AUTOMATION OF ENERGY DEMAND FORECASTING

by

Sanzad Siddique, B.S.

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Electrical and Computer Engineering

Milwaukee, Wisconsin

December 2013

ABSTRACT

AUTOMATION OF ENERGY DEMAND FORECASTING

Sanzad Siddique, B.S.

Marquette University, 2013

Automation of energy demand forecasting saves time and effort by searching automatically for an appropriate model in a candidate model space without manual intervention. This thesis introduces a search-based approach that improves the performance of the model searching process for econometrics models. Further improvements in the accuracy of the energy demand forecasting are achieved by integrating nonlinear transformations within the models. This thesis introduces machine learning techniques that are capable of modeling such nonlinearity. Algorithms for learning domain knowledge from time series data using the machine learning methods are also presented. The novel search based approach and the machine learning models are tested with synthetic data as well as with natural gas and electricity demand signals. Experimental results show that the model searching technique is capable of finding an appropriate forecasting model. Further experimental results demonstrate an improved forecasting accuracy achieved by using the novel machine learning techniques introduced in this thesis. This thesis presents an analysis of how the machine learning techniques learn domain knowledge. The learned domain knowledge is used to improve the forecast accuracy.

ACKNOWLEDGMENTS

Sanzad Siddique, B.S.

I would like to thank my wife for her support and to my mother for her love. Their commitment, in all of its forms, has helped and guided me through my studies over the years. I would like to thank my advisor, Dr. Richard Povinelli, for his guidance over the past two years. He has provided me great support in all aspects. I would like to thank the other committee members, Dr. Johnson and Dr. Corliss. I am grateful to Gasday and Dr. Brown for their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF COMMON NOTATION

Notation	Significance
Y_t	Time series value at time *t*
Y_{t-i}	Time series value at time *t-i*
φ_i	Autoregressive coefficient for Y_{t-i}
ε_t	Random error of the model at time *t*
θ_i	Moving average parameter for time *t-i*
c	Constant used in the model
η_i	Exogenous input parameter for time *t-i*
d_t	Exogenous input at time *t*
p	Autoregressive order
q	Moving average order
b	Exogenous input order

1 INTRODUCTION

Building forecasting models for energy demand is an active research field in engineering, statistics and econometrics. A literature review finds previous works that automate the process of determining the forecasting model attributes such as type, orders and parameters [1-8]. It is possible to build the forecasting model automatically with the help of existing econometrics techniques [1-3] and information minimization criteria techniques [5-8], but these techniques suffer from performance issues in terms of accuracy and speed, especially for the higher order models. Kavaklioglu [9], Azadeh [10, 11] and Oğcu [12] recently have applied machine learning approaches to learn forecasting model orders and parameters. The work of Valenzuela [4] is an example of integrating an automatic model discovery algorithm with domain knowledge. This thesis builds on the work of these authors. This thesis integrates machine learning and econometrics methods to create a novel approach to automatically learn forecasting model type, order and parameters.

1.1 Problem Statement

The forecasting model-building process manually searches the model space requiring significant time and effort. The model space consists of candidate models. A candidate model includes specifications of the model type, the model order and the model parameters. Having an automated process to construct the forecasting model by automatically searching the model space is a solution to this problem. The purpose of this thesis is to develop algorithms that contribute towards automating the process of building energy demand forecasting models. The algorithms introduced in this thesis automatically search the space of candidate models. The model space includes statistical models such as autoregressive and moving average (ARMA) models, autoregressive and moving average with exogenous inputs (ARMAX) models and machine learning models such as artificial neural network (ANN), regression tree (RT) and support vector regression (SVR) models. An overview of the statistical models and machine learning techniques are presented in chapter two. The search algorithm also determines the model attributes such as structure, order and parameter values.

1.2 Motivation for this work

The process of energy demand estimation involves model identification, parameter estimation and prediction using the identified model. Generally, the complete model identification process is complex and requires substantial manual effort. The aim of this thesis is reduce the manual effort in building a model for energy demand forecasting. The contribution of this thesis is a set of novel techniques that contribute towards automation of energy demand forecasting model learning. Automatic energy demand forecasting model learning requires a process that performs all of the necessary steps of building a statistical forecasting model. The statistical forecasting model building steps include identifying the model type, the model structure, and the parameter values.

Energy demand time series are nonlinear by nature [13]. Thus, recent work has applied nonlinear modeling techniques to energy demand forecasting [9-11]. Also, significant improvement in forecasting can be achieved by incorporating multiple forecasting techniques through hybridization [4, 14, 15] and by applying ensemble learning [14]. There are examples in the literature where nonlinear machine learning methods are combined with linear models using ensemble techniques [3, 15]. This thesis extends this work by building ensembles of linear and nonlinear models.

Accurate forecasting of energy demand requires domain knowledge. Domain knowledge may differ between the types of energy to be forecasted. When forecasting in new energy domains, it is likely that there insufficient domain knowledge to build an accurate forecasting model. Thus, this thesis proposes an algorithm that can extract domain knowledge from the energy demand signals. A nonlinear technique typically is needed for the representation of domain knowledge [4]. Machine learning techniques are incorporated into the overall process to facilitate learning of domain knowledge. The models proposed by this thesis incorporate domain knowledge learning mechanisms.

Seasonal decomposition of energy demand provides reasonable results [14, 16, 17]. This thesis proposes and examines techniques based on the seasonal decomposition and combination of the results from the models built on the decomposed datasets.

1.3 Scope of the Work

Energy demand is usually forecasted for long term, midterm and short term [18]. Long term forecasting helps make strategic decisions. Midterm forecasting is used for managing resources. Short term energy demand forecasting reduces excess energy generation, blackouts and negative economic impact [19, 20]. This thesis introduces techniques that can automate short term energy demand forecasting. However, the techniques introduced by this thesis may help midterm and long term forecasting as well.

Techniques introduced by this thesis are for forecasting the energy demand. It may be possible to apply the technology proposed in this thesis to other areas, but other applications are not examined here. The proposed algorithms are built and tested with energy demand data. Moreover, the methods are tested exclusively on natural gas and electricity demand.

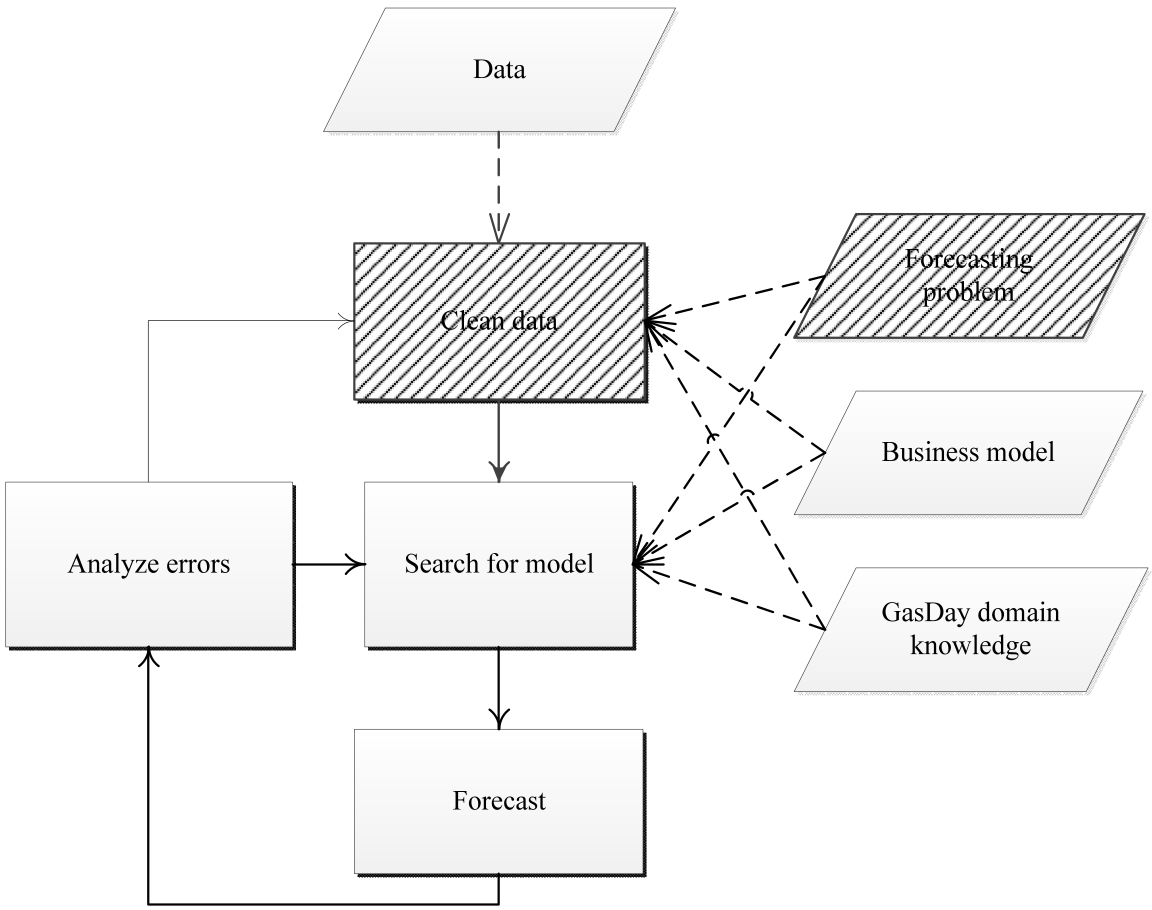


Figure 1.1 – Context of the work

An energy demand forecasting automation process is shown in Figure 1.1. The process starts with data cleaning. The candidate models from the model space are examined using the cleaned data. The forecasting problem, the business models and domain knowledge are used in the data cleaning and in the model searching processes. The selected models are used for forecasting the energy demand. The error is analyzed, and necessary changes are made in the data cleaning and the model searching processes. The data cleaning and the business model are not in the scope of this thesis. The thesis assumes that the data is clean and without anomalous data. This thesis focuses on the model search.

1.4 Summary of the Work

The presentation of this work starts by explaining simpler models and then extends the work towards more complex models. Time series that follow the ARMA model are presented first. The initial goal is to automate the process of finding a model among AR, MA or combined (ARMA) models with optimal orders and parameters. The automated process is done in several phases. The ARMA model identification involves two main steps: order identification and parameter estimation. The proposed research develops methods for finding the most appropriate model order. As there are different techniques available, the various techniques are tested and analyzed. Among these techniques, the Bayesian information criteria (BIC) technique determines the model orders most accurately. This thesis presents a search-based approach that overcomes the poor computational performance of a brute force BIC search. The search-based ARMA model discovery technique is extended to the discovery of the autoregressive and moving average with the exogenous inputs (ARMAX) model.

Besides linear components, energy demand time series data contains nonlinear and seasonal components [13]. Thus, it is desirable to improve forecasting accuracy by combining the linear models, such as ARMA and ARMAX, with nonlinear techniques and by considering the seasonal nature of the energy demand. Machine learning techniques such as ensemble learning [21], support vector regression [22, 23], artificial neural networks [10-12, 20] and regression trees [24, 25] are used widely for building nonlinear forecasting models. This thesis uses these techniques for modeling the nonlinearity of the underlying system and also to learn model dimensions and parameter values. Furthermore, improvement in forecasting accuracy can be achieved by incorporating multiple forecasting techniques such as hybridization [4, 14, 15] and ensemble learning [14]. This thesis suggests several ensemble learning methods that can forecast the energy demand accurately and automatically.

The first method, ENSEMBLE-REGRESSION, integrates the result from other statistical and machine learning techniques such as autoregressive and moving average with exogenous inputs (ARMAX), support vector regression (SVR), artificial neural networks (ANN) and regression trees (RT). Initially, energy demand is forecasted using those techniques. The results from each of those techniques are combined with the ensemble learning technique.

The second method, INPUT-MODELING, uses linear statistical modeling techniques and nonlinear machine learning techniques to model different sets of inputs. Those models are combined using the ensemble technique. The final forecasting result is obtained from the ensemble technique. The third method also performs input modeling, but uses residual instead of the actual demand data.

The third technique, MODELING-SEASONALITY, models the energy demand based on seasons. Some of the previous research works represent the decomposition of energy demand time series data based on the seasonal information [4, 14]. The decomposed data then is estimated using forecasting modeling techniques and then combined using the ensemble learning technique. Using these ideas, the energy demand time series data is partitioned into seasons. The separate seasonal datasets are modeled individually. Outputs from the individual models are combined using an ensembler. The final forecasted value is obtained from the ensemble output.

1.5 Summary of Contributions

The novel techniques proposed by this thesis are tested, and the test results are obtained. The test results show the accuracy and the performance improved by these techniques. The search-based algorithm is tested to estimate the orders of the econometrics models, including the autoregressive and moving average (ARMA), autoregressive with exogenous input (ARX) and the autoregressive and moving average with exogenous input (ARMAX) model. The three novel machine learning techniques are also tested, and the results are compared with the standard modeling techniques. The search-based approach show improved computational performance, and the machine learning techniques show improved accuracy.

1.6 Organization of the Thesis

This thesis contains five chapters. The first chapter provides an introduction to the thesis. The second chapter provides an overview on the forecasting techniques. The first section of the second chapter describes statistical and econometrics time series modeling techniques such as autoregressive (AR), moving average (MA), autoregressive with exogenous inputs (ARX), autoregressive and moving average (ARMA), autoregressive and moving average with exogenous inputs (ARMAX) and linear regression (LR). The second section in chapter two provides an overview of machine learning techniques such as artificial neural networks (ANN), regression trees (RT), support vector regression (SVR) and ensemble learning.

Chapter three introduces four novel techniques and algorithms. The first two sections of chapter three describe econometrics and statistical modeling techniques proposed by this thesis. The third section describes the new machine learning and hybrid techniques.

Chapter four describes the training and testing methods. The results are represented and analyzed in chapter four. Chapter five provides conclusions and suggestions for future work.

2 OVERVIEW OF FORECASTING TECHNIQUES

This chapter describes various forecasting techniques on which this thesis is based. These forecasting techniques include statistical and econometric models and machine learning techniques. The first section of this chapter describes the statistical and econometrics forecasting techniques. The second section describes various machine learning techniques used in this thesis. The third section presents the literature review of these techniques. The previous works in the relevant area are discussed in this section.

2.1 Statistical and Econometric Models

Many statistical and econometrics modeling techniques are available for forecasting time series data. These methods successfully forecast in many different areas such as financial [26], weather [27], energy [28] and household devices [29]. Hence, the methods are widely accepted. Forecasting modeling techniques include autoregressive (AR), autoregressive and moving average (ARMA), autoregressive moving average with exogenous input (ARMAX) and linear regression (LR). This section reviews these techniques.

2.1.1 Autoregressive Model

Autoregressive (AR) models are useful when the value to be forecasted is correlated to the previous values in the time series. The AR model is

$$Y_t = c + \varepsilon_t + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + ... + \varphi_p Y_{t-p}, \tag{2.1}$$

where Y_t indicates the time series value at time *t,* and Y_{t-i} indicates the value recorded at time *t-i*. The φ's represent the AR coefficients, where φ_i is the coefficient for Y_{t-i}.

Additional terms include a constant value *c* and a time-dependent normal random variable ε_t. Equation (2.1) is rewritten as

$$Y_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i Y_{t-i}, \tag{2.2}$$

where *p* represents the number of previous time series values to be incorporated into the model. This variable *p* is known as AR model order.

