ELECTRICITY PRICE FORECASTING USING ADVANCED MACHINE LEARNING TECHNIQUES

by

Bhav Sardana

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Applied Energy and Electromechanical Engineering

Charlotte

2020

Approved by:

____

Dr. Maciej Noras

____

Dr. Ümit Cali

____

Dr. Wesley Williams

____

Dr. Michael Smith

ABSTRACT

BHAV SARDANA. Electricity price forecasting using advanced machine learning techniques. (Under the direction of DR. MACIEJ NORAS)

Electricity Price Forecasts have been a very important part of the energy industry for a long time. Traditionally, it was used by utilities, but after the liberalization of electricity markets around the world, they are of prime importance not only for utilities, but also for portfolio managers, aggregators, retailers and generation companies. These different players can plan the dispatch and consumption schedule of power in accordance with price forecasts in order to maximize profit. Apart from this, electricity prices, in general, can exhibit extreme volatility. This volatility can be attributed to the change in electricity demand and the amount of energy generated/used from renewable sources (since the price of power generated from renewable sources can be very low). This calls for better modeling of electricity prices. The aim of this study is to explore various machine learning techniques like linear regression, gradient boosting, random forests, support vector machines and artificial neural networks for forecasting electricity prices of the German market. The main focus of this research is to explore the use of meta heuristic algorithm like particle swarm optimization to search for the best suited feature space for each algorithm.

ACKNOWLEDGEMENTS

This work would not have been possible without the guidance of Dr. Ümit Cali and Dr. Maciej Noras. Dr. Cali's support during the complete time period of my Master's degree was tremendous.

I thank all my committee members Dr. Williams and Dr. Smith. Thank you for your valuable time and feedback given for this work. I would also like to thank all my professors, Dr. Browne (Applied Mechatronics), Dr. Hildreth (Engineering Analysis), Dr. Hong (Technological Forecasting and Decision making, Computational Intelligence), Dr. Williams (System Dynamics) and Dr. Cail (Energy Generation and conversion, Energy Transmission and Distribution) for all the knowledge they have given me during this period.

The support and guidance provided by Dr. David Hill (Urban Institute) during these 2 years was invaluable.

I would also like to thank Vinayak Sharma and all my friends who have supported me during this time.

And last and most important, I want to thank my parents and my sister, without their support, this would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN Artificial Neural Network

CG Conjugate Gradient

EEX European Energy Exchange

ENTSO-E European Network of Transmission System Operators for Electricity

EPEX European Power Exchange

EPF Electricity Price Forecasting

GB Gradient Boosting

LR Linear Regression

OPSD Open Source Power System Data

PSO Particle Swarm Optimization

RF Random Forests

CHAPTER 1: INTRODUCTION

Over the last three decades, the structure of the electricity markets have drastically changed. In 1982, Chile started the liberalization and deregulation of the power markets [1]. In 1990s, many other countries followed this suit. British power sector reorganized in 1990, which included just England and Wales. In 2005, Scotland was included in the British market. Similarly, in 1992, Nordic market was opened for private competition. This just included Norway initially, but was joined by Sweden later. Similar reorganization was done by the Australian and North American markets [2]. In European power markets, this change has led to the setup of power exchanges, through which power is traded.

Electricity is very different from other commodities. It economically not feasible to store electricity. Hence, there is a need of constant balance between the supply and demand in order to keep the power system stable [3]. Along with this special property of the power system, the electricity demand is known to be dependent on the weather conditions, human behavior (business and other daily activities), which is in-turn dependent on the calendar variables like hour of the day, weekday or holidays. This makes price dynamics of the power industry completely different from other industries [4]. Policies also has an influence on the price dynamics in the electricity markets. Because of the Paris Agreement in 2015, which sets a goal of limiting the global increase in temperature to 2^o, Germany has targeted to increase the share of renewable sources in electricity generation. It aims to increase it to 40%-45% by the year 2025, up from 31.7% in 2015. This can have a major impact on the price at which electricity is traded.

