of the proposed work with previous work. The definition of SMAPE is presented in equation 3.15.

Although the proposed GAMMF is theoretically capable in both long term and short term time series, in this chapter, we mainly focus on short term prediction. Our prediction horizontal is five minutes ahead. The data of 2009 is used as training data and data of 2010 is used as testing data.

6.3.1 Segment of Time Series

In this section, the segmentation experiment was conducted on the residual series. The program was implemented in matlab. As mentioned in previous section, genetic algorithm and K-means are collaboratively used in the segmentation phase and the number of cluster for K-means must be pre-defined. To find the optimal number K , we perform segmentation experiment on different K (from 2 to 12) and compute the corresponding silhouette index. The experiment result is presented in Table 6.1 and it shows the cluster of five presenting the best silhouette index value.

Table 6.1 Silhouette index value of different number of clusters

Number of clusters	Silhouette index
2	0.974
3	0.984
4	0.982
5	0.985
6	0.972
7	0.968
8	0.968
9	0.977
10	0.978
11	0.964
12	0.971

Figure 6.2 The silhouette index of different number of clusters

Thus the cluster of five is adopted in the segmentation process.

To compare the similarity of segment of different length, polynomial interpolation is used to normalize the subsequences to facilitate the clustering.

(a)

(b)

(c)

Figure 6.3 The interpolation of segment of different length

Figure 6.4 illustrates how we compare the similarity between subsequences of different length. The series in Figure 6.4 (a) is an example of long subsequence and the Figure 6.4 (b) one is a short subsequence. It is hard to compare them directly. Here we use polynomial interpolation to normalize both subsequences to the length of 15. The result is as shown in Figure 6.4 (c). As the experiment result in chapter 5 shows that segment of 70 minutes is suitable for the solar radiation series of Singapore. Thus we convert all the segmented subsequence to the length of 70 minutes.

K-means algorithm is again chosen to group the segment into different clusters. The

measurement of similarity is of great importance for K-means. For the K-means tries to find a schema of clustering to maximize the average distance between different cluster and minimize the distance within the same cluster.

After grouping the subsequences into different clusters, the average silhouette index is computed to check the goodness of segmentation and clustering. And GA is used to search chromosome which brings the best segment result. Solar radiation time series of one month long is used in the experiment of segmentation and clustering. The parameter we use in GA algorithm is shown in Table 6.2.

Table 6.2 The parameters used in the segment experiment

parameter	value
Number of Generation	200
Population size	80
Number of clusters (used in K-means)	5
Length of Polynomial interpolation	70 minutes
Cross rate	0.7
Mutation rate	0.01

The experiment of GA is made to find the optimal solution of segmentation, as the opposite number of the silhouette index is used as the fitness function, our objective is to get the minimal value. The average and optimal fitness value of each generation is shown in Figure 6.4.

Figure 6.4 The average and optimal fitness value along with different numbers of generations

From Figure 6.4 we could clearly see that the fitness value goes downward along the generations and finally stop at a certain value. At the same time, the gap between mean fitness and best fitness decrease along the generations. That means the chrome converge to the optimal solution.

6.3.2 Training the Prediction Model

Upon finishing the segmentation phase, we will use the segmentation and clustering result to further develop the prediction models.

TDNN is used for training and further prediction. The reason is that it has great flexibility and ability to capture the nonlinear relationship lying in the series. As discussed in previous chapter, for TDNN structure, the sigmoid function was proved to be optimal in the prediction experiment. All the popular training methods were tested with various time delay and neuron sizes. It was experimentally determined that the fastest convergence with the smallest prediction error was obtained by using the LM (Levenberg-Marquard) method in the training phase 3-step time delay and five neurons in the hidden layer.

The data of each cluster is used to train a TDNN model separately to represent that

specific pattern. Five different prediction models on the pattern are obtained. Each model represents a specific pattern of the solar radiation. When predicting the solar radiation, which model is appropriate for current time is determined by the identification function, and then the proper model is used to conduct the prediction.

To better diversify the candidate prediction models and improve the prediction accuracy, a recently developed algorithm SVR is incorporated in the framework. A SVR model which is trained by the current day's data is used as an additional model. When none of the five TDNNs is appropriate for the prediction, the SVR model is used.

Before applying the trained model to predict the future trend of solar radiation, which model is the most appropriate should be first identified. That is to find out which model present the best prediction performance in the past w steps. Just as previous discussed in chapter 5, window size of six is proved to be suitable in pattern identification. To simplify the parameter of GAMMF and also to facilitate the comparison with the previous algorithm, window size of six is again adopted in the prediction experiment.

In the experiment, the solar radiation data of 2009 is used as the training data and the data of 2010 is used as the testing data. When predicting a daily solar radiation series of 2010. Genetic algorithm is applied to find optimal segment schema. Then the segmented subsequence is grouped into different clusters by K-Means. The data of different clusters is used to train a TDNN models to represent that specific pattern. SVR model is also used to better diversify the candidate prediction model set of the framework. A model identification phase is applied to find out which model is the most appropriate for current prediction task. Then the data is fed to the chosen model to conduct the prediction.

The flow chart of the framework is shown in Figure 6.5.

Figure 6.5 the flow chart of the experiment

6.3.3 Experimental Result

The solar radiation data of 2010&2009 is used to verify the performance of the proposed framework. Although the framework could conduct both long term and short term time series prediction theoretically, in this chapter only short term prediction is addressed. The framework is applied to predict solar radiation of 5 minutes ahead. SMAPE is used to measure the prediction performance

Figure 6.6 and Figure 6.7 show the comparison of the prediction series and the actual series. Figure 6.6 is prediction of daily solar radiation under clear weather condition. The prediction series is quite close to the actual one. Figure 6.7 is prediction of solar radiation under less ideal weather condition. The solar radiation fluctuates swiftly due to the fast changing weather. It is very hard to do prediction in such case. However, the GAMMF still captures the up and down spike of the solar series.