2.1.2 Moving Average Model

Moving average (MA) models are constructed by calculating the running average of the error generated at each point of time. Generally, the average values are weighted. The moving average model has the form

$$Y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t\text{-}1} + \theta_2 \varepsilon_{t\text{-}2} + \ldots.. + \theta_q \varepsilon_{t\text{-}q}, \tag{2.3}$$

where Y_t is the forecasted value at time *t*, which is a weighted average of the error at previous instances of time. The θ values are the coefficients of the moving average terms. Equation (2.3) is rewritten as

$$Y_t = c + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}, \tag{2.4}$$

where *q,* representing the number of previous error terms in the model, is known as the MA model order.

2.1.3 Autoregressive and Moving Average Model

An autoregressive and moving average (ARMA) model combines both autoregressive and moving average terms. It is one of the most commonly used order, η_i is the parameter of the exogenous input at time *i* and ε_t is a time-dependent random value that represents model error.

2.1.5 Autoregressive Moving Average with Exogenous Input Model

The autoregressive moving average with exogenous input (ARMAX) model is an extension of the ARMA model. It is similar to the ARX model with the additional moving average terms. The ARMAX model is

$$Y_t = c + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \sum_{i=0}^{b} \eta_i d_{t-i}, \tag{2.8}$$

where Y_t is the forecasted value at time *t*, *c* is a constant, *p* is the autoregressive orders, *q* is the moving average order, φ's are the autoregressive parameters, θ's are the moving average parameters, *d* is the exogenous inputs, *b* is the exogenous input order, η_i is the parameter of the exogenous input at time *i* and ε_t is random model error. Equation (2.8) is similar to Equation (2.7) for the ARX model, but with added moving average component, where the θ's are the moving average parameters, and *q* is the MA order.

2.1.6 Linear Regression Model

Linear regression (LR) models represent the relationship between a set of independent variables and a dependent variable. The dependent variable is correlated with each of the independent variables. The relationship is represented as

$$Y_t = c + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n, \tag{2.9}$$

where Y_t is the dependent variable and $x_1, x_2, \ldots, x_n$ are the independent variables. Each of these independent variables has a linear relationship with the dependent variable Y. The symbols $\alpha_1, \alpha_2, ..., \alpha_n$ represent the coefficients for respective independependent variables and are known as the parameters of the linear regression model. Variable c represents a constant offset. Equation (2.9) is rewritten as

$$Y_t = c + \boldsymbol{\alpha}\mathbf{X}_t, \tag{2.10}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_n]$ and $\mathbf{X}_t = [x_1, x_2, ..., x_n]^T$.

2.2 Machine Learning Techniques

Besides econometrics, statistical modeling and regression techniques, it is possible to incorporate machine learning techniques to forecast a time series. Machine learning techniques can be used for input selection and for learning the model dimension and parameters. It is possible to incorporate machine learning techniques with existing statistical and econometrics modeling techniques and to combine the results using an ensembler. Besides building the forecasting model by learning the model dimensions and parameters, it is possible to acquire domain knowledge for a particular area and apply it to different domain. Thus, machine learning techniques are capable of translating domain knowledge and are able to provide equivalent accuracy in forecasting without having complete domain knowledge compared to the accuracy obtainable by having domain knowledge.

Many machine learning techniques solve classification problems, but machine learning techniques also can be applied to regression problems. This section describes the machine learning techniques used in this thesis including ensemble learning, regression trees (RT), artificial neural networks (ANN) and support vector regressions (SVR).

2.2.1 Ensemble Learning

Ensemble learning combines results from other learners to provide a summary of results. Ensemble learners are used for classification [21] and regression [30, 31] problems. This thesis uses both the classification and regression ensemble techniques. The techniques are discussed in turn.

2.2.1.1 Majority Voting Ensemble

Majority voting treats each member (output from other machine learning techniques) equally and selects one output as a winner. The winner is the output chosen by the majority of the members. Thus, majority voting is the simplest ensembler and does not require model parameter learning. The result is obtained once the outputs from all members are available. This thesis uses the majority voting technique for selecting the forecasted value from more than one model output.

2.2.1.2 Ensemble Regression Algorithm

While majority voting selects a single output, ENSEMBLE-REGRESSION uses the outputs from all of the component models in determining the final output. ENSEMBLE-REGRESSION nonlinearly transforms the component model outputs and learns weights for each of the transformed outputs. If component model outputs were not transformed, ENSEMBLE-REGRESSION would be equivalent to linear regression, where the component model outputs are independent variables, and the weights are regression parameters. ENSEMBLE-REGRESSION combines the outputs from different modeling techniques.

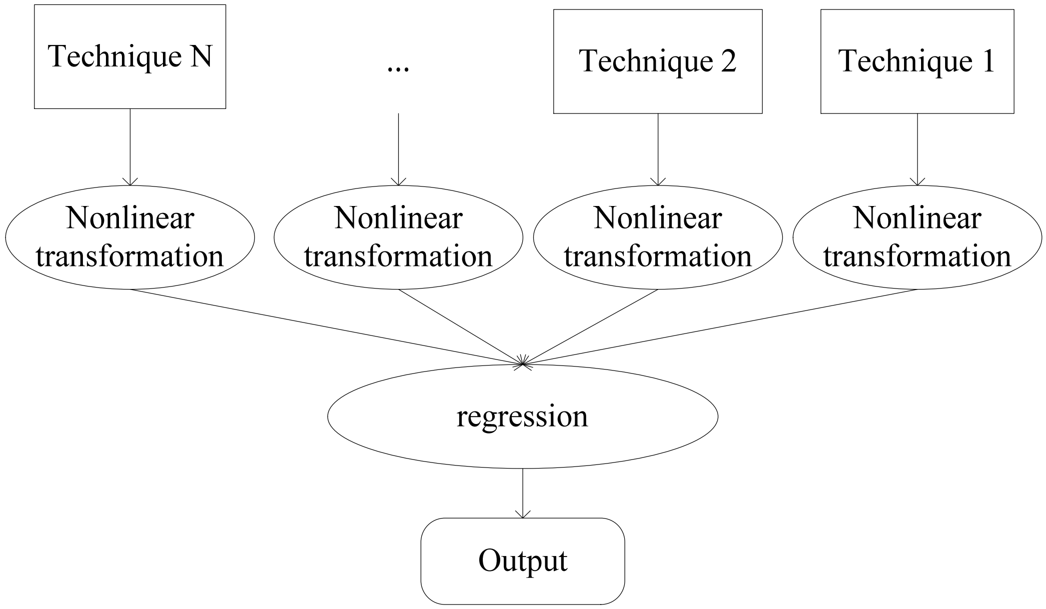


Figure 2.1 – ENSEMBLE-REGRESSION

Figure 2.1 represents ENSEMBLE-REGRESSION, where outputs from *N* forecasting techniques are combined using linear regression using a least square regression method [32]. The output from the individual component techniques can be transformed individually and nonlinearly before the regression.

2.2.2 Regression Tree

A regression tree is a special form of a binary decision tree used for building nonlinear regression models. A binary decision tree is a machine learning technique used for the classification, and a regression tree is used for regression. Like a binary decision tree, the decision nodes in a regression tree represent a decision based on the value of a given attribute. The leaves of the tree are learned using the forecasted values. There are fast and reliable algorithms available to learn the nodes and leaves [24]. Regression trees are used for forecasting [24, 25]. An advantage of using a regression tree is that it can forecast very quickly. Figure 2.2 shows an example of a regression tree. In this example, the inputs have four different attributes such as x_1, x_2, x_3 and x_4. Each internal node represents a condition on one these attributes. The leaves contain the output values (*i.e.*, values for Y). Once the complete regression tree is learned, an input is examined at the top node and is followed to the child nodes until a leaf is reached. The selected leaf represents the corresponding output for the given input. This thesis uses the regression tree to build a hybrid forecasting model, which is described in chapter three.

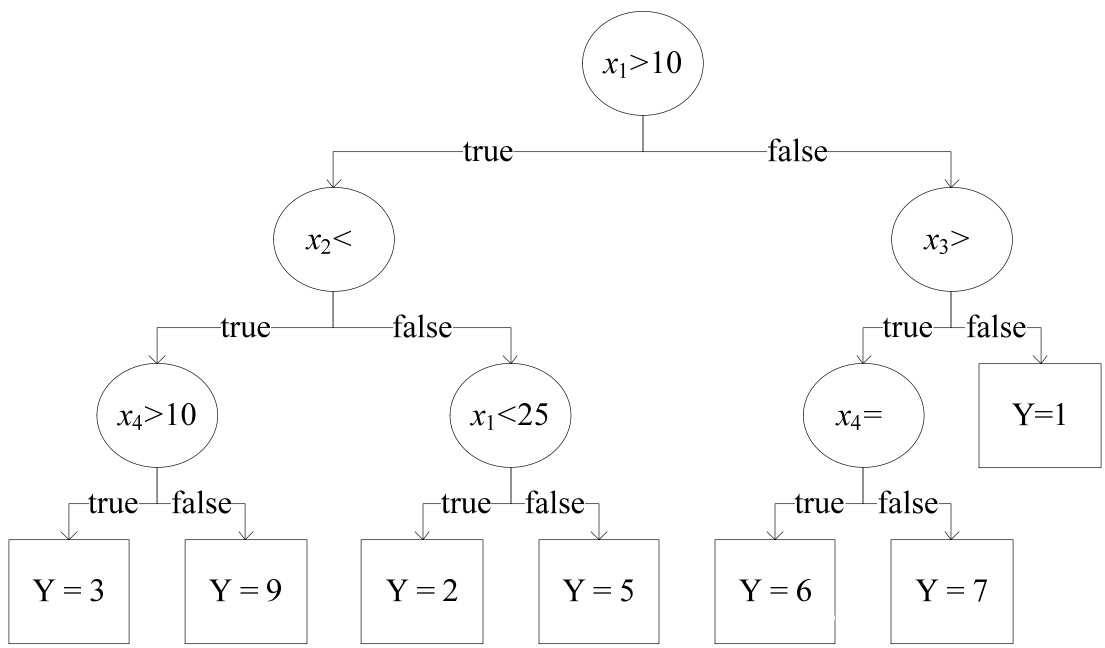


Figure 2.2 – Example regression tree

The regression tree can model a continuous variable, but as discrete values. It also does not extrapolate beyond the training data, as discussed in detail in section 4.2.3.

2.2.3 Artificial Neural Networks

Artificial neural networks (ANN) consist of fully or partially connected neurons. A neuron is a single processing unit in a neural network. Connections are made between the neurons and weights assigned for the connections. Each of those neurons has inputs, an activation function and outputs. A weight is associated with each of the inputs. The output is calculated by summing the weighted inputs and a bias value. The calculated output is processed by an activation function, and the final output is generated. The calculation taking place in a single neuron is

$$y = f\left(b_0 + \sum_{i=1}^{n-1} w_i x_i\right), \tag{2.11}$$

where x represents the input vector, y is the output, w is the weight vector, b_0 is the bias and $f(b_0, w, x)$ is the activation function. The activation function performs a transformation on the result. Most commonly, a sigmoid function is used as an activation function,

$$f(x) = \frac{1}{1+e^{-x}}. \tag{2.12}$$

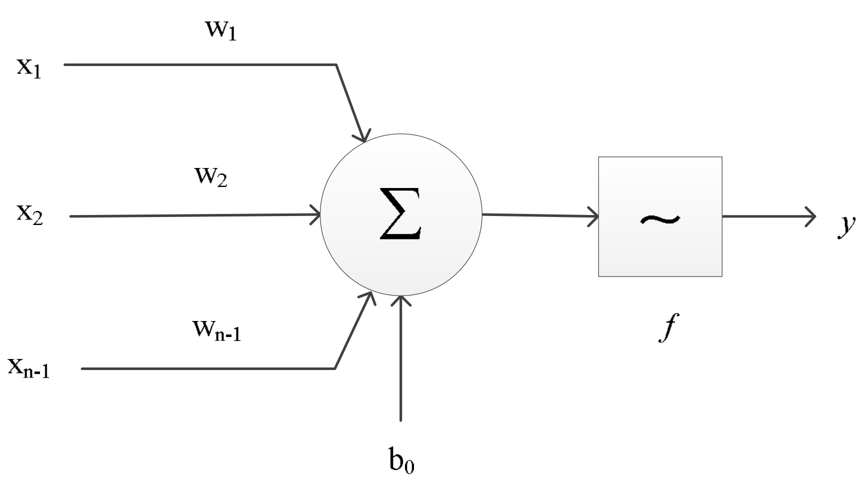


Figure 2.3 – Representation of a single node neuron

A single neuron of a neural network is shown in Figure 2.3. A single neuron works satisfactoraly only for linearly separable inputs. To support nonlinearity, a neural network with more than one neuron is needed. Neural networks can have multiple layers, where each of the layers consists of one or more neurons. The neurons from one layer are connected to the adjacent layer neurons. A multilayer neural network contains an input layer, an output layer and one or more hidden layers, as suggested by Figure 2.4.

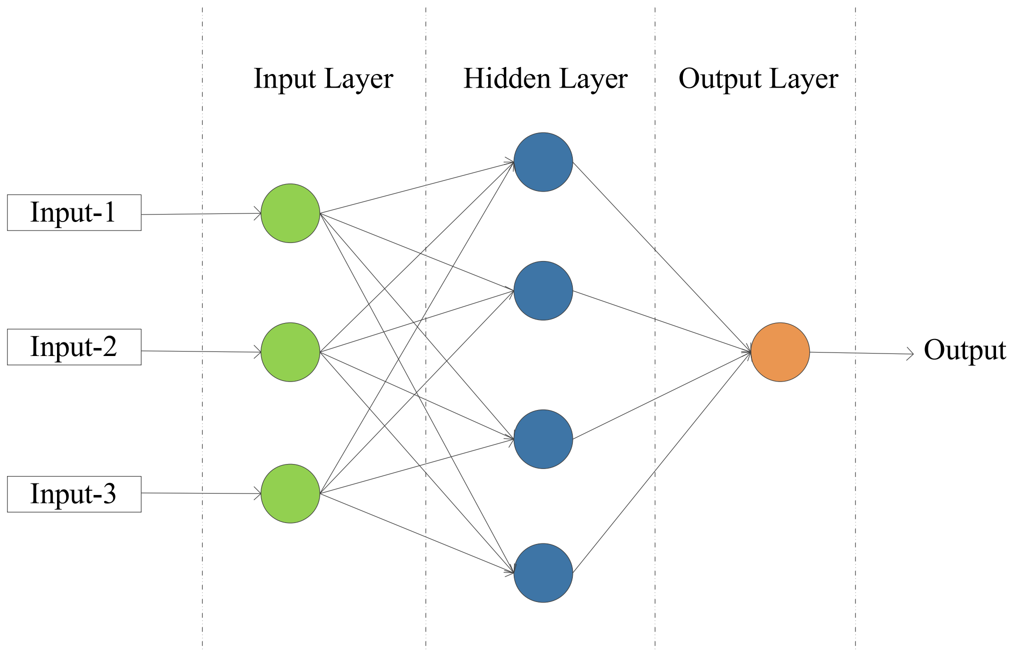


Figure 2.4 – Multilayer neural network

Figure 2.4 shows a multilayer feed-forward artificial neural network. A multilayer artificial neural network consists of fully or partially connected neurons and often can perform as an effective nonlinear model. The weight of the connections between the neurons can be learned using a suitable training algorithm. ANNs are used widely for energy demand forecasting [2, 10, 12]. This thesis uses an artificial neural network to build nonlinear forecasting models.