Due to these reasons, electricity price forecasts are a primary part of the decision making process of power companies [2]. Electric utilities are the most vulnerable to fluctuations in the power markets. This is because they cannot pass their costs to retail customers. This was evident from the California crises in 2000-2001 [5]. Price forecast which capture the volatility of the market with reasonable accuracy can help power generation, utility companies and even large consumers change their bidding strategy and/or own consumption/production schedule for maximizing their profits [4]. Also, accurate price forecasts can help utilities plan ahead, as over/under contracting can lead to huge losses or even lead to bankruptcy. An improvement in forecast accuracy of 1% can help a 1 GW peak load utility save up to $600,000 per year [6]. These savings can translate to millions of dollars every year for even a medium (5 GW peak load) size utility.

The purpose of this thesis is to investigate the application of various machine learning techniques for electricity price forecasting. Different models are built and tested on the German-Austrian markets. Models are varied by changing the input variables, the main forecasting engine and pre-processing and post-processing techniques.

1.1 Basics of German Electricity Markets

Germany is the largest electricity consumer in Europe. There are four major electricity producers, namely RWE, E.ON, Vattenfall and ENBW, which account for 70%-85% of the total power production [7, 8]. On the transmission side, there are four high voltage grid operators (the transmission system operators or TSOs). The German Power Market is also connected to other European markets, with different liquidity [9]. After the liberalization of the European power markets, there were two main markets established in Germany - wholesale power markets and control power markets [10]. The wholesale trade takes place at the EEX spot market and the derivatives market (futures and options market). The control markets are established to account for smaller imbalances at short notice.

In organized electricity markets, market clearing price (MCP) is determined by matching the demand and supply of electricity [4]. This is usually done in an auction done once every day. The orders for buying electricity are accepted in the increased order of prices until the demand is fulfilled. Similarly, orders for selling are accepted in a decreasing order of prices. This means, generators which offer selling at a low price will be the first ones to supply electricity to the grid. When the demand gets very low, generators often negative bids, as the cost of shutting down and ramping up a generator may be much higher than the loss from negative prices. This can also occur when the production from renewable sources is very high [4].

CHAPTER 2: LITERATURE REVIEW

The review is divided on the basis of techniques used for electricity price forecasting. The major CI techniques used in these papers include artificial neural networks(ANN), support vector machines (SVM), fuzzy inference systems (FIS), meta-heuristic algorithms (including evolutionary algorithms, swarm optimization, simulated annealing, etc.), hybrid techniques. As mostly all meta-heuristic techniques are used in conjunction with one of the other techniques (for parameter optimization), they have been covered in the each technique as a part of hybrid techniques.

2.1 Artifical Neural Networks

ANNs are the most widely used technique for electricity price forecasting. Because of its ability to learn nonlinear relations between various features, it is used either independently, or in combination with other techniques.

Panapakidis and Dagoumas used a feed forward neural network with various different inputs to build multiple short term price forecasting models [11]. They proposed six models with a different combination of input features including day of the week, hour of the day, holidays, hourly price of electricity with different lags (including 24, 25, 47, 48, 72, 96, etc.), hourly load of previous days, previous day's average load, natural gas prices. In all their models, they used Levenberg-Marquardt method for training the neural network. Singhal and Swarup also used ANN with three layers with backward propagation method for learning. Their model performed good on normal days, whereas poor performance was shown on unusual days [12]. Keles et. al. proposed using clustering techniques for selecting features for an ANN forecaster. They used k-nearest neighbours as the clustering algorithm [13]. Khosravi et. al.

tried to quantify the uncertainty of the electricity price prediction by ANNs [14]. They used a bootstrap method, which is essentially a re-sampling technique, for estimating a prediction interval (PI) for the forecasts with different confidence intervals. According to the authors, PIs can help decision making process by explaining the uncertainty of a given model. For Silvano et. al., ANNs did not perform as well as SVM or econometric techniques like ARMA-GARCH [15]. Voronin and Partanen used various techniques out of which one is neural network. Their neural network model performs better than ARIMA, but did not perform as well (in terms of error metrics) as the various hybrid techniques tested by the authors [16]. Anbazhagan and Kumarappan made use of a feed forward neural network (FFNN) for classification of electricity prices [17]. They extended this research by using input transformation using discrete cosine transform (DCT) in [18]. Dudek and Cerjan et. al. used multi layer perceptron (MLP) for making probabilistic price forecasts [19, 20]. Huang et. al. also used multiple layer perceptron (MLP) as one of the many models tested in their paper [21]. Hong and Wu called their NN a "Multi-layer Feedforward network" (MLF). Essentially, it is a FFNN. They used preprocessing techniques including PCA (Principle component analysis) [22]. Unsihuay-Vila et. al. used ANN to compare with their proposed evolutionary model [23]. Kunwar and Kumar used ANN and heuristic scheduling for demand side management [24].