Figure 6.6 The prediction of GAMMF on July 28th

Figure 6.7 The prediction of GAMMF on Feb 2nd

Table 6.3 MAPE of prediction of different month

	mean of SMAPE	variance of SMAPE
Jan	17.67	24.81
Feb	22.39	60.06
Mar	19.60	51.74
Apr	13.26	18.32
May	9.62	10.40
June	16.82	54.96
July	19.70	58.18
Aug	18.36	73.05
Sep	19.15	70.81
Oct	20.47	55.41
Nov	12.50	20.09
Dec	18.44	19.83

The result of the experiment justified the use of GA in prediction. The best result is achieved in the prediction of solar radiation of May which is 9.62, and the worst is achieved in February which is 22.39. The same thing happened in the variance column, the prediction of May also presents the lowest variance 10.40 while the prediction of August presents the highest variance 73.05.

6.4 Comparative Analysis

6.4.1 Comparative Analysis of Different Algorithms

To further assess the prediction capability of GAMMF, the experiment of GAMMF is compared with other algorithms. As presented in [139], ARMA and TDNN has been proved to have good accuracy in predicting daily solar radiation time series. The hybrid model [130] is developed based on ARMA and TDNN and has better prediction accuracy. MMF model is discussed in chapter 5. It presents the best prediction accuracy of daily solar radiation among these algorithms so far. In this section, the performance of proposed GAMMF will be compared with all the other algorithms.

Table 6.4 Mean of SMAPE of different prediction models

	GAMMF	MMF	Hybrid model	TDNN	ARMA
Jan	17.6	21.5	28.0	30.0	41.4
Feb	22.3	24.7	36.9	33.6	45.2
Mar	19.6	33.4	52.4	55.4	63.4
Apr	13.2	21.2	33.3	37.7	45.4
May	9.62	17.7	24.6	24.6	38.3
June	16.8	22.9	25.8	37.3	47.2
July	19.7	19.1	34.4	34.1	46.1
Aug	18.3	17.1	31.5	33.6	38.3
Sep	19.1	20.7	34.0	37.1	40.4
Oct	20.4	15.4	28.3	25.6	29.1
Nov	12.5	16.6	27.4	37.2	41.0
Dec	18.4	22.5	27.2	29.3	39.5

Table 6.5 Variance of SMAPE of different prediction models

	GAMMF	MMF	Hybrid model	TDNN	ARMA
Jan	24.8	89.6	79.1	416	290
Feb	60.0	106	205	301	362
Mar	51.7	184	676	803	642
Apr	18.3	94.5	117	684	734
May	10.4	57.4	68.4	155	134
June	54.9	70.4	230	575	229
July	58.1	93.7	100	542	587
Aug	73.0	56.7	104	524	275
Sep	70.8	61.6	125	509	425
Oct	55.4	37.8	54.8	447	129
Nov	20.0	94.7	312	540	585
Dec	19.8	81.5	127	215	135

Table 6.6 RMSE of different prediction models

	GAMMF	MMF	Hybrid model	TDNN	ARMA
Jan	113	122	130	137	137
Feb	148	161	172	174	206
Mar	113	138	150	177	174
Apr	79.3	109	124	141	170
May	96.6	111	118	119	152
June	52.6	91.0	105	104	146
July	104	101	106	135	136
Aug	90.0	90.4	103	153	131
Sep	87.1	99.0	113	180	150
Oct	79.4	81.1	88.1	128	118
Nov	93.7	85.4	94.8	118	118
Dec	105	109	125	191	146

Table 6.4 shows the proposed GAMMF has better prediction performance than hybrid model, TDNN, ARMA in any time. And also outperforms MMF in most months. The mean value of SMAPE of GAMMF is lower than MMF in 9 out of 12 months. Lower mean value of SMPAE means better prediction accuracy. Only in July, August and October, the mean value of SAMPE of MMF is slightly lower than GAMMF. The best improvement is achieved in May, which is 45.71%.

The variance of SMAPE for different prediction model in Table 6.5 is similar to Table 6.4. Compared to the other algorithms, the prediction result of GAMFF has lower variance in most months and only slightly higher than MMF in August, September and October. That means the stability of the prediction performance of GAMMF is better

than other algorithms. The best improvement of variance is also obtained in the prediction of May which is as high as 81.91%.

Root mean square error (RMSE) is also used to evaluate the performance of different prediction models. The result is presented in Table 6.6. We could see that the proposed GAMMF outperforms other algorithms in most months. This also proves GAMMF is superior over other models.

6.4.2 Comparative Analysis of Prediction Performance of Different Daily Integral Solar Radiation

The prediction performance of different level of daily integral solar radiation is compared to explore the correlation of weather condition and prediction performance. Low level of daily integral solar radiation means bad weather condition such as cloudy and rainy day. On the contrary, high level of daily integral solar radiation means sufficient sunshine in a clear day. As shown in Table 6.7, the mean value of SMAPE goes down as the daily integral solar radiation increases for all different algorithms. The prediction accuracy of GAMMF makes huge progress in the range of 0-50KW/m² compare to other algorithms. It improves the accuracy as high as 54.79%. In most ranges, GAMMF also beats other algorithms. One of GAMMF's advantages is that it shows very stable and consistent prediction performance under different weather condition. And it maintains the SMAPE below 18.88 which is very accurate at the same time. The only exception happens in the range from 350 KW/m² to 400 KW/m². The MMF and ARMA present lower SMAPE than GAMMF. But GAMMF still successfully keep it at 18.4, which is close to its performance in other ranges of daily integral.

The comparison of SMAPE of different level of daily integral of solar radiation between GAMMF and the other algorithms are presented in the following figures for better illustration. These figures clearly illustrate that GAMMF present better performance than other algorithms in almost any weather condition.