2.2.4 Support Vector Regression

Support vector regression (SVR) is a nonlinear regression technique built on top of the support vector machine technology [22]. Support vector regression uses quadratic programming to find the optimized margins (*i.e.*, the margin that fits the data most accurately). SVR is easily implemented through the support vector machine library [23] and commonly used for energy demand forecasting [3, 13]. Like SVM, it is possible to select different kernel functions for SVR. Selection of a nonlinear kernel function allows modeling the nonlinearity as shown in(2.14), (2.15), and (2.16). The target is to minimize

$$\frac{1}{2}\|w\|^2 \tag{2.13}$$

subject to

$$\hat{y}_t = \langle w + \varphi(x) \rangle + b, \tag{2.14}$$

$$y_i - \langle w + \varphi(x_i) \rangle - b < \varepsilon, \tag{2.15}$$

$$\langle w + \varphi(x_i) \rangle + b - y_i < \varepsilon, \tag{2.16}$$

where ε is the error boundary. This SVR is called ε-SVR.

Figure 2.5 shows an example of an ε-SVR with a linear kernel function. This thesis uses ε-SVR with nonlinear kernel functions to build nonlinear models.

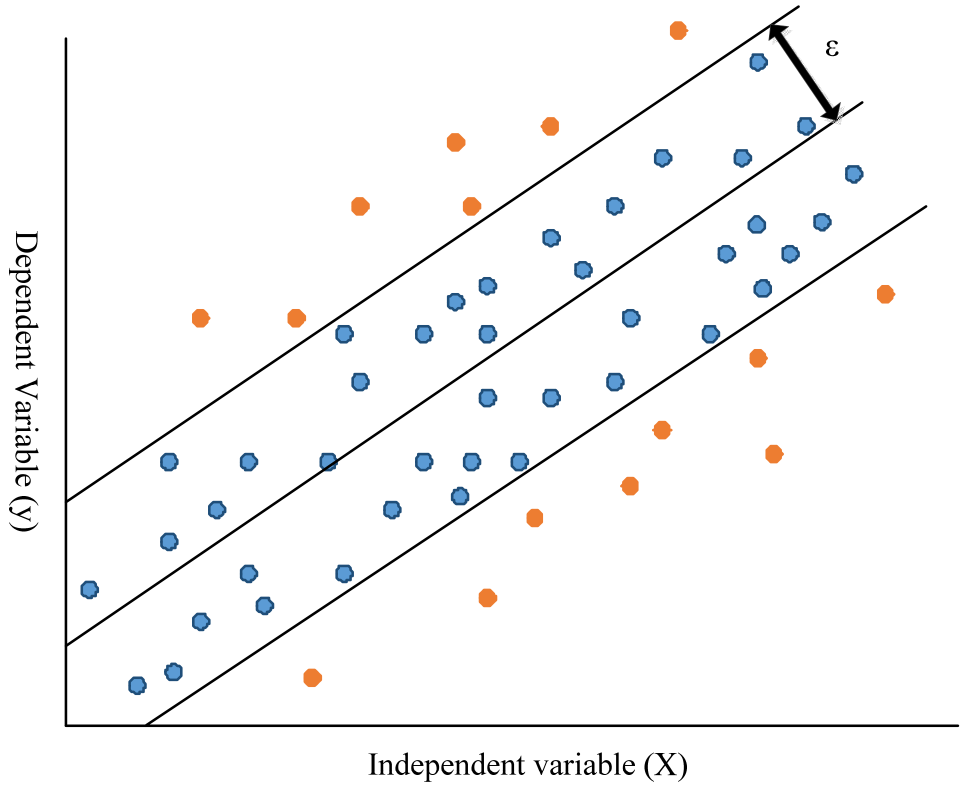


Figure 2.5 – Support vector regressions

Econometrics and statistical techniques are capable of building linear models. On the other hand, the machine learning techniques are suitable for modeling the nonlinearity. This thesis incorporates these techniques. The next chapter describes how these techniques are used in this thesis.

2.3 Literature Review

Econometrics and statistical models and machine learning forecasting techniques are presented in the previous two sections. This section provides a background on how these techniques are used by other researchers. This section also describes works that influence this thesis. This section contains two subsections. The first subsection describes previous works on model order searching of the autoregressive moving average (ARMA) models and autoregressive and moving average with exogenous inputs (ARMAX) models.

The second subsection describes the previous work done using machine learning techniques to build forecasting models.

2.3.1 Previous Work on Searching for the Models

This section describes the approaches that are used by other people to build forecasting models using econometrics and statistical modeling techniques. Building time series forecasting models requires three steps to be followed as shown in Figure 2.6, model detection, model order determination and model parameter estimation.

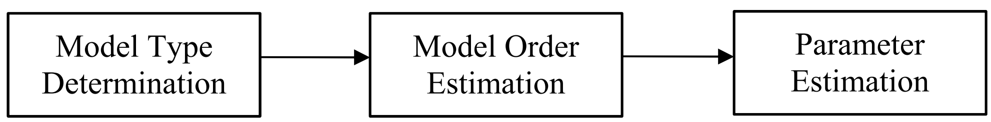


Figure 2.6 – Searching time series forecasting model

Literature searches have found significant research taking place to address the different steps of building of forecasting models. Among these three steps, the order estimation is the most important part, and a significant amount of previous work is available in this area.

Box and Jenkins defined a modeling technique for building an ARMA model [1]. The method suggests using the autocorrelation and the partial autocorrelation values for determining the type and the orders of the time series model. This method can identify the order of the autoregressive (AR) and the moving average (MA) separately, but it is unable to determine the model orders of an ARMA model.

Han-Fu and Zhao [33] proposed eigenvalue analysis of the covariance matrix to estimate the ARMA orders. The process is iterative. Thus, the technique is faster than other available techniques such as brute force search of information criteria value [5-8].

The technique is adaptive with new data. Thus, a recalculation is not necessary when the new data is available. This technique can be extended for exogenous inputs. This technique assumes that the roots for AR and MA parameters are within the unit circle and an upper bound for the orders is known. The eigenvalue analysis of covariance technique works successfully for AR2MA2. For higher orders, the estimated values are differing, and an order correction is needed.

The literature search also found other techniques to discover the orders [34-37] and parameters [38, 39]. The article of Gang, Wilkes and Cadzow [35] analyze the effect of the ARMA root locations in pole zero diagrams and draw conclusions based on their analysis. A similar technique is followed in this thesis. Smadi and Wilkes [37] use a similar technique to discover higher ARMA orders. Many of these ARMA orders and parameter estimation techniques [34-37, 39] use eigenvalue analysis of the covariance matrix, and the technique can be extended to searches ARMAX model.

The information minimization criteria techniques [5-8] are widely used for determining the model type and the orders. An information criterion is defined in the information theoretic as a measurement of the information loss. As the model building process involves generalization, there is a chance that, some sort of information is lost when the modeling process has taken place and it is required to know how much information is lost with the built model. The information criteria technique provides a mechanism to measure this information loss. Apart from representing the information loss, the information criteria value also represents the complexity of the model. Complexity of the model increases the information criteria value. The popular information minimization criteria include Akaike information criterion (AIC) [6], Corrected AIC (AICc) [5], Bayesian information criterion (BIC) [7] and Hannan-Quinn information criterion (HQC) [8].

The information minimization criteria techniques are commonly used to search for best model in the model space [40-42]. Thus, the information criteria technique is successfully applied in building forecasting models in different areas [42-44]. The available information criteria minimization technique suggests a brute force approach when searching the minimum information criteria value for the candidate models in the model space. This thesis presents a search-based approach that improves the computational performance of the brute force approach. The technique is discussed in the following chapter.

2.3.2 Previous Works on the Machine Learning Techniques

Machine learning techniques are widely used for the building the forecasting models [12, 13]. Literature searches have found the previous works where the machine learning techniques are used in combinations with the econometrics models. The results from different techniques are integrated to obtain better forecasting accuracy [3, 15]. For example, the hybridization of ARMA and intelligent techniques introduced by Valenzuela [4] uses the artificial neural network (ANN), genetic algorithms (GA) and the fuzzy logic with the ARMA model [3]. Two different approaches are suggested in the technique. The first approach uses the Box-Jenkins technique [1] and incorporates the fuzzy logic to allow the automatic learning of the model attributes. The second approach uses the ANN with ARMA models. This thesis uses an ensemble regression technique to combine the results obtained using different econometric and machine learning techniques. Details of the ensemble regression technique are provided in the following chapter.

The artificial neural network (ANN) [10, 45-48] and the regression tree (RT) [24, 25] techniques also are widely used for forecasting modeling. This thesis uses the ANN and RT to build individual models and also to build input models. Details are provided in the following chapter. Recent work has shown that use of support vector regression can improve forecasting accuracy [9, 12, 14]. The seasonal decomposition techniques also improve the forecasting accuracy [12, 16]. The seasonal decomposition approach suggested by Shuhai [14] uses the support vector regression techniques to build forecasting models. The technique can model nonlinearities as well as the seasonality. The prediction problems are broken into smaller and simpler prediction sub-problems. These smaller sub-problems are solved using available forecasting techniques. The results are combined together, and the final forecasting result is obtained. The modeling seasonality technique presented in this thesis in the following chapter is motivated by the research work of Shuai [14].

The machine learning techniques are used for feature extraction [49, 50], input preprocessing [50, 51] and knowledge extraction [52, 53], because a nonlinear transformation of the input is needed for these, and the machine learning techniques are capable of performing this nonlinear transformation. This thesis uses both the machine learning and the econometrics techniques for preprocessing of the inputs and to extract information that can represent domain knowledge.

3 AUTOMATIC DISCOVERY OF TIME SERIES FORECASTING MODELS

This chapter presents novel algorithms to automatically generate time series forecasting models for energy demand. To identify the model, the algorithms use statistical time series modeling and machine learning techniques.

Automatic model discovery searches for the optimal time series forecasting model structure and parameters. The optimal forecasting model is the one that generates the minimum forecasting error among the candidate models. The first two sections in this chapter present statistical modeling techniques for searching for suitable autoregressive and moving average (ARMA) and autoregressive and moving average with exogenous inputs (ARMAX) models. The rest of the chapter deals with the application of machine learning techniques such as artificial neural networks (ANN), regression trees (RT), support vector regression (SVR) and ensemble learning. These techniques were introduced in chapter two.

3.1 Discovery of ARMA Models

The ARMA modeling technique commonly is used for energy demand forecasting because forecasted demand is highly correlated with the previous demands [54, 55]. Thus, ARMA is a good initial modeling technique for energy demand forecasting. However, ARMA modeling requires identification of the model orders and estimation of the parameter values.

This thesis examines two approaches to the automatic discovery of ARMA models, the Box-Jenkins method [1] and the eigenvalue analysis of the covariance matrix approach [33]. While the Box-Jenkins method can identify correctly the orders of an autoregressive (AR) or moving average (MA) model, it is unable to identify the orders of a joint ARMA model. The eigenvalue analysis of the covariance matrix [33] can correctly identify the orders of an ARMA model up to an AR2MA2, but it fails for higher order models. Thus a new approach is required for automatically finding the correct ARMA model orders and estimating the model's parameters.

This thesis presents an information theoretic approach for ARMA model order estimation, which yields more accurate order estimates than either the Box-Jenkins or the eigenvalue analysis of the covariance matrix approaches. The Bayesian information criterion (BIC) [7] is used to identify the ARMA model orders. An initial approach to using the BIC is the BRUTE-FORCE-BIC algorithm. BRUTE-FORCE-BIC searches for the model with the minimum BIC value. Figure 3.1 presents an example result of BRUTE-FORCE-BIC for a one-dimensional AR order search. The example model is

$$Y_t = \varepsilon_t + 0.50Y_{t\text{-}1} - 0.40Y_{t\text{-}2} + 0.35Y_{t\text{-}3} - 0.30Y_{t\text{-}4} + 0.25Y_{t\text{-}5}, \tag{3.1}$$

where Y_t is the forecasted value at time t, and ε is a random variable normally distributed with zero mean and unit variance.

The BIC value is calculated as

$$BIC = -2\ln(L) + k\ln(n), \tag{3.2}$$

where k is the total number of model parameters, n is the total number of data points and L is the likelihood of the model. For the ARMA model,

$$k = p + r + 1, \tag{3.3}$$

where p and r are the ARMA orders. The likelihood of the model is autoregressive (AR) or moving average (MA) model, it is unable to identify the orders of a joint ARMA model. The eigenvalue analysis of the covariance matrix [33] can correctly identify the orders of an ARMA model up to an AR2MA2, but it fails for higher order models. Thus a new approach is required for automatically finding the correct ARMA model orders and estimating the model's parameters.

This thesis presents an information theoretic approach for ARMA model order estimation, which yields more accurate order estimates than either the Box-Jenkins or the eigenvalue analysis of the covariance matrix approaches. The Bayesian information criterion (BIC) [7] is used to identify the ARMA model orders. An initial approach to using the BIC is the BRUTE-FORCE-BIC algorithm. BRUTE-FORCE-BIC searches for the model with the minimum BIC value. Figure 3.1 presents an example result of BRUTE-FORCE-BIC for a one-dimensional AR order search. The example model is

$$Y_t = \varepsilon_t + 0.50Y_{t\text{-}1} - 0.40Y_{t\text{-}2} + 0.35Y_{t\text{-}3} - 0.30Y_{t\text{-}4} + 0.25Y_{t\text{-}5}, \tag{3.1}$$

where Y_t is the forecasted value at time t, and ε is a random variable normally distributed with zero mean and unit variance.

The BIC value is calculated as

$$BIC = -2\ln(L) + k\ln(n), \tag{3.2}$$

where k is the total number of model parameters, n is the total number of data points and L is the likelihood of the model. For the ARMA model,

$$k = p + r + 1, \tag{3.3}$$

where p and r are the ARMA orders. The likelihood of the model is

$$L = \arg\max P(D \mid \hat{\phi}, \hat{\theta}, \hat{c}), \tag{3.4}$$

where D is the observed data and $\hat{\phi}$, $\hat{\theta}$ and $\hat{c}$ are the estimated ARMA parameters.

Figure 3.1 shows BIC values from BRUTE-FORCE-BIC for AR orders one to 10. The data D is generated by simulating (3.1). BRUTE-FORCE-BIC finds the minimum BIC value of five, which is the correct order of (3.1).