Apart from the above mentioned papers using ANN in its simple form, there are a number of papers which make use of NNs in some form or the other. Shafie-khah et. al. and Chen et. al. proposed a radial bias function neural network (RBFNN) to model non linear components of the price series [25, 26]. Coelho and Santos also used RBFNN with a gaussian activation function [27]. Lin et. al. also used RBFNN. They used a methodology similar to [28] and proposed an enhanced RBFN (ERBFN) by using an orthogonal experimental design (OED) [29]. This showed similar advantages to the methodology in [28]. The nonlinear component was modeled by another NN

- adaptive wavelet neural network (AWNN) by Wu and Shahidehpour in [30, 31]. Amjady et. al. proposed a new, 2 stage neural network [32]. The first stage was initialized using random weights. The calculated parameters and weights were fed to the second stage NN, along with the price forecast of the next hour by the first stage. They called this a Hybrid Neural Network (HNN). Anbazhagan and Kumarappan proposed a simple Recurrent Neural Network (SRNN, also known as Elman network) [33]. This was able to estimate the temporal and spacial patterns more accurately. In this paper, they also compared the proposed technique with various other NNs including fuzzy neural network (FNN), AWNN, and neural network with wavelet transform (NNWT). Lin et. al. proposed a combination of probability NN (PNN) and OED to create an enhanced probability neural network (EPNN) [28]. According to the authors, the proposed model improves the estimation of complex interactions of various factors which arise due to fluctuating load, temperature changes and transfer flow. Keynia suggested using a two stage approach for modeling electricity prices [34]. The technique is based on mutual information (MI) criterion, which selects input features for a composite neural network (CNN). CNN is a combination of multiple NNs with a modified data flow structure. Lahmiri proposed using generalized regression neural network (GRNN) ensemble combined with variational mode decomposition (VMD) [35]. The model was compared with EMD based GRNN. Gollou and Ghadimi used WTNN, based on wavelet transform and NNs [36]. Sharma and Srinivasan proposed a combination of coupled excitable systems and recurrent neural networks to model intermittent peaks in electricity prices, for improving decision making in electric markets [37]. Lago et. al. proposed using a DNN with exogenous inputs for modeling electric prices [38]. They also suggested implementation of feature selection based on Bayesian optimization. Also, the authors showed that modeling multiple markets together improves the forecast accuracy. Dynamic filter weights neural networks are used by [39]. Mandal et. al. used a unique recursive neural network (RNN) [40].

They used similar day classification to model future prices.

2.1.1 Extreme Learning Machines

A relatively new technique in literature, which is based on a single layer feed forward NN, is Extreme Learning Machines (ELM). This is already extensively used for electricity price forecasting. ELMs have the main advantage of faster learning and almost 1000 times faster performance than backward propagation. [26, 41, 42, 43, 44, 45] make use of the new technique. Chen et. al. proposed using ELM, which is a single layer feed forward network, for fast electricity price forecasting [26]. The methodology incorporates bootstrapping to model uncertainty in the system. They used Australian electricity market operator (AEMO) price data as a case study. Wan et. al. proposed a two stage methodology for probabilistic forecasting for electricity prices [41]. In the first stage, ELM was used to obtain a point forecast. In the second stage, noise variance was measured using maximum likelihood method. They used a bootstrap algorithm for prediction interval estimation. The experiments were carried out with AEMO data and compared with BPNN, persistance, and ELM + Bootstraping methods. Their methodology was at-least 100 times faster than conventional bootstraping based NNs. Yang et. al. created a hybrid model combining wavelet transform, KELM based on self adapting PSO and ARMA. The performance was measured using PJM, Australian and Spanish market [42]. In [43], ELM coupled with wavelet technique was used to create a hybrid method for forecasting electricity prices. The accuracy was improved using ensemble methods. WT was used to analyze hidden features of the price time series. The decomposed high and low frequency series have better statistical properties. Multiple models based on WELM were created in order to create ensembles to improve forecast error.