Table 6.7 Mean value of SMAPE of different level of daily integral solar radiation

Daily integral solar radiation (KW/m ²)	GAMMF	MMF	Hybrid model	TDNN	ARMA
0-50	30.2	66.9	69.0	103	106
50-100	18.8	29.9	38.0	64.3	65.1
100-150	17.4	20.3	30.9	42.2	49.9
150-200	16.0	20.0	29.5	29.2	43.0
200-250	18.2	21.0	31.6	32.1	42.6
250-300	18.4	19.6	28.5	29.1	39.2
300-350	18.8	21.1	27.5	30.3	33.1
350-400	18.4	11.8	24.7	24.5	16.0

Figure 6.8 The comparison of prediction of GAMMF & ARMA model in different level of daily integral solar radiation

Figure 6.9 The comparison of prediction of GAMMF & TDNN model in different level of daily integral solar radiation

Figure 6.10 The comparison of prediction of GAMMF & hybrid model in different level of daily integral solar radiation

Figure 6.11 The comparison of prediction of GAMMF & MMF model in different level of daily integral solar radiation

6.5 Summary

In this chapter, GAMMF, a novel approach for time series prediction is introduced. It is assumed that there are several patterns which will repeatedly occur in the stochastic time series. Genetic algorithm is used to find the optimal segment of the solar radiation series. K-means is used to group the segmented subsequence into five different clusters. One TDNN model is trained for each cluster to represent that specific pattern. A SVR model is also used combined with these trained TDNN models in the prediction. During the prediction phase, the pattern that current solar radiation series belongs to is identified first. Then the actual data is fed to chosen TDNN model to conduct prediction. If no pattern is suitable for prediction of current time, the trained SVR model will be used as the prediction model.

To verify the performance of the proposed GAMMF, the actual solar radiation data of Singapore which is obtain from the observation of Nanyang Technological University is used to conduct the experiment. The prediction performance of GAMMF is compared with other justified models such as ARMA, TDNN, hybrid model and MMF. The result shows that GAMMF has great advantage in accuracy and stability.

Future work will focus on further improvement of the clustering accuracy and on finding a better model to extract the relationship of data with the certain pattern.

Chapter 7. Conclusion and Future Work

7.1 Conclusion

As stated in chapter 1, our aim and motivation is to provide accurate and consistency prediction of solar radiation time series. This thesis begins with the review of existing algorithms which can be used to model solar radiation. The purpose of these algorithms is to predict the future value of solar radiation time series, which is very crucial for many industrial applications. We have studied the development of both empirical approaches and time series analysis approaches. Based on them, we proposed several novel prediction algorithms. The performance of the proposed algorithms is assessed with actual solar radiation time series. Our research is summarized in the following paragraphs.

In chapter 2, we carried out a review of the development of solar radiation prediction. It includes empirical approaches and time series analysis approaches. The empirical approaches usually try to model the relationship between solar radiation and other meteorological parameters. The time series analysis approaches try extracting the relationship between the past and future value of the solar radiation. Work which has been done on this topic is discussed.

In chapter 3, we used both traditional ARIMA model and TDNN model in the prediction of solar radiation time series data. Their performances are compared. We found both methods capable of modeling the solar radiation series. TDNN is more sensitive to the training data thus can capture the fast changing trend. But it is very sensitive to noise and frequently made unacceptable errors. ARIMA is a more consistent, but its prediction accuracy is not as good as TDNN.

As different approaches have their own advantages and disadvantages, we adopted a combination schema in chapter 4. We observed that there are both linear and nonlinear component in the solar radiation series. ARMA is used to model the linear component and TDNN is used to predict the nonlinear component. The model which combined them is called hybrid model. Before using ARMA, several empirical models of solar radiation are used in the detrending phase to generate stationary series. In the experiment, our hybrid model outperforms individual model.

In chapter 5, a novel MMF framework was proposed to predict the future value of solar radiation time series. We assumed that there are several different underlying patterns in the solar radiation time series. To extract these patterns, the solar radiation series is first segmented into subsequences with fixed length. Then K-means algorithm is used to group the subsequences into different cluster. For each cluster, a TDNN model is trained to represent that specific pattern. When predicting the future value of the solar radiation time series, the proper pattern it belongs to is identified. Then the appropriate TDNN model is used to conduct the prediction. In the experiment, the MMF's performance is compared with ARMA, TDNN and hybrid model. It shows that MMF outperforms other algorithms in many aspects.

In chapter 6, we proposed the GAMMF framework which is developed based on MMF. Although the segmentation with fixed length which we adopted in chapter 5 is easy to implement, it is a simplification of the reality. In this chapter a variable segmentation schema is used. Genetic algorithm is used to find the optimal segmentation schema. After segmentation and clustering, one unique TDNN model is trained for each cluster. Each trained TDNN model represents a specific pattern. When conducting prediction, which model is appropriate for the current time is first determined by the model identification algorithm. To further improve the prediction accuracy of our framework, a SVR model is also trained and serves as an additional model. When none of these TDNN models is appropriate for current prediction, the additional SVR model will be used.

All in all, we have developed methodologies which outperform traditional statistical approach in both accuracy and consistency. Still, there are things left for future exploration.

7.2 Future Work

The research described in this thesis has explored the performance of using data mining approach in time series prediction. The experiment result shows that data mining approach is superior over traditional approaches in many aspects. As we have mentioned in respective chapters, there are some potential future directions from this research.

Many time series prediction approaches require the input to be stationary while most real time series is non-stationary. Thus preprocessing is needed to transform it into stationary. Traditional preprocess methods are differencing and detrending. However, in the experiment we found it is far from adequate. Some time series present different statistical property in different time period. Differencing and detrending are not capable to generate a stationary series for such kind of data. There are also a lot of missing data and recording error found in such data set. Identifying them from the useful data is another challenging work. Some researchers converted the time series data to the frequency domain to detect the abnormal data. But there is still a lot of work to do in this field.