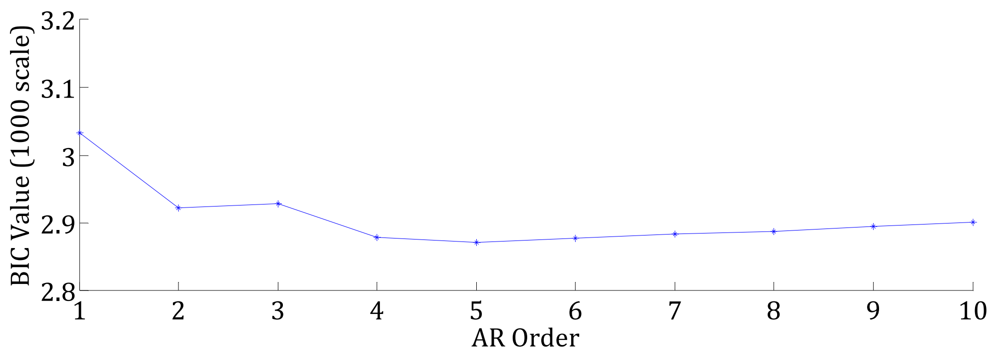


Figure 3.1 – Information criteria minimization to estimate AR order

Similarly, the brute force search technique is applicable to the two dimensional space of AR and MA orders. The brute force algorithm is shown in Algorithm 3.1 –.

```
BRUTE-FORCE-BIC (maxAROrder, maxMAOrder, timeSeries)
minBIC ← ∞
arOrder ← 0
maOrder ← 0
for p = 1 to maxAROrder
        for q = 1 to maxMAOrder
                calculate BIC using (3.2)
                if BIC < minBIC
                        minBIC = BIC
                        arOrder = p
                        maOrder = q
return arOrder and maOrder
```

Algorithm 3.1 – BRUTE-FORCE-BIC

The BRUTE-FORCE-BIC requires substantial computation time [33] because it searches for the model orders using all possible order values. BRUTE-FORCE-BIC requires the maximum estimated AR and MA orders to be known a priori. The calculation of the BIC value is $O\left((p \times q)^2 \times n\right)$, where p and q are the AR and MA order, respectively, and n is the number of data points. Hence, choosing a large maximum order requires substantial computation time, whereas assuming a smaller order boundary may fail to find the correct orders because the actual orders are not examined. Although BRUTE-FORCE-BIC provides a better estimation of the actual ARMA orders than the Box-Jenkins [1] and the eigenvalue analysis of the covariance matrix [33] techniques, there are limitations caused by incorrectly selecting the maximum possible orders and performance issues involved with the order estimation process.

This thesis presents a search-based algorithm that overcomes the limitations of the BRUTE-FORCE-BIC. The new algorithm, SEARCH-BASED-BIC, uses the monotonicity of the BIC with respect to the model orders. SEARCH-BASED-BIC finds the ARMA orders that minimizes the BIC.

SEARCH-BASED-BIC starts with the candidate AR and MA orders of zero. The BIC value is calculated for the current candidate orders (i.e., AR order = 0 and MA order = 0). The algorithm also calculates the BIC value for the neighboring orders for a given box distance. For example, if the box distance is one, the algorithm calculates the BIC value for AR, MA order pairs AR1, MA1 and AR1MA1. Once the BIC values are calculated for all the neighbors, the order pair with minimum BIC is set as the candidate orders. If the previous candidate orders are found having the minimum BIC value, then the process stops, and the current candidate orders are used as the model orders. Otherwise, the algorithm continues exploring the neighbors of the current candidate orders and moves to the neighbor that has the minimum BIC value.

```
SEARCH-BASED-BIC (depth, timeSeries)
minBIC ← ∞
arOrder ← 0
maOrder ← 0
while not (arOrder = prevAROrder and maOrder = prevMAOrder)
        prevAROrder ← arOrder
        prevMAOrder ← maOrder
        for p = max(arOrder–depth, 0) to arOrder+depth
                for q = max(maOrder–depth, 0) to maOrder+depth
                        calculate BIC using (3.2)
                        if BIC < minBIC
                                minBIC = BIC
                                arOrder = p
                                maOrder = q
return arOrder and maOrder
```

Algorithm 3.2 – SEARCH-BASED-BIC

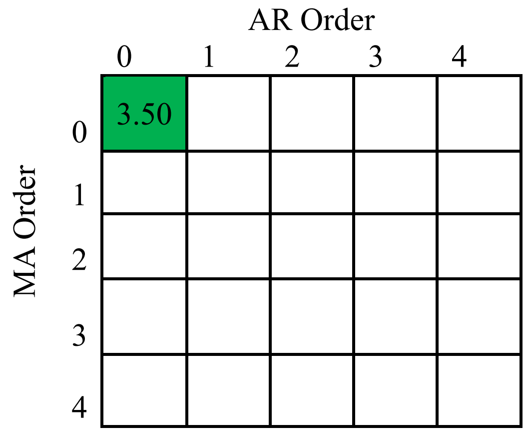
If the BIC is not strictly monotonic with respect to the model orders, SEARCH-BASED-BIC may find a local minimum. For example, in Figure 3.1, the candidate order two has a smaller BIC value than the candidate order three. With depth one, the search would identify the order as two, which is incorrect. Thus, a search depth greater than one may be required. Empirically, we have found that for orders up to 10, a search depth of three avoids local minima. It is also found from the experiment that accuracy of forecasting does not differ significantly if the orders obtained by the search-based approach differ from the actual orders. Chapter four presents more details on this experiment.

Figure 3.2 provides an example of SEARCH-BASED-BIC for finding the ARMA orders of the model

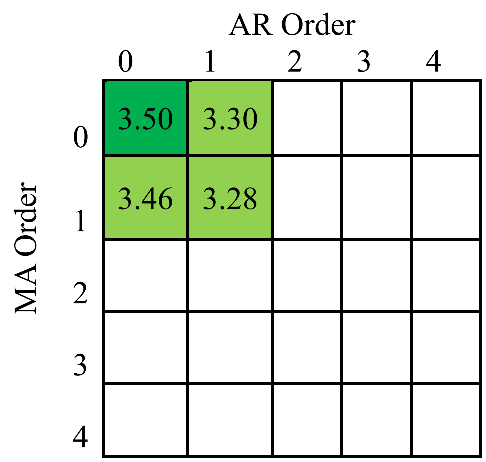
$$Y_t = \varepsilon_t + 0.5Y_{t\text{-}1} - 0.4Y_{t\text{-}2} - 0.5\varepsilon_{t\text{-}1} - 0.4\varepsilon_{t\text{-}2}, \tag{3.5}$$

where Y_t is the forecasted value at time *t,* and ε is random noise normally distributed with zero mean and unit variance. A synthetic dataset is constructed with 2000 points using the autoregressive model, which is representative to the energy demand dataset. There are additional components present in the energy demand dataset other than the autoregressive terms, but, those terms are not of interest for the demonstration of the BIC search technique and are considered as noise.

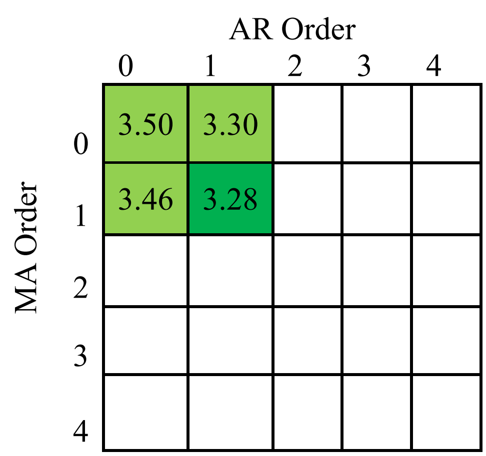
In this example, the algorithm starts searching the ARMA model orders with the candidate AR order and the MA order as zero. The BIC value is calculated for the candidate model (i.e., AR0MA0). Step (a) in Figure 3.2 shows an initialization of the model orders and the BIC value calculation. In step (b), the algorithm calculates the BIC values for all neighbors (within the configured box distance value, which is one) model order pairs. A minimum value is found for the order pair (1, 1), so AR1MA1 is set as current candidate model in step (c). Similarly, in step (d), the BIC values are calculated for all neighboring order pairs. In step (e), the algorithm finds its current candidate model as AR2MA2. This process repeats until the current point is a local minimum BIC value compared to its surrounding BIC values. The algorithm ends at step (e) with AR2MA2 as the winning model. The cells in the tables contain the calculated corresponding 10^{-3} BIC value.



Step (a) starts with candidate orders (0, 0)



Step (b) calculates adjacent BIC values



Step (c) AR1MA1 has the minimum BIC value

MA Order \ AR Order	0	1	2	3	4
0	3.50	3.30	3.00		
1	3.46	3.28	2.95		
2	3.06	2.90	2.85		
3					
4					

Step (d) calculates surrounding BIC values

MA Order \ AR Order	0	1	2	3	4
0	3.50	3.30	3.00		
1	3.46	3.28	2.95		
2	3.06	2.90	2.85		
3					
4					

Step (e) current point moved to orders (2, 2)

MA Order \ AR Order	0	1	2	3	4
0	3.50	3.30	3.00		
1	3.46	3.28	2.95	2.92	
2	3.06	2.90	2.85	2.86	
3		2.87	2.86	2.87	
4					

Step (f) AR2MA2 has the final minimum BIC value

Figure 3.2 – Search-based approach example

MA Order \ AR Order	0	1	2	3	4
0	3.50	3.30	3.00	2.91	2.92
1	3.46	3.28	2.95	2.92	2.90
2	3.06	2.90	2.85	2.86	2.87
3	3.04	2.87	2.86	2.87	2.88
4	2.97	2.87	2.87	2.87	2.88

Figure 3.3 – Brute force search example

Figure 3.3 illustrates the BRUTE-FORCE-BIC, which calculates BIC values for all possible candidate orders. In this example, the maximum AR and MA orders are set as five. Thus, a total of 25 BIC calculations are required by BRUTE-FORCE-BIC, whereas SEARCH-BASED-BIC requires only 14 BIC calculations and avoids the 11 most expensive BIC calculations.

3.2 Discovery of ARMAX Models

The ARMA modeling approach can be extended by adding exogenous terms yielding the ARMAX (Autoregressive moving average with exogenous inputs) modeling approach. The search-based approach is extended to discovering the ARMAX model by searching the product space of $p \times q \times r^m$, where *p*, *q* and *r* are the AR, MA and exogenous orders, respectively, and *m* is the number of exogenous input variables. The advantage of SEARCH-BASED-BIC in comparison BRUTE-FORCE-BIC is even more apparent in searching the higher order ARMAX space. This can be shown by analyzing the complexity of both BRUTE-FORCE-BIC and SEARCH-BASED-BIC. If the maximum order n is assumed for each of the AR, MA and exogenous, terms and *m* is the number of exogenous variables, the BRUTE-FORCE-BIC requires $O\left(n^{m+2}\right)$ BIC calculations. In contrast, SEARCH-BASED-BIC requires $(2d+1)^{m+2}-(2d)^{m+2}$ BIC calculations, which is $O\left(d^{m+1}\right)$, where d is the search depth. Since n is expected to be much greater than *d,* and typically d is a small constant, SEARCH-BASED-BIC is substantially faster than BRUTE-FORCE-BIC.

The extended algorithm for identifying the exogenous orders along with the ARMA orders is Search-Based-ARMAX-Orders.

SEARCH-BASED-ARMAX-ORDERS (*depthVector, orderDimension, timeSeries*)
minBIC ← ∞ *orderVector* ← *Array [size = orderDimension, value = 0]* **while not (***orderVector = prevOrderVector***)**   *prevOrderVector* ← *orderVector*   **for each** *neighbor* **in** findNeighbor (*orderVector, depthVector)*   calculate BIC for *neighbor* using (3.2)   **if** *BIC < minBIC*   *minBIC = BIC*   *orderVector = neighbor* **return** *orderVector*

Algorithm 3.3 – SEARCH-BASED-ARMAX-ORDERS

Once the AR, MA and exogenous orders are known, the parameters are determined by the maximum likelihood parameter estimation technique [56]. Thus, an ARMAX model can be built automatically to forecast energy demand. The SEARCH-BASED-ARMAX-ORDERS does not determine which exogenous terms should be used with the model, but this technique can be adapted to include an additional search option for dropping exogenous terms.

3.3 Incorporating Machine Learning Techniques

The ARMAX model provides a good linear forecasting model. However, the forecast can be improved by incorporating nonlinearity into the model. Energy demand time series data is nonlinear by nature [13]. Learning of domain knowledge is also another important factor to have a robust model. Machine learning techniques are useful for representing the nonlinearity and for learning domain knowledge. To get further improved forecasting accuracy, this thesis examines machine learning techniques to create nonlinear forecasting models. These improved algorithms are discussed below.

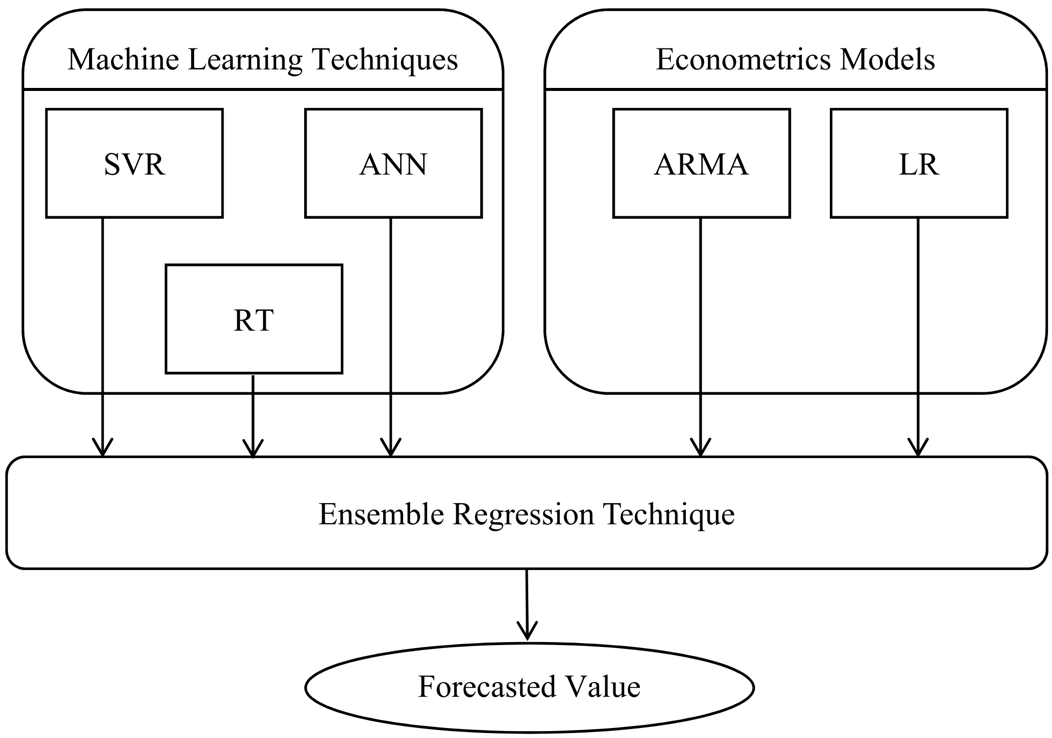


Figure 3.4 – ENSEMBLE-REGRESSION learning

3.3.1 Ensemble Regression Learning

Chapter two presented the application of machine learning techniques to forecast modeling. This thesis combines the forecasts made by different models using ENSEMBLE-

REGRESSION. This thesis uses a search-based approach to find the structure and parameters for the forecasting models. Forecasting results are obtained from time series forecasting modeling techniques such as ARMAX, linear regression (LR) and other machine learning techniques such as artificial neural network (ANN), support vector regression (SVR) and regression trees (RT). The forecasted results from all these techniques are combined together with the ENSEMBLE-REGRESSION technique as illustrated in Figure 3.4.