2.1.2 Hybrid Neural Networks

There are many uses of NNs as a part of a hybrid modeling system. The literature covering these models, which are a part of a hybrid system, are discussed in the section covering hybrid algorithms. Wang et. al. proposed a two layer decomposition method and develop a hybrid forecast engine with fast ensemble emperical mode decomposition (FEEMD), variable mode decomposition (VMD), backware propagation NN (BPNN) optimized by a firefly algorithm (FA) [46]. The application of VMD to decompose high frequency intrinsic mode functions (IMFs) generated by FEEMD improved the forecast accuracy. The hybrid model was compared with BP, FA-BP, FEEMD-FA-BP and VMD-FA-BP. Chaâbane combined ANN with auto-regressive fractional integrated moving average (ARFIMA) [47]. This combination leveraged the unique advantages of both the techniques. Xiao et. al. proposed a wavelet neural network (WNN) modified using singular spectrum analysis (SSA) coupled with cuckoo search (CS) for optimizing the parameters of their models [48]. The proposed CS algorithm is Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method. Kim proposed a combination of Cuckoo search and Levenberg-Marquardt for training an FFNN [49]. Another hybrid technique was used by [50] and [51]. They proposed a hybrid neuro evolutionary system (HNES) based on NN, evolutionary algorithms (EA) and a modified data flow. Maximum relevance minimum redundancy principle was used for pre-processing the input features before feeding to the hybrid model. Lago et. al. proposed formulation of a deep learning framework for tackling the unpredictability of electricity prices [52]. The framework composed of 4 models - deep NN (DNN) which is an extension of MLP, hybrid LSTM-DNN model (LSTM - Long short term memory), a hybrid GRU-DNN model (GRU - gated recurrent unit), and convolutional neural network (CNN). For tuning the parameters of these models, a Parzen estimator was used [53]. Abedinia et. al. proposed a combinational neural network based forecast engine [54]. The structure composed of cascaded NNs and each NN has multiple layers. A chemical reaction optimization (CRO) algorithm was used for hyper parameter optimization. Ebrahimain et. al. proposed a WT based ANN. WT was used as a pre-processor to decompose the input series to different frequency components. These components are forecasted by the ANN forecaster [55]. Bento et. al. used a bat algorithm and scaled conjugate gradient descent for quicker and better learning of an FFNN [56]. Gao et. al. use a multi-block forecast engine composed of various combinations of different types of NNs [57]. The whole structure is optimized by an intelligent algorithm like enhanced shark smell algorithm. Wang et. al. used a dynamic choosing ANN (DCANN), which can dynamically chose the inputs to the NN [58]. It is a combination of Self organizing maps (SOM) and ANN. The parameters were optimized using a Cuckoo search algorithm. Singh et. al. proposed a generalized neural network [59]. This method overcomes various limitations of traditional ANNs. Peng et. al. made use of a combination of long-short term memory network and a differential algorithm [60].