Further improving the segmentation algorithm is yet another challenging task. The optimal segmentation is splitting the time series when its statistical property changes so that the subsequence will be of identical statistical property. However detecting the changing point is very difficult. There is already quite a lot of research on this topic but no one offers a satisfactory solution for it.

Clustering of time series is also an important topic. Most of the clustering algorithm is based on the assumption that the number of cluster is known or pre-defined. However, this is not true in many cases. Especially for time series data set with little prior knowledge, it is very hard to define the number of clusters beforehand. There has been quite a number of research work aim to address this problem and find the appropriate number of cluster for a data set automatically. But none of them can provide a universal solution for any data set. Thus there is still a lot of work to do in this topic.

Finding proper model to extract the relationship or pattern of certain data set is yet another important topic. Although TDNN have been proved to be good in modeling the nonlinear relationship lying in different time series, it still has some disadvantages. When there is noise data lying in the training data set, TDNN may yield huge error. Also, TDNN sometimes suffers from the over-fitting problem. Actual time series data set often contains series of varied length. However, little research is done on prediction model which can take input with varied length.

Although we have proposed several novel approaches to predict the future value of time series, only short term time series prediction is addressed in our experiment.

Extending our work to long term prediction would be the future direction. Also, we would like to extend the experiment to other kinds of time series data such as financial data and energy price data besides solar radiation series.

Author's Publications

Journal Publications

J. Wu and C. K. Chan, "Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN," *Solar Energy*, vol. 85, pp. 808-817, May 2011.

J. Wu and C. K. Chan, "Prediction of solar radiation with Multi Model Framework," *Energy Conversion and Management*, vol. 76, pp. 347-355, Dec 2013.

J. Wu, C. K. Chan, Y. Zhang, B.Y. Xiong, Q.H. Zhang, "Prediction of solar radiation with genetic approach combining Multi-Model Framework", *Renewable Energy*, Conditional Accepted.

Conference Publications

J. Wu, C. Chan, J. Loh, F. Choo, and L. H. Chen, "Solar radiation prediction using statistical approaches," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, 2009, pp. 1-5.

J. Wu and C. K. Chan, "Detrending of Non-stationary Solar Radiation Time Series for Improved Prediction Performance" in *Infocomm Technologies in Competitive Strategies 2010. ICT2010*.

J. Wu and C. K. Chan, "The Prediction of Monthly Average Solar Radiation with TDNN and ARIMA," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, 2012, pp. 469-474.

Bibliography

- [1] P. J.A., "Evaporation from water surface in relation to solar radiation," *Trans Roy Soc Austr*, vol. 46, pp. 114-118, 1940.
- [2] B. W. A. Duffie J.A., "Solar engineering of thermal process," *New York: Wiley*, 1991.
- [3] A. Louche, G. Notton, P. Poggi, and G. Simonnot, "Correlations for Direct Normal and Global Horizontal Irradiation on a French Mediterranean Site," *Solar Energy*, vol. 46, pp. 261-266, 1991.
- [4] R. Dogniaux and M. Lemoine, "Classification of radiation sites in terms of different indices of atmospheric transparency," *Palz W.(áliteur), Solar Energy R&D in the European Community, Series F*, vol. 2, pp. 94-107, 1983.
- [5] K. Gopinathan, "A simple method for predicting global solar radiation on a horizontal surface," *Solar & wind technology*, vol. 5, pp. 581-583, 1988.
- [6] K. Zabara, "Estimation of the global solar radiation in Greece," *Solar & wind technology*, vol. 3, pp. 267-272, 1986.
- [7] K. Gopinathan, "A general formula for computing the coefficients of the correlation connecting global solar radiation to sunshine duration," *Solar Energy*, vol. 41, pp. 499-502, 1988.
- [8] P. Jain, "Comparison of techniques for the estimation of daily global irradiation and a new technique for the estimation of hourly global irradiation," *Solar & wind technology*, vol. 1, pp. 123-134, 1984.
- [9] A. P. Baig A., Mufti A., "A novel approach to estimate the clear day global radiation," *Renew Energy* vol. 1, pp. 119-23, 1991.
- [10] S. N. Kaplanis, "New methodologies to estimate the hourly global solar radiation; Comparisons with existing models," *Renewable Energy*, vol. 31, pp. 781-790, May 2006.
- [11] G. U. Yule, "On the method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers," *Philosophical Transactions of the Royal Society London, Series A*, vol. 226, pp. 267-298, 1927.
- [12] G. E. P. Box, G. M. Jenkins, and Macgrego.Jf, "Some Recent Advances in Forecasting and Control .2.," *Journal of the Royal Statistical Society Series C-Applied Statistics*, vol. 23, pp. 158-179, 1974.
- [13] A. Zellner, "An introduction to Bayesian inference in econometrics," *New Yor: Wiley*, 1971.
- [14] E. Parzen, "ARARMA models for time series analysis and forecasting " *Journal of Forecasting*, vol. 1, pp. 67-82, 1982.
- [15] J. G. De Gooijer, Klein, A., "On the cumulated multi-step-ahead predictions of vector autoregressvie moving average processes," *International Journal of Forecasting*, vol. 7, pp. 501-513, 1991.
- [16] M. J. Artis, Zhang, W., "BVAR forecasts for the G-7.," *International Journal of Forecasting*, vol. 6, pp. 349-362, 1990.
- [17] Z. Wang, Bessler, D. A., "Forecasting performance of multivariate time series models with a full and reduced randk: An empirical examination.," *International Journal of Forecasting*, vol. 5, pp. 241-247, 2004.
- [18] S. Hokoi, M. Matsumoto, and M. Kagawa, "Stochastic models of solar radiation and outdoor temperature," *ASHRAE Transactions (American Society of Heating, Refrigerating and Air-Conditioning Engineers);(United States)*, vol. 96, 1990.