The technique is represented as

$$\hat{Y}_t = c + \sum_{m=1}^{M} \alpha_m \hat{Y}_t^{(m)}, \tag{3.6}$$

where is the final estimated value, $\hat{Y}_t^{(m)}$ is the output from model *m*, *c* is the intercept and *M* is the total number of models. The ensemble regression algorithm works as follows

ENSEMBLE-REGRESSION (*modelCount, timeSeries*)
/* Training*/ *modelVector ← Initialize with all available models* *outputVector ← Array [size = modelCount, value = 0]* **for** *index = 1* **to** *modelCount*   *outputVector[index] ← mdoelVector[index]*.estimate(*timeSeries*) *coeffVector ← Array [size = modelCount, value = 0]* *[coeffVector, intercept] ←* learnModelCoefficient(*timeSeries*) ***/* Forecasting */*** *forecastedValue ← intercept* **for** *index = 1* **to** *modelCount*   *forecastedValue ← forecastedValue + outputVector[index]*coeffVector[index]* **return** *forecastedValue*

Algorithm 3.4 – ENSEMBLE-REGRESSION

3.3.2 Input Modeling Technique

Forecasting model inputs may be related to the output in either a linear or a nonlinear manner. The nonlinearity may vary from input to input. Modeling the nonlinear relationship between input and output variable is expected to yield a better overall forecast. Figure 3.5 shows the input modeling techniques.

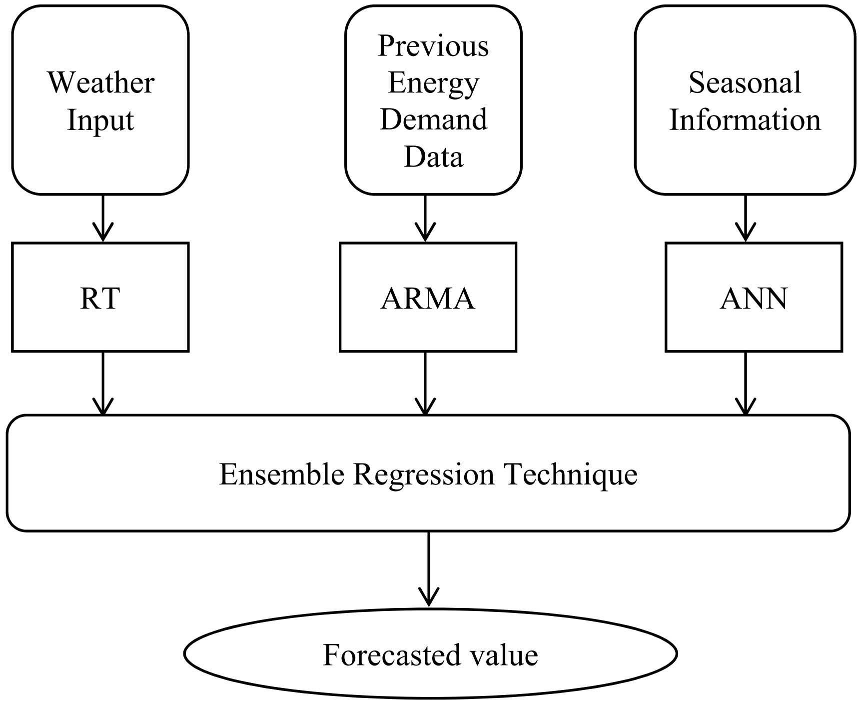


Figure 3.5 – ENSEMBLE-REGRESSION with input modeling

This technique includes two stage learning. In the first stage, a set of statistical and machine learning techniques are used for modeling the inputs. A specific group of candidate models are examined against different input types. The most appropriate model among the candidate models in a group is chosen for each individual input set. For example, different candidate autoregressive and moving average (ARMA) models are tested against the previous energy demand data. The most suitable ARMA model is chosen for modeling the previous energy demand input data. Similarly, the most appropriate regression tree (RT) model and artificial neural network (ANN) model is chosen for the weather inputs and the seasonal information, respectively. The energy demand forecast output is generated from each of those selected models. The second stage contains an ensemble learner that combines the outputs from the first stage.

The input modeling technique can be described mathemtically,

$$W_i \leftarrow \min_{\varepsilon_m}\{m = 1 : M, \varepsilon_m = Model_m(x_i)\}, \tag{3.7}$$

Finally,

$$\hat{Y}_t = c + \sum_{i=1}^{I} \gamma_i W_i, \tag{3.8}$$

where $\hat{Y}_t$ is the final estimated value, W_i is the output from model m, c is the intercept, γ_i is the regression parameter for the i^{th} model and M is the total number of models. The Equation (3.7) describes the technique of selecting a suitable model for an input. The input x_i is tested with all available candidate models. The model that gives the minimum error ε is chosen as the appropriate model for that input. The input modeling algorithm is presented below.

Preprocessing of the inputs is not required in this technique because the machine learning models transform the inputs into the desired nonlinear forms. Hence, this technique is capable of learning where nonlinear transformations for the inputs are needed. Thus, learning of domain knowledge can be achieved by using this technique.

```
INPUT-MODELING (modelCount, inputCount, timeSeries)
/* Training*/
modelVector ← Initialize with all available models
selectedModelVector ← Array [size = seasonCount, value = 0]
inputVector ← Array [size = inputCount, value = 0]
for inputIndex = 1 to inputCount
        minModelError ← ∞
        for modelIndex = 1 to modelCount
                modelError ← modelVector[modelIndex].estimate(timeSeries)
                if modelError < minModelError
                        minModelError ← modelError
                        selectedModelVector[inputIndex] ← modelVector[modelIndex]

outputVector ← Array [size = inputCount, value = 0]
for index = 1 to inputCount
        outputVector [index] ← selectedModelVector[index].estimate(timeSeries)
coeffVector ← Array [size = inputCount, value = 0]
[coeffVector, intercept] ← learnModelCoefficient(timeSeries, outputVector)

/* Forecasting */
forecastedValue ← intercept
for index = 1 to inputCount
        output ← selectedModelVector[index].estimate(timeSeries)
        forecastedValue ← forecastedValue + output *coeffVector[index]
return forecastedValue
```

Algorithm 3.5 – INPUT-MODELING

3.3.3 Modeling Seasonality

Energy demand data is highly influenced by the seasons [14, 16]. Figure 3.6 shows how the daily natural gas demand is changed based on the season. The figure shows the daily natural gas demand for a city in the U.S. Clearly, the trend in one season is different from the trend in another season. Hence, it is expected to get simpler and more precise models if the dataset is divided into smaller parts based on the seasons.

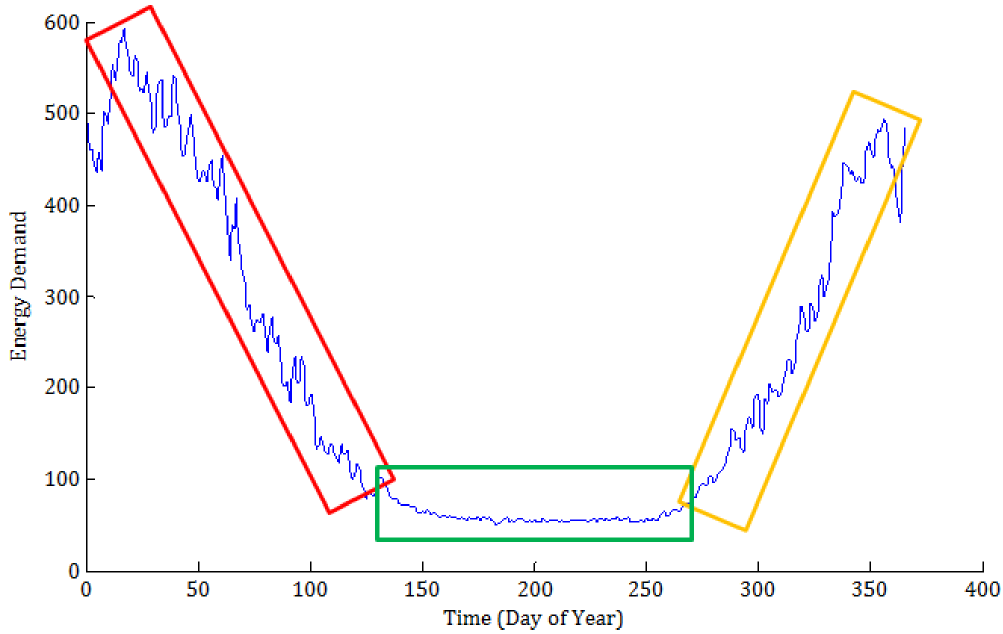


Figure 3.6 – Dividing the data based on seasons

In this technique, the energy demand dataset is separated and divided into multiple datasets based on the seasonal or periodic characteristic (*e.g.*, seasons, months, or days in a week). Each separated dataset is modeled individually using a model search technique. A model is selected based on the minimum error. The outcome of the model selection is used as training data for a set of artificial neural networks (ANN). The day of year and the temperature are used as inputs for the ANNs. Each of the ANNs decides a season in which a particular day belongs. The output from each seasonal model is then selected by the ANNs using majority voting ensemble learning technique. Figure 3.7 illustrates the technique.

The number of seasons is an input for the modeling seasonality algorithm. To obtain better accuracy, a different number of seasons can be tested, and the one with the minimum error should be selected. Thus a 'number of seasons can be determined'. This thesis uses three seasons as an example.

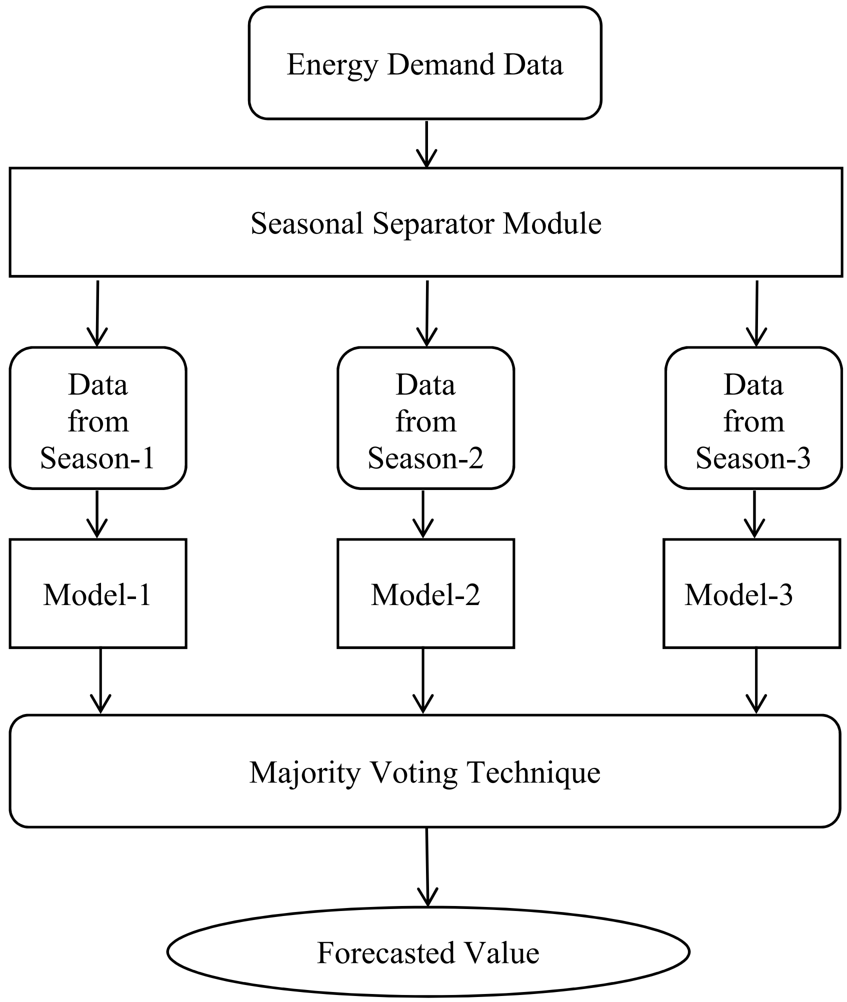


Figure 3.7 – Learning by modeling seasonality

This modeling seasonality technique includes three-stage learning. The first stage is a separator module that separates the energy demand forecasting data into different datasets based on the seasonal information such as the day of year and the temperature of the day, an ANN is trained and used for the separation. The next stage contains a set of statistical and machine learning techniques that are used for modeling each of the seasonal datasets. A suitable model is searched and selected for each of the individual dataset. The third stage contains a majority voting module that determines which output to select. The temperature and the day of the year are used for making the selection.

The modeling seasonality technique can be described mathemtically, for each seasonal dataset $X_s \in D$

$$Model_s \leftarrow \min_{\varepsilon_m} \left\{ m = 1 : M, \varepsilon_m = Model_m \left(X_s \right) \right\}, \tag{3.9}$$

Finally,

$$\hat{Y}_t \leftarrow \min_{\varepsilon_s} \left\{ s = 1 : S, \varepsilon_s = Model_s \left(X_t \right) \right\}, \tag{3.10}$$

where $\hat{Y}_t$ is the final estimated value, $Model_s$ is the output from model *m* and *M* is the total number of models. The Equation (3.9) describes the technique of selecting a suitable model for the separated seasonal dataset. The input seasonal dataset X_s is tested with all available candidate models. The model that gives the minimum error ε is chosen as the appropriate model for that seasonal dataset. In Equation (3.10), each of the selected models from Equation (3.9) is tested to check which one gives the minimum error. The model that gives the minimum error is considered as the appropriate model for the input set.

This modeling seasonality technique includes three stages of learning; division of the data based on seasonal information, choose a model for each dataset and ensemble the result. The modeling seasonality algorithm is presented below.

```
MODELING-SEASONALITY (modelCount, seasonCount, timeSeries)
/* Training*/
modelVector ← Initialize with all available models
selectedModelVector ← Array [size = seasonCount, value = 0]
dataSetVector ← SeparateTimeSeries(timeSeries, seasonCount)

for seasonIndex = 1 to seasonCount
        currentDataSet ← dataSetVector[seasonIndex]
        minModelError ← ∞

        for modelIndex = 1 to modelCount
                modelError ← modelVector[modelIndex].estimate(currentDataSet)
                if modelError < minModelError
                        minModelError ← modelError
                        selectedModelVector[seasonIndex] ← modelVector[modelIndex]

learnSeasonSelection(timeSeries, featureVector)

/* Forecasting*/
seasonIndex ← selectSeason(timeSeries)
forecastedValue ← modelVector[seasonIndex].estimate(dataSetVector[seasonIndex])

return forecastedValue
```

Algorithm 3.6 – MODELING-SEASONALITY

In this algorithm, the most appropriate model is searched for each individual season. A set of features is selected for learning the season identification. These features are used to learn the season identification models. The ENSEMBLE-REGRESSION is used for selecting the winning season based on the minimum model error. Finally, the model for the winning season is used for calculating the final forecasted value.

The novel techniques introduced by this thesis are described in this chapter. It is important to evaluate the performance of these techniques. The next chapter evaluates the performance of these techniques, and presents the testing methods and the test results.

4 EVALUATION AND ANALYSIS OF THE METHODS

This thesis presents novel techniques to automate energy demand forecasting. These novel techniques include the search-based approach and machine learning techniques that are described in chapter three. This chapter presents the evaluation of these novel techniques. This chapter contains two sections. The first section presents the testing process and the results of the search-based approach. The second section contains the testing and the results of the machine learning techniques.

4.1 Evaluation of SEARCH-BASED-BIC

SEARCH-BASED-BIC searches for the appropriate model in the candidate model space. This approach identifies the orders of an autoregressive and moving average (ARMA), autoregressive with exogenous inputs (ARX) and autoregressive and moving average with exogenous input (ARMAX) models. This section examines the accuracy of the order estimates. The search-based approach overcomes the poor computational performance of the brute force technique. This section also compares the computational performance of the search-based technique with the brute force technique. The first subsection provides details on the accuracy testing. The performance testing follows in section 4.1.2.