2.2 Support Vector Machines

Che and Wang used support vector regression (SVR) to model the non linear component of electricity price [61]. Their proposed model used SVR in conjunction with ARIMA, which was used to model the residual after getting a prediction from the SVR model. Another team that uses a combination of SVM and a time-series method is [62]. They proposed using least square SVM (LSSVM). According to the authors, the function approximation by LSSVM is easier and less complex but also less accurate when used large data-sets. Hence they follow a similar strategy as [61], and used ARIMAX instead of ARIMA as the second model to overcome limitations of LSSVM. Yan and Chowdhury also used a similar methodology and implemented LSSVM-ARMAX hybrid model [63]. Shrivastava et. al. used SVM for estimating a prediction interval [64]. Their model was fast and computationally less extensive. It used SVM for estimation of the lower and upper bounds of the future prices. It was tuned with the help of particle swarm optimization (PSO). They extensively compared the performance for their model to other models including quantile regression (QR), classification and regression trees (CART), bootstrapping, lower upper bound estimation NNs (LUBE-NN). Saini et. al. also utilize the power of SVM for function approximation [65]. They used a genetic algorithm (GA) to optimize its parameters, which are updated every 4 weeks (or 28 days). They compared this model with linear regression and a heuristic method. Sayeghi et. al. proposed a hybrid technique which uses LSSVM as a base forecaster [66]. They used it along with a MIMO engine to forecast both load and price simultaneously. A quasi-oppositional artificial bee colony (QOABC) was used to optimize the parameters of the SVM based forecast engine. Sayeghi and Ghasemi used a feature selection based on mutual information (MI) and WT [67]. They also developed a new technique called chaos gravitational search algorithm (CGSA) - based on GSA and chaos theory, to optimize the LSSVM forecaster. Another paper to make use of a hybrid methodology based on SVM is [68]. The authors used an SVM engine as the base forecaster, combined with self organizing maps (SOM), which cluster the input data before feeding it to SVM, and PSO, which optimizes the parameters of the SVM forecaster. Zhang and Tan proposed using chaotic least square SVM (CLSSVM) as a part of another hybrid framework [69]. They combined wavelet transform, CLSSVM and exponential generalized auto-regressive conditional hetroskedastic (EGARCH) methods to make a forecasting model. Their tests were performed on the Spanish market data as well as locational marginal price of PJM market (Pennsylvania-New Jersey-Maryland). Yan and Chowdhury created a multiple SVM based mid term electricity price (market clearing price - MCP) forecasting model [63]. Accurate mid term price forecasting is difficult as immediate past prices can not be used and hence the modeling peaks becomes difficult. Other major problems in modeling electricity prices, apart from relation between demand and supply, are factors like business strategy including unethical behavior by competitors. To overcome these challenges, the authors used a multiple SVM model, which has 4 layers - input, SVM data classification, SVM price prediction and output layer. Amjady and Keynia introduced multiple hybrid methodologies out of which one uses an SVM based forecaster [70]. Voronin and Partanen used SVM as a comparison model for their proposed model [71]. Zhang et. al. also create a hybrid model with LSSVM as the base forecaster [72]. They used WT for decomposition of input data, LSSVM to generate forecasts of the decomposed data and a PSO for optimizing the parameters of LSSVM. Yan and Chowdhury also use LSSVM for MCP forecasting [73]. Sarikprueck et. al. use support vector classification to forecast the peak days/hours in the prices and support vector regression for estimating the value of both peak and non peak days [74]. Pinto et. al. used an SVM based forecasting methodology for corroborate the decision making process for biding in electricity markets [75]. Ghasemi et. al. proposed a 3 part forecast engine [76]. First part used flexible wavelet packet transform (FWPT) to decompose original time series into multiple frequencies. Then a conditional mutual information (CMI) based feature selection method was employed. In the second part, a multiple input multiple output (MIMO) based non linear LSSVM (NLSSVM) and ARIMA forecaster was used for modeling linear and nonlinear components. Third part used ABC based TV-SABC to optimize NLSSVM parameters during learning. ABC is modified by upgrading the global search that uses gbest information and by selecting time varying coefficients instead of random coefficients.

Another variation of SVM, which is used in literature is RVM - Relevance vector machines. It is used to get a probabilistic output and uses a Bayesian formulation. [77] and [26] used RVM. [77] used RVM with different kernels to obtain price forecast for the next term and then combine these using multiple regression and GA optimization to obtain a point forecast. [26] used it as one of the models to compare with their proposed methodology. [78] used SVM to forecast the direction of change for next-day electricity prices for German and Austrian EEX price data. Informative Vector machine - another variation of SVM was also used in one paper. [79] used a combination of IVM and local regression for forecasting electricity prices. IVM is another probabilistic alternative to SVM.