- [19] B. H. Chowdhury and S. Rahman, "Forecasting sub-hourly solar irradiance for prediction of photovoltaic output," in *19th IEEE Photovoltaic Specialists Conference*, 1987, pp. 171-176.
- [20] L. Mora-Lopez and M. Sidrach-de-Cardona, "Multiplicative ARMA models to generate hourly series of global irradiation," *Solar Energy*, vol. 63, pp. 283-291, 1998.
- [21] B. Widrow, Rumelhart, D.E., Lehr, M.A., "Neural networks: Applications in industry, business and science," *Communications of the ACM*, vol. 37, pp. 93-105, 1994.
- [22] R. Sharda, "Neural networks for the MS/OR analyst: An application bibliography," *Interfaces*, vol. 24(2), pp. 116-130, 1994.
- [23] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.
- [24] A. Lapedes, Farber, R., "Nonlinear signal processing using neural networks: prediction and system modeling.," *Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM*, 1987.
- [25] A. J. Jones, "Genetic algorithms and their applications to the design of neural networks," *Neural Computing and Applications*, vol. 1, pp. 32-45, 1993.
- [26] H. Bacha, Meyer, W., "A neural network architecture for load forecasting.," in *Proceedings of the IEEE international Joint Conference on Neural Networks*, 1992, pp. 442-447.
- [27] D. Srinivasan, Liew, A.C., Chang, C.S., "A neural network short-term load forecaster," *Electric Power Systems Research*, vol. 28, pp. 227-234, 1994.
- [28] S. Makridakis, Anderson, A., Carbone, R., Fildes, R., Hibdon, M., Lewandowski, R., "The accuracy of extrapolation methods: Results of a forecasting competition," *Journal of Forecasting*, vol. 1, pp. 111-153, 1982.
- [29] A. S. Weigend, Gershenfeld, N.A., "Time Series Prediction: Forecasting the Future and Understanding the Past," *Addison-Wesley, Reading, MA*, 1993.
- [30] N. Kohzadi, Boyd, M.S., Kermanshahi, B., Kaastra, I., "A comparison of artificial neural network and time series models for forecasting commodity prices," *Neurocomputing*, vol. 10, pp. 169-181, 1996.
- [31] M. Mohandes, S. Rehman, and T. Halawani, "Estimation of global solar radiation using artificial neural networks," *Renewable Energy*, vol. 14, pp. 179-184, 1998.
- [32] A. Sfetsos and A. Coonick, "Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques," *Solar Energy*, vol. 68, pp. 169-178, 2000.
- [33] J. Soares, A. P. Oliveira, M. Z. Božnar, P. Mlakar, J. F. Escobedo, and A. J. Machado, "Modeling hourly diffuse solar-radiation in the city of São Paulo using a neural-network technique," *Applied Energy*, vol. 79, pp. 201-214, 2004.
- [34] J. Cao and S. Cao, "Study of forecasting solar irradiance using neural networks with preprocessing sample data by wavelet analysis," *Energy*, vol. 31, pp. 3435-3445, 2006.
- [35] A. Mellit, M. Benghanem, and S. A. Kalogirou, "An adaptive wavelet-network model for forecasting daily total solar-radiation," *Applied Energy*, vol. 83, pp. 705-722, 2006.
- [36] J. Cao and X. Lin, "Study of hourly and daily solar irradiation forecast using diagonal recurrent wavelet neural networks," *Energy Conversion and Management*, vol. 49, pp. 1396-1406, 2008.
- [37] S. Al-Alawi and H. Al-Hinai, "An ANN-based approach for predicting global radiation in locations with no direct measurement instrumentation," *Renewable*

- Energy*, vol. 14, pp. 199-204, 1998.
- [38] M. Mohandes, A. Balghonaim, M. Kassas, S. Rehman, and T. Halawani, "Use of radial basis functions for estimating monthly mean daily solar radiation," *Solar Energy*, vol. 68, pp. 161-168, 2000.
 - [39] M. Benghanem and A. Mellit, "Radial basis function network-based prediction of global solar radiation data: application for sizing of a stand-alone photovoltaic system at Al-Madinah, Saudi Arabia," *Energy*, vol. 35, pp. 3751-3762, 2010.
 - [40] K. Reddy and M. Ranjan, "Solar resource estimation using artificial neural networks and comparison with other correlation models," *Energy Conversion and Management*, vol. 44, pp. 2519-2530, 2003.
 - [41] A. Sözen, E. Arcaklioğlu, M. Özalp, and N. Çağlar, "Forecasting based on neural network approach of solar potential in Turkey," *Renewable Energy*, vol. 30, pp. 1075-1090, 2005.
 - [42] A. Mellit, M. Benghanem, A. H. Arab, and A. Guessoum, "A simplified model for generating sequences of global solar radiation data for isolated sites: Using artificial neural network and a library of Markov transition matrices approach," *Solar Energy*, vol. 79, pp. 469-482, 2005.
 - [43] L. Mart ń, L. F. Zarzalejo, J. Polo, A. Navarro, R. Marchante, and M. Cony, "Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning," *Solar Energy*, vol. 84, pp. 1772-1781, 2010.
 - [44] S. Rehman and M. Mohandes, "Artificial neural network estimation of global solar radiation using air temperature and relative humidity," *Energy Policy*, vol. 36, pp. 571-576, 2008.
 - [45] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Network*, vol. 10, pp. 988-999, 1999.
 - [46] N. Cristianini, Shawe-Talor, J., "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods," *Cambridge University Press*, 2000.
 - [47] A. J. Smola, Scholkopf, B., "A tutorial on support vector regression," *Royal Holloway College, London, UK, NeuroCOLT Tech. Rep.*, 1998.
 - [48] J. W. Zeng and W. Qiao, "Short-Term Solar Power Prediction Using an RBF Neural Network," *2011 Ieee Power and Energy Society General Meeting*, 2011.
 - [49] J. L. Chen, H. B. Liu, W. Wu, and D. T. Xie, "Estimation of monthly solar radiation from measured temperatures using support vector machines - A case study," *Renewable Energy*, vol. 36, pp. 413-420, Jan 2011.
 - [50] W. Z. Lu, W. J. Wang, A. Y. T. Leung, S. M. Lo, R. K. K. Yuen, Z. B. Xu, and H. Y. Fan, "Air pollutant parameter forecasting using support vector machines," *Proceeding of the 2002 International Joint Conference on Neural Networks, Vols 1-3*, pp. 630-635, 2002.
 - [51] W. J. Wang, Z. B. Xu, and J. W. Lu, "Three improved neural network models for air quality forecasting," *Engineering Computations*, vol. 20, pp. 192-210, 2003.
 - [52] B. E. Vieux and P. B. Bedient, "Estimation of rainfall for flood prediction from WSR-88D reflectivity: A case study, 17-18 October 1994," *Weather and Forecasting*, vol. 13, pp. 407-415, Jun 1998.
 - [53] H. Prem and N. S. Raghavan, "A support vector machine based approach for forecasting of network weather services," *Journal of Grid Computing*, vol. 4, pp. 89-114, 2006.
 - [54] T. B. Trafalis, Ince, H., "Support vector machine for regression and applications to financial forecasting," presented at the IEEE-INNS-ENNS Int. Joint Conference on Neural Netowrks 2000, 2000.