4.1.1 Accuracy Testing Method

This section describes how the accuracy of SEARCH-BASED-BIC is evaluated. The accuracy is measured considering two different aspects. First, the accuracy is measured using synthetic datasets, where the orders and the parameters are known. Second, the forecasting accuracy is measured for sample of daily natural gas flow and daily electricity demand. The datasets are described in the respective sections.

Synthetic datasets are used to check the accuracy of the model order estimation using the search-based approach. The orders and the parameters are known for the synthetic dataset. Thus, it is possible to compare the estimated order with the actual order using the synthetic datasets as opposed to the real datasets, where the orders are unknown. However, it is possible to measure the forecasting accuracy for natural gas and electricity datasets.

The next three subsections present the testing of the search-based approach on ARMA, ARX and ARMAX models. The fourth section presents the forecasting accuracy on natural gas and electricity datasets.

4.1.1.1 ARMA Model Testing Method

SEARCH-BASED-BIC determines the orders of an ARMA model. The order estimation accuracy is tested using synthetic datasets with known model orders. A set of 12 synthetic datasets with different ARMA model orders are created using the MATLAB arima function from the econometrics toolbox [57]. The arima function takes the autoregressive parameters, the moving average parameters and the number of data points as inputs. The parameters are generated using a random variable with the constraint of having the polynomial roots within the unit circle so that a stable ARMA system is obtained. Each of these datasets is tested with the SEARCH-BASED-BIC, and the orders are determined. Tables 4-1 and 4-2 show the results obtained from the testing.

Estimated / Actual	1	2	3	4	5	Mean	Variance
1	9	3				1.3	0.2
2	11	1				1.1	0.1
3	8	4	0			1.3	0.2
4	6	2	3	0	1	2.0	1.5

Table 4.1 – AR order confusion matrix

Estimated / Actual	1	2	3	4	5	Mean	Variance
1	8	4				1.3	0.4
2	7	0	5			1.8	0.6
3	5	2	2	3		2.3	0.8
4	4	4	2	2		2.2	0.8

Table 4.2 – MA order confusion matrix

Tables 4.1 and 4.2 represent the estimated orders against the actual known orders. The mean and the variance of the estimated orders are presented in the two rightmost columns, respectively. The variance indicates the uncertainty of the estimated orders. The larger the variance the more uncertain the estimated order is. The estimated orders are often smaller than the actual orders. A possible reason for the under-estimation is that some of the ARMA roots are near the center of the unit circle and do not contribute substantially to the dynamics of the system. Thus, taking an ARMA root from any arbitrary location within the unit circle does not give the expected results. To test this hypothesis, the roots are selected from the right side of the unit circle as illustrated in Figure 4.1

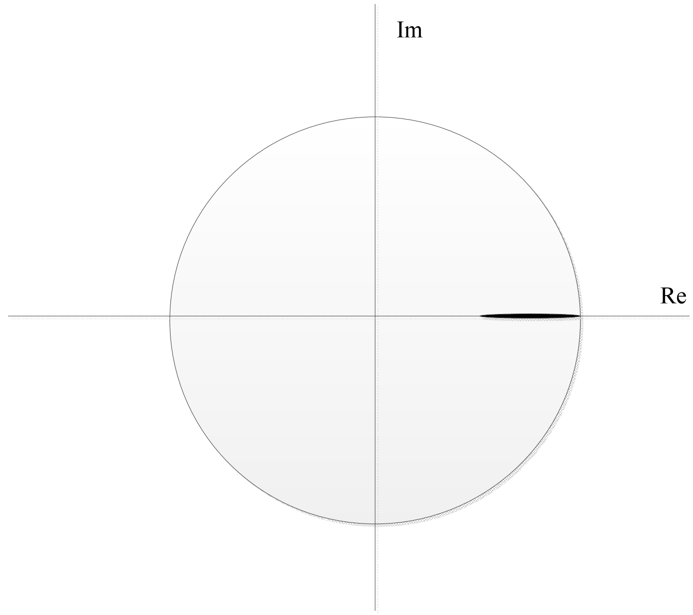


Figure 4.1 – ARMA root selection location in unit circle

The testing is repeated with the ARMA roots taken from the shaded location in Figure 4.1, and the results are shown in Table 4.3 and Table 4.4.

Estimated / Actual	1	2	3	4	5	Mean	Variance
1	6	6				1.5	0.3
2	2	4	6			2.3	0.6
3		6	2	4		2.8	0.8
4		3	1	2	6	3.9	1.6

Table 4.3 – AR order confusion matrix

Estimated / Actual	1	2	3	4	5	Mean	Variance
1	11	1				1.1	0.1
2	5	5	2			1.8	0.5
3	1	2	4	5		3.1	0.9
4		1	4	2	5	3.9	1.1

Table 4.4 – MA order confusion matrix

The average values of the estimated orders are improved and are close to the actual orders, but the individual estimated orders are mostly higher than the actual orders, but still are closer to the actual order. Here, the effects of the parameters are examined by choosing the parameters randomly from the unit circle and the results are compared with the result when the parameters are taken from the selected region. The latter case gives the better result.

To check the impact of choosing a nearby order instead of the actual order, forecasting accuracies are calculated for both the actual and the estimated order. A total of 1000 synthetic data points are generated with random parameters. Among the 1000 points, 500 points are used as data to train the ARMA model parameters for a candidate ARMA orders pair. The remaining 500 data points are used for testing to the forecasting accuracy using those estimated parameters and the candidate orders.

The forecasting error is calculated as mean absolute percentage error (MAPE)

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left(\left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \right), \tag{4.1}$$

where Y_t is the actual output, $\hat{Y}_t$ is the forecasted output and n is the total number of data points.

The reason for using MAPE is that it gives a fair comparison when different datasets are used and when the unit of measurements are not the same. This thesis uses natural gas, electricity, and synthetic datasets with different units of measurements. Hence, it is reasonable to calculate the MAPE rather than using other units. The model searching process also minimizes the MAPE value for the machine learning techniques, the model with the minimum MAPE value is considered as the appropriate model.

Table 4-5 shows the forecasting error obtained for the actual and the estimated orders.

Actual		Estimated	
Orders	MAPE	Orders	MAPE
1,3	31.02	2,1	30.99
3,1	25.40	2,1	25.40
3,3	23.85	3,1	23.83
4,4	26.32	1,4	26.38
4,3	26.82	3,2	26.83

Table 4.5 – Forecasting error for estimated and actual orders

Table 4-5 shows a small difference in the forecasting error for the estimated and the actual orders, which supports the validity of the results obtained by the order estimation technique.

4.1.1.2 Testing the ARX Model

SEARCH-BASED-ARMAX- ORDERS are used for estimating the orders of an autoregressive with exogenous (ARX) model. A group of 12 synthetic datasets with 1000 points are used for testing the ARX order estimation using the minimum Bayesian information criteria (BIC) value. The exogenous input variable is generated from the normal distribution with zero mean and unit variance. The test results are shown in Tables 4-6 and 4-7.

Estimated / Actual	1	2	3	4	5	Mean	Variance
1	12					1.0	0.0
2	5	7				1.6	0.2
3		4	8			2.7	0.2
4			4	8		3.7	0.2

Table 4.6 – AR order confusion matrix

Estimated / Actual	1	2	3	4	5	Mean	Variance
1	12					1.0	0.0
2	9	3				1.3	0.2
3	4	2	5		1	2.3	1.4
4	3	0	1	8		3.2	1.6

Table 4.7 – Exogenous (X) order confusion matrix

Table 4.6 and Table 4.7 show that the average estimated AR orders are close to the actual orders, whereas the estimated orders for the exogenous variable are smaller than the actual orders.

4.1.1.3 Testing the ARMAX Model

The autoregressive and moving average with exogenous (ARMAX) model is tested with a group of 60 datasets with 1000 data points each. The AR, MA and the exogenous parameters are chosen randomly. MA and exogenous inputs are chosen from the normal random distribution with zero mean and one and 0.25 variance, respectively. The exogenous variables are added with the ARMA model with random weights. Thus, there is a direct correlation between exogenous inputs and the output.

Tables 4-8, 4-9 and 4-10 show the confusion matrix for AR, MA, and the exogenous order.

Actual \ Estimated	1	2	3	4	5	Mean	Variance
1	32	12	11	2		1.7	0.8
2	9	30	8	7	6	2.5	1.4
3		10	35	11	4	3.1	0.6
4		2	25	27	6	3.6	0.5

Table 4.8 – AR Order confusion matrix

Actual \ Estimated	1	2	3	4	5	Mean	Variance
1	59	1				1.0	0.2
2	42	17	1			1.3	0.2
3	36	12	12			1.6	0.6
4	31	12	12	4		3.5	1.0

Table 4.9 – MA Order confusion matrix

Actual \ Estimated	1	2	3	4	5	Mean	Variance
1	55	4	0	1		0.8	0.3
2	29	27	4	0		1.6	0.4
3	30	9	20	1		1.9	0.9
4	22	1	21	12	4	2.6	1.8

Table 4.10 – X Order confusion matrix

The results from Table 4.8,Table 4.9, and Table 4.10 show that the orders are not estimated reasonably. Use of the similar normal distribution for the exogenous and the error terms can be a reason for such results, since the system is not able to distinguish between the exogenous input and the noise. Further analysis shows that, if a regular linear or nonlinear signal (e.g., sine or cosine values of running numbers) is used as exogenous input, the AR and the MA orders are estimated almost precisely, but the exogenous orders are incorrect.

4.1.1.4 Testing the Forecasting Accuracy Using Real Dataset

The accuracy of the search-based approach is tested with the natural gas and electricity datasets. The gas dataset consists of daily natural gas usage and temperature for 2800 days from a specific location in the United States. The electricity dataset consists of daily load and temperature for 800 days from another specific location in the United States. For both the datasets, an ARX model is built with temperature as the exogenous input. A nonlinear transformation known as heating degree day [45] is made to the temperature using the below formula

$$HDD_t = \max\left(0, T_t - T_{ref}\right), \tag{4.2}$$

where T_t is the temperature at day t and T_{ref} is the reference temperature.

The heating degree days (HDD) with value with reference temperature as 55^0F and the Fourier terms (sine, cosine) of the day of the year and the day of the weeks are also used as exogenous inputs. The first 50% points from the datasets are taken as training datasets, the next 30% are taken as development datasets and the remaining 20% are taken as testing datasets. The models are trained with the training datasets. The trained models are evaluated using the development datasets. The training and the evaluation processes are repeated with different number of previous terms for the energy demand and the temperature. The model with the minimum BIC value is chosen as the appropriate model. The winning model is retrained using the training and the development dataset and is evaluated using the test dataset. Thus, the forecasted values are obtained.

The SEARCH-BASED-BIC approach finds the minimum BIC value for AR order 36 and exogenous order three for the gas dataset. The forecasting error is calculated for the minimum BIC orders, and the result is shown in Table 4-11.

	BIC	MAPE
Initial model	3.25e+04	9.16
Minimum BIC	3.21e+04	6.36

Table 4.11 – Comparison of the forecasting error for the gas dataset

For the electricity dataset, the search-based approach finds out AR order as 26 and exogenous order as two with a MAPE of 8.10%. Thus, searching the model space may find a model that gives a better accuracy compared to the initial model.

4.1.2 Computational Performance Testing

The SEARCH-BASED-BIC approach estimates the orders faster than the BRUTE-FORCE-BIC technique without compromising accuracy. The performance improvement can be measured by comparing the number of BIC calculations required by the brute force technique with the number of BIC calculations required by the SEARCH-BASED-BIC approach. In terms of computational complexity, the calculations of the BIC are the most expensive operation is the model search process. Other tasks such as iterating through the loop, variable initialization, and comparison of values have constant time complexity and are negligible when compared to the BIC calculation time complexity, which is $O\left(n^2\right)$, where n is the number of parameters.

To compare the number of BIC calculations, a group of 48 ARMA datasets are generated with random orders with the range from one to 6, and the maximum order is assumed as 10. The orders are estimated using both the BRUTE-FORCE-BIC and SEARCH-

BASED-BIC algorithms. The total number of BIC calculations is measured for order depth from one to three. Table 4.12 shows the number of average BIC calculations required by SEARCH-BASED-BIC for various search depths.

Search Depth	Average number of BIC calculations for BRUTE-FORCE-BIC	Average number of BIC calculations for SEARCH-BASED-BIC	Frequency that SEARCH-BASED-BIC found the same minimum as BRUTE-FORCE-BIC
1	100	40.0	81.3%
2	100	62.2	97.9%
3	100	75.4	100.0%

Table 4.12 – Performance improvement by the SEARCH-BASED-BIC approach

Furthermore, the BIC calculations saved by the search-based approach are for the higher orders, which require more time to calculate. Hence, the overall performance improvement is achieved in terms of the time complexity.

4.2 Evaluating the Machine Learning Techniques

This thesis presents novel techniques to improve the forecasting accuracy by incorporating machine learning techniques. The forecasting accuracy is calculated for these techniques and is compared with the commonly used techniques, including autoregressive and moving average (ARMA) models, artificial neural networks (ANN), regression trees (RT) and linear regression (LR) models. The result also is compared with the result from the Marquette University GasDay™ linear model. Most of the experiments in this section are performed with the natural gas or electricity datasets. Some experiments are done with a synthetic dataset for demonstration purposes. To make a fair comparison, all techniques use the same input sets (i.e., same number of autoregressive terms, same exogenous inputs). The synthetic dataset is defined specifically where needed. The natural gas and electric datasets use five autoregressive terms with temperature and day of year as exogenous inputs. Temperature is transformed using (4.2), and Fourier terms (sine and cosine values) of the day of the year are taken when domain knowledge is presumed to be available. If a different set of inputs other than those mentioned above are used an experiment, they are described with the respective testing.

This section contains four subsections. The first subsection presents the forecasting accuracy for the individual techniques. The next three subsections present the testing details of the techniques presented by this thesis such as ENSEMBLE-REGRESSION technique, input modeling technique, and modeling of the seasonality.

4.2.1 Individual Techniques

Individual forecasting techniques such as autoregressive and moving average (ARMA), artificial neural network (ANN), linear regression (LR), and regression tree (RT) are used. Each of the individual techniques uses the available natural gas and electricity datasets as input, and the availability of the domain knowledge is presumed. Each of the natural gas and the electricity datasets is divided into training, development and testing with the ratio of 50%, 30% and 20%, respectively. For each of the techniques, the training datasets are used to train the model. Then the models are evaluated using the development datasets. The process is repeated by varying the number of previous terms for the energy demand and the temperature. The final models are selected based on the minimum BIC and MAPE values (for the econometrics and the machine learning models, respectively) obtained during the development phase. The final models are retrained with the combined training and development dataset and are evaluated using the test datasets.

The forecasting test results for the natural gas and electricity demand are presented in Table 4-13 and Table 4-14, respectively. The results are generated using 16 autoregressive terms, the temperature of the day is transformed using Equation 4.2 and the seasonal information such as the day of year and the day of week with their Fourier values (sine and cosine).

	ARMA	ANN	LR	RT
MAPE	15.28	8.13	7.58	8.11

Table 4.13 – Forecasting accuracy for individual techniques for natural gas dataset

	ARMA	ANN	LR	RT
MAPE	9.13	7.27	8.10	6.89

Table 4.14 – Forecasting accuracy for individual techniques for electricity dataset

If the search-based approach is applied for each of these techniques, as it is done for ARMA in section 4.1.1.4, a better result is obtained. The search-based approach is applied for the natural gas dataset and the result is shown in the table.