- [55] W. Ji and K. C. Chee, "Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN," *Solar Energy*, vol. 85, pp. 808-817.
- [56] F. E. H. Tay, Cao, L.J., "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, pp. 309-317, 2001.
- [57] F. E. H. Tay, Cao, L.J., "Improved financial time series forecasting by combining support vector machines with self-organizing feature map," *Intell. Data Anal.*, vol. 5, pp. 339-354, 2001.
- [58] F. E. H. Tay, Cao, L.J., "Modified support vector machines in financial time series forecasting," *Neurocomputing*, vol. 48, pp. 847-861, 2002.
- [59] V. T. Gestel, Suykens, J.A.K., Baestaens, D.E., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B., "Financial time series prediction using least squares support vector machines within the evidence framework," *IEEE Trans. Neural Network*, vol. 12, pp. 809-821, 2001.
- [60] A. Abraham and H. B. Liu, "Swarm intelligence based rough set reduction scheme for Support Vector Machines," *Isi 2008: 2008 Ieee International Conference on Intelligence and Security Informatics*, pp. 200-202, 2008.
- [61] P. Ongsritrakul and N. Soonthornphisaj, "Apply decision tree and support vector regression to predict the gold price," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, 2003, pp. 2488-2492.
- [62] Y. K. Bao, Liu, Z.T. Wang, W. , "Forecasting stock composite index by fuzzy support vector machines regression," presented at the Conference on Machine Learning and Cybernetics, 2005.
- [63] D. Sansom. D.C., T., Saha, T.K., "Evaluation of support vector machine based forecasting tool in electricity price forecasting for australiannational electricity market participants," presented at the Australian Universities Power Engineering Conference, 2002.
- [64] H. C. Z. Huang, C.J. Hsu, W.H. Chen, S. Wu, "Credit rating analysis with support vector machines and neural networks: A market comparative study," *Decision Support System*, vol. 37, pp. 543-558, 2004.
- [65] R. P. I. Bose, "Using support vector machines to evaluate financial rate of dotcoms," presented at the Pacific Asia Conference on Information Systems, 2005.
- [66] S. L. Y. Hur, "Customer churning prediction using support vector machines in online auto insurance service," presented at the International Symp. on Neural Networks, Lecture Notes in Computer Science, 2005.
- [67] C. S. L. P.F. Pai, "Using support vector machines to forecast the production values of the machinery industry in TaiWan," *Int. J. Adv. Manuf. Technol.*, , vol. 27, pp. 205-210, 2005.
- [68] B. J. C. M.W. Chang, C.J. Lin, "EUNITE network competition:Electricity load forecasting," *National Taiwan University Press*, 2001.
- [69] M. Mohandes, "Support vector machines for short-term load forecasting," *Int. J. Energy Resource*, vol. 26, pp. 335-345, 2002.
- [70] A. N. L. Tian, "A novel approach for short-term load forecasting using support vector machines," *Int. J. Neural System*, vol. 14, pp. 329-335, 2004.
- [71] Y. Bao, Y. Lu, and J. Zhang, "Forecasting stock price by SVMs regression," *Artificial Intelligence: Methodology, Systems, and Applications*, pp. 295-303, 2004.
- [72] N. R. Y.Ji. J. Hao, A. Lendasse, "Direct and recursive prediction of time series using mutual information selection," presented at the IWAAN, 2005.
- [73] X. Li, C. Sun, and D. Gong, "Application of support vector machine and similar

- day method for load forecasting," *Advances in Natural Computation*, pp. 428-428, 2005.
- [74] Y. J. He, Y. C. Zhu, and D. X. Duan, "Research on hybrid ARIMA and support vector machine model in short term load forecasting," *ISDA 2006: Sixth International Conference on Intelligent Systems Design and Applications, Vol 1*, pp. 804-808, 2006.
 - [75] T. Warren Liao, "Clustering of time series data—a survey," *Pattern Recognition*, vol. 38, pp. 1857-1874, 2005.
 - [76] M. K. J. Han, "Data Mining: concepts and Techniques,," *Morgan Kaufmann, San Francisco*, pp. 346-389, 2001.
 - [77] A. J. R. Krishnapuram, O. Nasraoui, L. Yi, , "Low-complexity fuzzy relational clustering algorithms for web mining," *IEEE Trans. Fuzzy System*, vol. 9, pp. 595-607, 2001.
 - [78] H. P. K. M. Ester, J. Sander, X. Xu, , "A density-based algorithm for discovering clusters in large spatial databases," presented at the International Conference on Knowledge Discovery and Data Mining, Portland, 1996.
 - [79] M. B. M. Ankerst, H.P. kriegel, J. Sander, "OPTICS: ordering points to identify the clustering structure," presented at the ACM-SIGMOD International Conference on Management of Data, Philadelphia, 1999.
 - [80] J. Y. W. Wang, R. Muntz,, "STING: a statistical information grid approach to spatial data mining," presented at the International Conference on Very Large Data Base (VLDB'97), Athens, Greek, 1997.
 - [81] J. S. P. Cheeseman, Ed., *Bayesian classification (AutoClass): theory and results, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.) (Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, Cambridge, 1996, p.^pp. Pages.*
 - [82] K. Košmelj and V. Batagelj, "Cross-sectional approach for clustering time varying data," *Journal of Classification*, vol. 7, pp. 99-109, 1990.
 - [83] B. B. T.W. Liao, J. Forester, E. Hailman, C. Hansen, R.C. Kaste, J. O'May, "Understanding and projecting the battle state," presented at the 23rd Army Science Conference, Orlando,FL., 2002.
 - [84] S. K. X. Golay, G. Stoll, D. Meier, A. Valavanis, P. Boesiger,, "A new correlation-based fuzzy logic clustering algorithm for fMRI," *Mag. Resonance Med.*, vol. 40, pp. 249-260, 1998.
 - [85] J. J. Van Wijk and E. R. Van Selow, "Cluster and calendar based visualization of time series data," in *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, 1999, pp. 4-9, 140.
 - [86] V. Kumar, S. Mohanty, A. Kumar, R. P. Misra, M. Santosham, S. Awasthi, A. H. Baqui, P. Singh, V. Singh, and R. C. Ahuja, "Effect of community-based behaviour change management on neonatal mortality in Shivgarh, Uttar Pradesh, India: a cluster-randomised controlled trial," *The Lancet*, vol. 372, pp. 1151-1162, 2008.
 - [87] F. K. C.S. Moller-Levet, K.H. Cho, O. Wolkenhauer,, "Fuzzy clustering of short time series and unevenly distributed sampling points," in *International Symposium on Intelligent Data Analysis*, Berlin, Germany, 2003.
 - [88] Y. Kakizawa, R. H. Shumway, and M. Taniguchi, "Discrimination and clustering for multivariate time series," *Journal of the American Statistical Association*, vol. 93, pp. 328-340, 1998.
 - [89] R.H. Shumway, "Time-frequency clstering and discriminant analysis," *Stat. Probab, Lett.*, vol. 63, pp. 307-314, 2003.

- [90] A. B. G. S. Policker, "Nonstationary time series analysis by temporal clustering," *IEEE Trans. Syst. Man Cybernet.-B: Cybernet.*, vol. 30, pp. 339-343, 2000.
- [91] T. Warren Liao, "A clustering procedure for exploratory mining of vector time series," *Pattern Recognition*, vol. 40, pp. 2550-2562, 2007.
- [92] L. R. R. J.G. Wilpon, "Modified k-means clustering algorithm for use in isolated word recognition," *IEEE Trans. Acoust. Speech Signal Process*, vol. 33, pp. 587-594, 1985.
- [93] G. P. K. C.T. Shaw, "Using cluster analysis to classify time series," *Physica D*, vol. 58, pp. 288-298, 1992.
- [94] P. T. C. Goutte, E. Rostrup, "On clustering fMRI time series," *Neuroimage*, vol. 9, pp. 298-310, 1999.
- [95] L. K. H. C. Goutte, M.G. Liptrot, E. Rostrup,, "Feature-space clustering for fMRI meta-analysis," *Hum. Brain Mapping*, vol. 13, pp. 165-183, 2001.
- [96] F. L. C. T.C. Fu, V.Ng, R. Luk, "Pattern discovery from stock time series using self-organizing maps," presented at the KDD Workshop on Thepmoral Data Mining, San Francisco, 2001.
- [97] L. M. D. Owsley, L. E. Atlas, and G. D. Bernard, "Self-organizing feature maps and hidden Markov models for machine-tool monitoring," *Ieee Transactions on Signal Processing*, vol. 45, pp. 2787-2798, Nov 1997.
- [98] J. L. M. Vlachos, E. Keogh, D. Gunopulos, "a wavelet-based anytime algorithm for k-means clustering of time series," presented at the 3rd SIAM International Conference on Data Mining, San Francisco, 1997.
- [99] D. Piccolo, "A distance measure for classifying ARIMA models," *Journal of Time Series Analysis*, vol. 11, pp. 153-164, 2008.
- [100] R. Baragona, "A simulation study on clustering time series with meta-heuristic method," *Quad. Stat.*, vol. 3, pp. 1-26, 2001.
- [101] J. Beran and G. Mazzola, "Visualizing the relationship between two time series by hierarchical smoothing models," *Journal of Computational and Graphical Statistics*, vol. 8, pp. 213-238, Jun 1999.
- [102] P. S. M. Ramoni, P. Cohen,, "Bayesian clustering by dynamics," *Mach. Learning*, vol. 47, pp. 91-121, 2002.
- [103] Y. M. Xiong and D. Y. Yeung, "Mixtures of ARMA models for model-based time series clustering," *2002 Ieee International Conference on Data Mining, Proceedings*, pp. 717-720, 2002.
- [104] D. Tran and M. Wagner, "Fuzzy entropy clustering," in *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, 2000, pp. 152-157.
- [105] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 719-725, Jul 2000.
- [106] B. Knab, A. Schliep, B. Steckemetz, and B. Wichern, "Model-based clustering with hidden Markov models and its application to financial time-series data," *Between Data Science and Applied Data Analysis*, pp. 561-569, 2003.
- [107] C. Li and G. Biswas, "Temporal pattern generation using hidden Markov model based unsupervised classification," *Advances in Intelligent Data Analysis, Proceedings*, vol. 1642, pp. 245-256, 1999.
- [108] C. Li and G. Biswas, "A Bayesian approach to temporal data clustering using hidden Markov models," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 543-550.
- [109] G. E. P. Box, "An Unexpected Route to Time-Series - a Citation Classic