	ARMA	ANN	LR	RT
MAPE	15.37	6.51	6.36	7.76

Table 4.15 – Forecasting accuracy for individual techniques for natural gas dataset

4.2.2 Ensemble Regression

Results obtained from different econometrics and machine learning techniques are combined using the ENSEMBLE-REGRESSION technique described in section 2.2.1. The generalized linear model (GLM) is used for ENSEMBLE-REGRESSION. A quadratic model is used with the MATLAB default canonical link function [58] that gives the minimum error. The MATLAB default canonical function is 'identity' (normal distribution). The empirical result shows that the canonical function gives better result than other link functions such as 'log' (poisson distribution), 'logit' (binomial distribution).

The ENSEMBLE-REGRESSION is applied to the gas and electricity demand datasets. The first 80% of the data points are used for the training, and the remaining are used for the testing. The training process starts with training the individual models, which is described in the previous section. The outputs from the individual models are taken as inputs for the ensemble regression technique, which is trained using a generalized linear model.

The testing process starts by obtaining results from individual models using the test dataset. The results from the individual models are passed to the ensemble regression module, and the final forecasted values are obtained. The MAPE forecasting errors are 6.26% and 6.05%, respectively, which is better than any of the respective individual modeling techniques. Hence, the forecasting accuracy is improved by the ENSEMBLE-REGRESSION technique.

4.2.3 Modeling the Inputs

The ENSEMBLE-REGRESSION technique described in the previous subsection uses all available inputs for each of the individual models, whereas the input modeling technique uses a single input set instead of the complete input set for each of the individual models. Thus, each of the input sets is modeled individually using different modeling techniques and are combined using ENSEMBLE-REGRESSION. The technique is capable of learning domain knowledge, and testing is performed to demonstrate that. The technique is tested with the synthetic dataset, and the results are verified with the real gas and electricity datasets. The synthetic dataset is created using the Equations.

$$y = 3y_1 + 15y_2 + y_3 + y_4, \tag{4.3}$$

$$y_1 = \sin\left(doy \times 2pi/360\right), \tag{4.4}$$

$$y_2 = \max\left(rand - 1\right), \tag{4.5}$$

$$y_3 = AR2MA2, \tag{4.6}$$

$$y_4 = 5, \tag{4.7}$$

where *doy* is the day of the year, y_1, y_2, y_3, y_4 are the preprocessed input components and y is the final output. The AR2MA2 model in (4.6) is generated with zero mean and unit variance.

The purpose of using the synthetic dataset here is to demonstrate how the technique is capable of modeling the nonlinear transformations of inputs. Hence, an input with a nonlinear transformation is sufficient to demonstrate this fact. However, additional terms are included to make the synthetic dataset closer to the real dataset. The synthetic dataset is constructed using the autoregressive terms, the nonlinear transformed value of the seasonal information such as the day of year and a random input value that represents exogenous inputs. Similar components are present in the real energy demand datasets.

A total of 1000 data points are created, and following results are obtained. The variables y_1, y_2 and y_3 are modeled using support vector regression (SVR), regression tree (RT) and the ARMA model, respectively. Table 4.16 shows the results obtained from the testing.

	Benchmark		Input Modeling
	With domain knowledge	Without domain knowledge	
MAPE	37.20	68.24	31.23

Table 4.16 – Forecasting accuracy for input modeling using the synthetic dataset

To compare the performance, two sets of benchmarks are created using the linear regression model. One benchmark technique uses domain knowledge, and the other does not. The test result shows that the input modeling method proposed by this thesis exceeds the accuracy of both of the benchmarks. The testing is performed on the same dataset as it is done with the benchmark techniques.

To check the applicability of the technique in real data, this testing is repeated with the gas and electrical datasets, using sixteen autoregressive terms with temperature and day of year as exogenous inputs. Temperature is transformed using (4.2), and Fourier terms (sine and cosine values) of day of year are taken when the domain knowledge are presumed to be available. For the real datasets, the list of the modeling techniques for each input set is presented in Table 4.17.

Variable	Technique
AR terms	ARMA
Temperature	RT
Day of year	RT
Day of week	ANN

Table 4.17 – Modeling technique for individual input

Table 4.18 shows the test result for the gas dataset by using above input-model sets. Table 4.19 represents the result for the electricity dataset also by using the same input-model sets. For the electricity datasets, the input modeling technique exceeds the forecasting accuracy of both the benchmarks. For the natural gas dataset, the input modeling technique is far more accurate than the benchmark of no domain knowledge. The accuracy of the input modeling technique is close to the accuracy of benchmark of having domain knowledge.

	Benchmark		Input Modeling
	With domain knowledge	Without domain knowledge	
MAPE	7.58	27.13	7.83

Table 4.18 – Modeling technique for individual inputs for natural gas

	Benchmark		Input Modeling
	With domain knowledge	Without domain knowledge	
MAPE	8.10	11.39	5.86

Table 4.19 – Modeling technique for individual inputs for electricity

Further analysis on the temperature input shows that a nonlinear transformation is made to the temperature by the input modeling technique, which is similar to having domain knowledge that suggests a similar nonlinear transformation in the temperature. Thus, domain knowledge is learned by the input modeling technique. Figure 4-2 shows the normalized temperature with the values between -1 to 1 and its nonlinearly transformed output for the gas dataset.

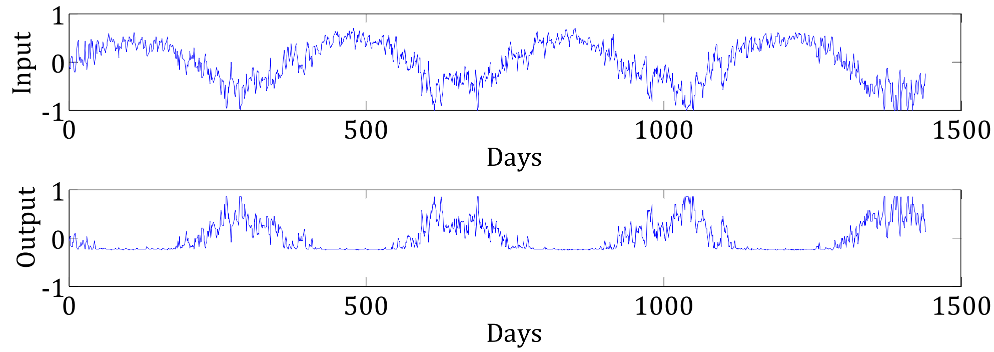


Figure 4.2 – Input and output patterns for scaled temperature for the gas dataset

If we plot the actual temperature against the individual model output (i.e., the preprocessed and nonlinearly transformed input temperature) in Figure 4.3, we can observe the similar representation of domain knowledge as presented by Equation 4.2. Thus, the input modeling technique is capable of learning domain knowledge.

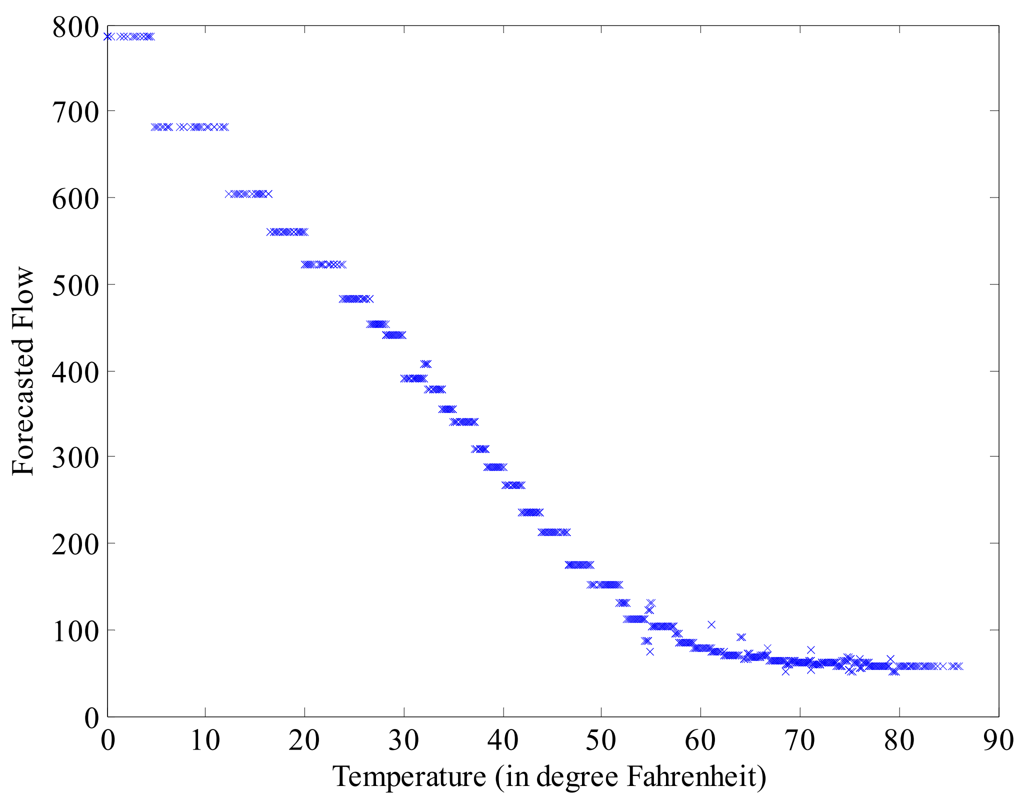


Figure 4.3 – Modeling of the temperature for gas

Figure 4.3 shows the nonlinear transformation on the temperature. This nonlinear transformation provides better forecasting accuracy compared to the output when actual temperature is used as an input.

When tested with the electricity dataset, the transformation of the temperature shows different behavior, presented in Figure 4.4. The behavior is also consistent with domain knowledge of electricity. Unlike natural gas, where the demand becomes nearly constant after a certain threshold of temperature, the electricity demand increases after the threshold temperature, as suggested by Figure 4.4.

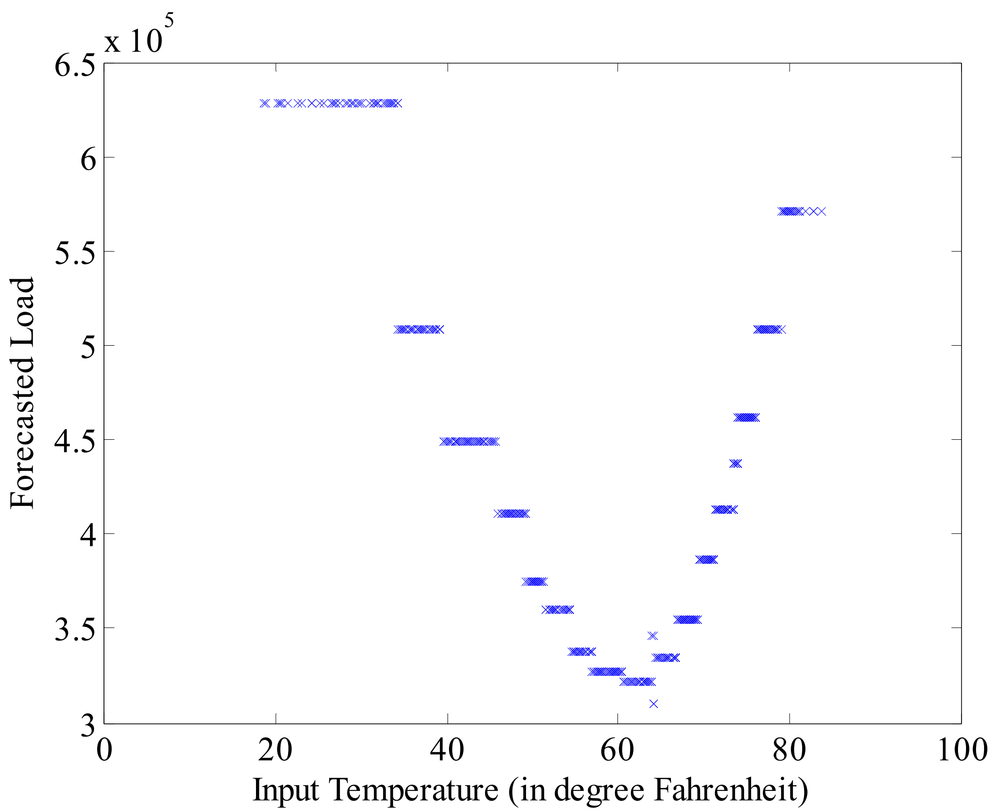


Figure 4.4 – Modeling of the temperature for electricity

One of the problems of using the input modeling technique is that the output from the regression tree is discrete instead of being continuous. Even though other continuous methods are tested with the temperature, and regression tree provides the best result. Further research with other continuous models is needed. Another potential problem with the regression tree is that it may not perform well for the unusual cold or hot days. Test results show that, using the regression tree, the output does not change significantly for exceptionally cold or hot temperatures. This is also an opportunity for further research.

4.2.4 Modeling Seasonality

The energy demand datasets are broken into three seasons. The decomposition of the dataset is performed using a neural network classifier. Temperature and the day of the year are fed as the inputs of the classifier module. A support vector machine was tested as a classifier, but neural networks provided better accuracy. Once separated, each of the three seasonal datasets is modeled individually, and then one of the models' outputs is selected by the weighted voting ensemble technique. The combined result, along with the individual seasonal output, is shown in Tables 4-10 and 4-21 for the gas and the electricity datasets, respectively.

Modeling Seasonality	MAPE
Complete dataset	6.17
Season-1	6.13
Season-2	5.60
Season-3	6.15
Combined	5.99

Table 4.20 – Gas demand forecasting error for modeling seasonality

Modeling Seasonality	MAPE
Complete dataset	8.10
Season-1	6.48
Season-2	7.46
Season-3	11.55
Combined	7.11

Table 4.21 – Electricity demand forecasting error for modeling seasonality

For both the gas and electricity datasets, the combined results shows that the accuracy is improved after applying the modeling seasonality technique.

The MAPE values presented in this chapter are calculated for the test dataset. The result is calculated using the Gasday linear regression model for the same natural gas dataset with additional weather inputs such as precipitation, dew points and wind speed. For the test dataset, the MAPE of the Gasday linear regression model is 4.09%. The best result from this thesis is not able to exceed this accuracy. However, the models presented in this thesis made significant improvement for the simpler models. These techniques are also capable of learning domain knowledge and can provide significant accuracy when domain knowledge is not present. Further research into the techniques presented in this thesis should be able to build more complex models and should achieve more accurate forecasting results.

Comparative analysis among the above three techniques shows that the ENSEMBLE-REGRESSION technique works best for the gas datasets, whereas the input modeling techniques provides the best results for the electricity demand forecasting. The relative accuracy of the input modeling technique is lower compared to the other two techniques because the input modeling technique is not using historical temperature data, whereas the other two techniques are using that. Further research and analysis is needed to use the historical data with the input modeling technique. The modeling seasonality technique improves the accuracy for both the natural gas dataset and the electricity demand dataset. Overall, the modeling seasonality technique provides the best performance among all three machine learning techniques introduced in this thesis.

5 CONCLUSIONS

This thesis contributes to the automation of energy demand forecasting by introducing novel techniques. These novel techniques include searching the model space for a suitable model, machine learning techniques for improving the forecasting accuracy, and techniques for learning domain knowledge from the data. The previous chapter includes the test results from these novel techniques. The test methods and the test results demonstrate the contribution of this thesis towards the complete automation of energy demand forecasting.