- Commentary on Time-Series Analysis - Forecasting and Control by Box,G.E.P., and Jenkins,G.M.," *Current Contents/Social & Behavioral Sciences*, pp. 22-22, Oct 30 1989.
- [110] E. Mckenzie, "General Exponential Smoothing and the Equivalent Arma Process," *Journal of Forecasting*, vol. 3, pp. 333-344, 1984.
 - [111] P. C. Young, "Spatial Time-Series - Analysis - Forecasting - Control - Bennett,Rj," *Environment and Planning A*, vol. 13, pp. 513-514, 1981.
 - [112] H. Akaike, "Citation Classic - a New Look at the Statistical-Model Identification," *Current Contents/Engineering Technology & Applied Sciences*, pp. 22-22, 1981.
 - [113] D. A. Dickey, Fuller, W.A., "Likelihood ratio statistics for autoregressive time series with a unit root," *Econometrica*, vol. 49, pp. 1057-1072, 1981.
 - [114] P. Wintoft and L. R. Cander, "Short-term prediction of foF2 using time-delay neural network," *Physics and Chemistry of the Earth Part C-Solar-Terrestrial and Planetary Science*, vol. 24, pp. 343-347, 1999.
 - [115] S. Makridakis and M. Hibon, "The M3-Competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, pp. 451-476, Oct-Dec 2000.
 - [116] P. R. Bevington and D. K. Robinson, *Data reduction and error analysis for the physical sciences*, 3rd ed. Boston: McGraw-Hill, 2003.
 - [117] F. H. Alsadah, F. M. Ragab, and M. K. Arshad, "Hourly Solar-Radiation over Bahrain," *Energy*, vol. 15, pp. 395-402, May 1990.
 - [118] H. F. Zou, F. F. Yang, and G. P. Xia, "Time series forecasting model using a hybrid ARIMA and neural network," *Proceedings of the 2005 Conference of System Dynamics and Management Science*, Vol 2, pp. 934-939, 2005.
 - [119] S. Park, S. W. Kim, and W. W. Chu, "SBASS: Segment based approach for subsequence searches in sequence databases," *Computer Systems Science and Engineering*, vol. 22, pp. 37-46, Jan-Mar 2007.
 - [120] X. Hui, Yunfa, H., "Data mining based on segmented time warping distance in time series database," *Computer Research and Development*, vol. 42-48, 2005.
 - [121] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, pp. 645-678, May 2005.
 - [122] F. Martinez-Alvarez, A. Troncoso, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Energy Time Series Forecasting Based on Pattern Sequence Similarity," *Ieee Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1230-1243, Aug 2011.
 - [123] D. Berndt, Clifford, J., "Using dynamic tim warping to find patterns in time series," presented at the AAAI-94 Workshop on knowledge discovery in databases(KDD-94), Seattle, Washington, 1994.
 - [124] R. K. Blashfield, "Finding Groups in Data - an Introduction to Cluster-Analysis - Kaufman,L, Rousseeuw,Pj," *Journal of Classification*, vol. 8, pp. 277-279, 1991.
 - [125] H. Le Capitaine and C. Frelicot, "A Cluster-Validity Index Combining an Overlap Measure and a Separation Measure Based on Fuzzy-Aggregation Operators," *Ieee Transactions on Fuzzy Systems*, vol. 19, pp. 580-588, Jun 2011.
 - [126] J. Dunn, "Well Separated Clusters and Optimal Fuzzy Partitions," *Cybernetics*, vol. 4, pp. 95-104, 1974.
 - [127] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, pp. 443-473, 2006.
 - [128] D. Ruta, B. Gabrys, and C. Lemke, "A Generic Multilevel Architecture for Time Series Prediction," *Ieee Transactions on Knowledge and Data Engineering*, vol. 23, pp. 350-359, Mar 2011.

- [129] L. Hontoria, J. Aguilera, and P. Zufiria, "An application of the multilayer perceptron: Solar radiation maps in Spain," *Solar Energy*, vol. 79, pp. 523-530, 2005.
- [130] J. Wu and C. K. Chan, "Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN," *Solar Energy*, vol. 85, pp. 808-817, May 2011.
- [131] J. R. Chen, "Making subsequence time series clustering meaningful," *Fifth IEEE International Conference on Data Mining, Proceedings*, pp. 114-121, 2005.
- [132] V. S. Tseng, C. H. Chen, C. H. Chen, and T. P. Hong, "Segmentation of time series by the clustering and genetic algorithms," *ICDM 2006: Sixth IEEE International Conference on Data Mining, Workshops*, pp. 443-447, 2006.
- [133] E. Keogh and J. Lin, "Clustering of time-series subsequences is meaningless: implications for previous and future research," *Knowledge and Information Systems*, vol. 8, pp. 154-177, Aug 2005.
- [134] F. L. Chung, T. C. Fu, V. Ng, and R. W. P. Luk, "An evolutionary approach to pattern-based time series segmentation," *Ieee Transactions on Evolutionary Computation*, vol. 8, pp. 471-489, Oct 2004.
- [135] W. Huang, Y. Nakamori, and S. Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers & Operations Research*, vol. 32, pp. 2513-2522, Oct 2005.
- [136] K. J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, pp. 307-319, Sep 2003.
- [137] V. N. Vapnik, *The nature of statistical learning theory*. New York: Springer, 1995.
- [138] V. N. Vapnik, *The nature of statistical learning theory*, 2nd ed. New York: Springer, 2000.
- [139] C. K. C. J. Wu, "The Prediction of Monthly Average Solar Radiation with TDNN and ARIMA.," presented at the IEEE International Conference on Machine Learning and Applications, Boca Raton, FL, 2012.