This chapter provides a conclusion to the thesis. The first section presents the summary of the thesis work done. Future work is described in the next section.

5.1 Summary of the Thesis

This thesis achieved two goals that contribute towards the automation of energy demand forecasting; 1) to search for the appropriate model in the candidate model space and 2) to improve the overall forecasting accuracy. To have a completely automated energy demand forecasting system in place, the complete model space search is necessary over which the maximum accuracy can be obtained. This thesis presents techniques for searching the model space and for improving the energy demand forecasting accuracy advancing the automation of energy demand forecasting.

From the test results, the order search algorithm offers a performance improvement compared to the brute force method without compromising accuracy, as shown in section 4.1.2. Overall, the BIC search technique does not show a good result for the ARMAX model when compared with the results for the ARMA and the ARX model. To identify the problem, further analysis is required for the ARMAX order estimation technique.

The test results show that forecasting accuracy is improved when machine learning and econometrics modeling techniques are combined together using ENSEMBLE-REGRESSION as introduced in this thesis. However, the accuracy of the ENSEMBLE-REGRESSION is dependent on the accuracy of each of the individual model outputs. Increasing the accuracy of the individual models should lead into further improvement of the overall results.

The input modeling technique is capable of input preprocessing. This technique also is able to learn and represent domain knowledge from the data. The input modeling technique even shows better accuracy than the linear model using prior domain knowledge, because the input modeling technique is capable of modeling the nonlinearities.

The modeling seasonality technique divides the dataset based on seasons using an intelligent decomposition system. These divided datasets are model individually, and are combined by ensemble, yielding improved accuracy.

5.2 Future Work

Test results show that performance and accuracy improvements are achieved by the novel techniques proposed by this thesis, but there is room for improvement. This section describes ideas that might help to obtain better results.

The SEARCH-BASED-BIC algorithm starts from the lowest candidate orders. It can be worthwhile to use the estimated order from another technique and then start the search from that point. Additional performance improvement is expected by this approach. The results from the other technique are not necessarily very accurate, but should be reasonably closer to the actual result. Otherwise, the cost of the search will increase.

This thesis searches for the model structure for each of the individual machine learning or econometrics technique. The similar search approach can be applied to the input selection. This thesis uses the same set of inputs for each individual modeling technique. Instead, algorithms could be developed to discard unnecessary inputs. This may contribute towards improving the accuracy as well as reducing the computational complexity.

The machine learning techniques introduced by this thesis use the historical energy demand data and the current temperature as an input. It is also important to include historical temperatures and also other relevant inputs, such as other weather variables and economic variables. Each of these new set of inputs should be modeled using different techniques and should be integrated. Also, it can be useful to build mechanisms for learning more complex domain knowledge using the additional sets of inputs.

The modeling seasonality technique is highly sensitive to the initial seasonal boundary selection. Different initial boundaries can be tested. Also, this thesis uses simple modeling techniques to model the individual separated datasets. It is worth investigating more complex approaches such as the other techniques introduced in this thesis (*e.g.*, ensembling or input modeling) for modeling the individual dataset, and a better forecasting accuracy may be obtained. Further dividing the time series data can be another improvement. This thesis was tested by dividing the dataset into three seasons; the number of divisions can be increased to check the effect. Other than seasons, datasets can be divided based on weekly or other periodic factors.

For the test results presented in this thesis, statistical significance is not tested. To check the statistical significance of the results of the order search techniques presented in this thesis paper, a chi-square test can be performed. The null hypothesis for this test is “the estimated orders are equal to the actual orders”. However, the statistical significant test for the time series requires significant effort. A complex mechanism is required for the cross validation tests of time series data.

BIBLIOGRAPHY

[1] G. E. P. Box, G. M. Jenkins and G. C. Reinsel, Time Series Analysis: Forecasting and Control. Hoboken, N.J.: John Wiley, c2008; 4th ed, 2008.

[2] E. Erdogdu, "Electricity demand analysis using cointegration and ARIMA modelling: A case study of Turkey," Energy Policy, vol. 35, pp. 1129-1146, 2, 2007.

[3] Y. Hou, "Forecast on consumption gap between cities and countries in china based on ARMA model," in 3rd International Conference on Intelligent Networks and Intelligent Systems (ICINIS), 2010, 2010, pp. 342-345.

[4] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L. J. Herrera, A. Guillen, L. Marquez and M. Pasadas, "Hybridization of intelligent techniques and ARIMA models for time series prediction," Fuzzy Sets Syst., vol. 159, pp. 821-845, 4/1, 2008.

[5] K. P. Burnham and K. P. Burnham, Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach / Kenneth P. Burnham, David R. Anderson. New York: Springer, c2002; 2nd ed, 2002.

[6] H. Akaike, "A new look at the statistical model identification," IEEE Transactions on Automatic Control, vol. 19, pp. 716-723, 1974.

[7] G. Schwarz, "Estimating the Dimension of a Model," The Annals of Statistics, vol. 6, pp. 461-464, Mar, 1978, 1978.

[8] E. J. Hannan and B. G. Quinn, "The Determination of the Order of an Autoregression," Journal of the Royal Statistical Society. Series B (Methodological), vol. 41, pp. 190-195, 1979.

[9] K. Kavaklioglu, "Modeling and prediction of Turkey's electricity consumption using Support Vector Regression," Appl. Energy, vol. 88, pp. 368-375, 1, 2011.

[10] A. Azadeh, S. F. Ghaderi, S. Tarverdian and M. Saberi, "Integration of artificial neural networks and genetic algorithm to predict electrical energy consumption," Applied Mathematics and Computation, vol. 186, pp. 1731-1741, 3/15, 2007.

[11] A. Azadeh, S. F. Ghaderi and S. Sohrabkhani, "A simulated-based neural network algorithm for forecasting electrical energy consumption in Iran," Energy Policy, vol. 36, pp. 2637-2644, 7, 2008.

[12] G. Oğcu, O. F. Demirel and S. Zaim, "Forecasting Electricity Consumption with Neural Networks and Support Vector Regression," Procedia - Social and Behavioral Sciences, vol. 58, pp. 1576-1585, 10/12, 2012.

[13] A. Lendasse, J. Lee, V. Wertz and M. Verleysen, "Forecasting electricity consumption using nonlinear projection and self-organizing maps," Neurocomputing, vol. 48, pp. 299-311, 10, 2002.

[14] W. Shuai, T. Ling and Y. Lean, "SD-LSSVR-based decomposition-and-ensemble methodology with application to hydropower consumption forecasting," in Fourth International Joint Conference on Computational Sciences and Optimization (CSO), 2011, 2011, pp. 603-607.

[15] S. Asadi, A. Tavakoli and S. Reza Hejazi, "A new hybrid for improvement of auto-regressive integrated moving average models applying particle swarm optimization," Expert Syst, vol. 39, pp. 5332-5337, April 2012, 2012.

[16] J. C. Lam, H. L. Tang and D. H. W. Li, "Seasonal variations in residential and commercial sector electricity consumption in Hong Kong," Energy, vol. 33, pp. 513-523, 3, 2008.

[17] (2011). X-12 ARIMA reference manual version 0.3. Available: http://www.census.gov/ts/x12a/v03/x12adocV03.pdf.

[18] A. Azadeh and Z. S. Faiz, "A meta-heuristic framework for forecasting household electricity consumption," Applied Soft Computing, vol. 11, pp. 614-620, 1, 2011.

[19] D. K. Ranaweera, G. G. Karady and R. G. Farmer, "Economic impact analysis of load forecasting," IEEE Transactions on Power Systems, vol. 12, pp. 1388-1392, 1997.

[20] B. F. Hobbs, U. Helman, S. Jitprapaikulsarn, Sreenivas Konda and D. Maratukulam, "Artificial neural networks for short-term energy forecasting: Accuracy and economic value," Neurocomputing, vol. 23, pp. 71-84, 12/7, 1998.

[21] L. I. Kuncheva and J. J. Rodríguez, "A weighted voting framework for classifiers ensembles," Knowledge and Information Systems, December 2012, 2012.

[22] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Statistics and Computing, vol. 14, pp. 199-222, August 2004, 2004.

[23] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, pp. 27, April 2011, 2011.

[24] Ying-Chun Guo, "Knowledge-enabled short-term load forecasting based on pattern-base using classification & regression tree and support vector regression," in Fifth International Conference on Natural Computation, 2009. ICNC '09, 2009, pp. 425-429.

[25] H. Mori, N. Kosemura, K. Ishiguro and T. Kondo, "Short-term load forecasting with fuzzy regression tree in power systems," in IEEE International Conference on Systems, Man, and Cybernetics, 2001, 2001, pp. 1948-1953 vol.3.

[26] Weiqiang Wang, Ying Guo, Zhendong Niu and Yujuan Cao, "Stock indices analysis based on ARMA-GARCH model," in IEEE International Conference on Industrial Engineering and Engineering Management, 2009. IEEM 2009. 2009, pp. 2143-2147.

[27] S. Mukhopadhyay, P. K. Panigrahi, A. Mitra, P. Bhattacharya, M. Sarkar and P. Das, "Optimized DHT-RBF model as replacement of ARMA-RBF model for wind power forecasting," in International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013, 2013, pp. 415-419.

[28] Z. Ming, S. Lianjun, T. Kuo and Z. Lin, "Time-sharing based ARMA-GARCH hourly electricity price forecasting approach," in International Conference on Risk Management & Engineering Management, 2008. ICRMEM '08. 2008, pp. 299-304.

[29] X. Li, L. Ding, M. Shao, G. Xu and J. Li, "A novel air-conditioning load prediction based on ARIMA and BPNN model," in Asia-Pacific Conference on Information Processing, 2009. APCIP 2009. 2009, pp. 51-54.

[30] E. Frank, L. Trigg, G. Holmes and I. H. Witten, "Naive Bayes for Regression," Machine Learning, pp. 1-20, 1998.

[31] A. C. de Pina and G. Zaverucha, "Combining attributes to improve the performance of naive bayes for regression," in IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence), 2008, pp. 3210-3215.

[32] Create linear regression models. Available: http://www.mathworks.com/help/stats/linearmodel.fit.html.

[33] H. Chen and W. Zhao, "New Method of Order Estimation for ARMA/ARMAX Processes," SIAM Journal on Control and Optimization, vol. 48, pp. 4157-4176, March 2010, 2010.

[34] A. Al-Smadi and D. M. Wilkes, "On estimating ARMA model orders," in IEEE International Symposium on Circuits and Systems, 1996. ISCAS '96., Connecting the World., 1996, 1996, pp. 505-508 vol.2.

[35] G. Liang, D. M. Wilkes and J. A. Cadzow, "ARMA model order estimation based on the eigenvalues of the covariance matrix," IEEE Transactions on Signal Processing, vol. 41, pp. 3003-3009, 1993.

[36] Z. Z. Wang and D. W. Hu, "A new algorithm for the ARMA model estimation," in Proceedings of the 30th IEEE Conference on Decision and Control, 1991. 1991, pp. 2030-2031 vol.2.

[37] A. Al-Smadi and D. M. Wilkes, "Data-adaptive higher order ARMA model order estimation," in Proceedingson of IEEE on Visualize the Future, Southeastcon '95, 1995, pp. 210-213.

[38] H. Cui and R. Shan, "ARMA model parameter optimized estimate method," in First ACIS International Symposium on Cryptography and Network Security, Data Mining and Knowledge Discovery, E-Commerce & its Applications and Embedded Systems (CDEE), 2010, 2010, pp. 22-26.

[39] H. Chen and W. Zhao, "Identification of both coefficients and orders for ARMAX system," in Proceedings of the 48th IEEE Conference on Decision and Control, 2009 Held Jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. 2009, pp. 7250-7255.

[40] D. Burshtein and E. Weinstein, "Some relations between the various criteria for autoregressive model order determination," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 33, pp. 1017-1019, 1985.

[41] Y. Jiang, Y. Huang and X. Ye, "A new singular value decomposition method for AR model order selection via vibration signal analysis," in Sixth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD '09, 2009, pp. 567-572.

[42] M. A. Metzger and K. Mitsumoto, "Forecasting multivariate time-series: Confidence intervals and comparison of performances of feed-forward neural network and statespace models," in International Joint Conference on Neural Networks, 1991., IJCNN-91-Seattle, 1991, pp. 915 vol.2.

[43] T. B. Cline, "Identification of seasonal water supply forecasting models using akaike's information criterion," in 18th IEEE Conference onDecision and Control Including the Symposium on Adaptive Processes, 1979, 1979, pp. 874-875.

[44] K. Ofuji and S. Kanemoto, "Price forecasting of japan electric power exchange using time-varying AR model," in International Conference on Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. 2007, pp. 1-6.

[45] C. Yan and J. Yao, "Application of ANN for the prediction of building energy consumption at different climate zones with HDD and CDD," in 2nd International Conference on Future Computer and Communication (ICFCC), 2010, 2010, pp. V3-286-V3-289.

[46] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," Expert Syst. Appl., vol. 37, pp. 479-489, 1, 2010.

[47] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," Applied Soft Computing, vol. 11, pp. 2664-2675, 3, 2011.

[48] C. Li and J. Hu, "A new ARIMA-based neuro-fuzzy approach and swarm intelligence for time series forecasting," Eng Appl Artif Intell, vol. 25, pp. 295-308, 3, 2012.

[49] W. Wettayaprasit and P. Nanakorn, "Feature extraction and interval filtering technique for time-series forecasting using neural networks," in IEEE Conference on Cybernetics and Intelligent Systems, 2006, 2006, pp. 1-6.

[50] M. M. Othman, M. H. H. Harun and I. Musirin, "Forecasting short term electric load based on stationary output of artificial neural network considering sequential process of feature extraction methods," in IEEE International Power Engineering and Optimization Conference (PEDCO) Melaka, Malaysia, 2012, 2012, pp. 485-489.

[51] M. Simunek and E. Pelikán, "Temperatures data preprocessing for short-term gas consumption forecast," in IEEE International Symposium on Industrial Electronics, 2008. ISIE 2008. 2008, pp. 1192-1196.

[52] K. J. Siddiqui and S. M. Nugen, "Knowledge based system for weather information processing and forecasting," in Geoscience and Remote Sensing Symposium, 1996. IGARSS '96. 'Remote Sensing for a Sustainable Future.', International, 1996, pp. 1099-1101 vol.2.

[53] Y. Alsultanny, "Successful forecasting for knowledge discovery by statistical methods," in Conference on Information Technology: New Generations (ITNG), 2012 Ninth International, 2012, pp. 584-588.

[54] S. S. Pappas, L. Ekonomou, D. C. Karamousantas, G. E. Chatzarakis, S. K. Katsikas and P. Liatsis, "Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models," Energy, vol. 33, pp. 1353-1360, 9, 2008.

[55] Y. Ohtsuka, T. Oga and K. Kakamu, "Forecasting electricity demand in Japan: A Bayesian spatial autoregressive ARMA approach," Computational Statistics & Data Analysis, vol. 54, pp. 2721-2735, November 2010, 2010.

[56] Estimate ARIMA or ARMAX modl parameters. Available: http://www.mathworks.com/help/econ/arima.estimate.html.

[57] MATLAB arima class. Available:
http://www.mathworks.com/help/econ/arimaclass.html.

[58] MATLAB Generalized Linear Model Documentation. Available:
http://www.mathworks.com/help/stats/glmfit.html.