2.3 Fuzzy Inference Systems (FIS)

Catalao et. al. proposed a hybrid methodology, which used ANFIS as the base forecaster [80]. They combined WT, ANFIS and PSO. PSO was used to optimize the membership function of ANFIS, whereas WT was used to decompose input data into low and high frequency components. Motamendi et. al. also propose a hybrid forecasting engine based on FIS [81]. In their publication, the initial price and demand forecasts are made by a MIMO engine, to which all available explanatory variables are input. This generates forecasts for a target hour and a period prior to the target hour. Then the variations in the historical demand and prices from observed value is modeled. For this, data association mining techniques are used. The forecasts from step 1 are tuned using info from step 2 with the help of a Fuzzy Inference System. Pousinho et. al. used a combination of PSO and ANFIS [82]. PSO was used to optimize the membership function of ANFIS like in [80]. In ANFIS, the neural networks estimate fuzzy rules from the given data automatically. Catalao et. al. proposed a wavelet neuro fuzzy (WNF) hybrid system [83]. Their methodology is based on an ANFIS forecast model, coupled with wavelet transform to pre-process the input series into different frequency components. The proposed technique is computationally very quick and can be utilized in cases where speed is more important than accuracy. Osório et. al. proposed a combination of ANFIS, evolutionary PSO and wavelet transform (WT) without any exogenous inputs [84].

2.4 Other models

There are various other models which are used in price forecasting literature. The overview of these papers is given in this section.

Gaillard et. al. introduced the winning methodology of team "TOLOLO" for load forecasting and price forecasting tracks of GEFCom2014 competition [85]. They used 3 methods for the price forecasting. First is quantile generalized additive model (quantGAM), second is based on combining individual forecasters including autoregressive (AR), threshold AR (TAR), AR with load forecast (ARX), TARX, linear regression, random forest regressors. Third is a quantile regression model with lasso penalty function. Serinaldi et. al. used generalized additive models [86]. Nowotarski and Weron proposed a quantile regression based technique for modeling prediction intervals for probabilistic forecasting of electricity prices [87]. They used a combination of quantile regression and forecast averaging of various point forecasts. This provided an interval forecast of the price. In probabilistic forecasting, combining probabilistic forecasts does not give the correct probability for the interval [88]. Lei and Feng proposed a grey model called PGM(1,2,a,b) enhanced with parameter optimization for the least square method using PSO [89]. The model performed well for short term price forecasting for different electricity markets. Haldrup et. al. suggested using vector auto regressive (VAR) model for forecasting electricity price in markets where power transmission undergoes occasional periods of congestion [90]. The proposed model has long memory, which helps in more accurate price forecasts. Wu et. al. introduced a recursive dynamic factor analysis (RDFA) algorithm to reduce the complexity of new and useful functional principle component analysis technique (FPCA) [91]. In the proposed algorithm, the principle components (PCs) and PC scores were followed and foretasted recursively using kalman filter (KF). The proposed method showed impressive improvements for forecasting electricity price for Australian and New England markets. Dorini used a stochastic finite impulse response model, who's parameters were estimated by a recursive least square method [92]. Mandal et. al. proposed a hybrid technique based on WT, Firefly algorithm, fuzzy ARTMAP to forecast electricity prices for ontario market [93]. Fuzzy ARTMAP (predictive adaptive resonance theory) is better at preserving old learned concepts in memory than ANNs. Fuzzy ARTMAP is suited for incremental learning. Dong et. al. used a combination of EMD and ARIMA to model electricity prices [94]. According to them, this model performed better than SARIMA as EMD was able to estimate the nonlinear parts of price time series. Nowotarski et. al. proposed that a constrained least square based model is more robust at predicting electricity prices more accurately across different markets [95]. They suggested that ordinary least square and Bayesian model averaging based forecast averaging schemes are not suitable for day ahead electricity price prediction. Alvarez-Ramirez and Escarela-Parez tried to find correlation between different electricity markets [96]. They used detrended fluctuations to temporally quantify the correlations. Jin et. al. proposed SOM Cluster Pattern Sequence-based Next Symbol Prediction (SCPSNSP) algorithm based on pattern sequence based forecaster (PSF) and ANN [97]. Wang and Liu introduced a novel fuzzy forecasting method based on clustering and axiomatic fuzzy set classification [98]. In the first stage, clustering algorithm was used to create different intervals. In second stage, a data set labeled with fuzzy trend was generated with the help of fuzzy logic relations and trends of past data points. This proposed method can predict the future value as well as a fuzzy trend of the data. Karsaz et. al. proposed a methodology to predict both electric load and prices [99]. They use a cooperative co-evolutionary (Co-Co) algorithm. In the past, Co-Co has been used for evolution of ANN as well as solving decomposable problems [100]. In this paper, Co-Co is used to find the relation in input and output data and it determines the order of the system. Maciejowska and Nowotarski explored an extension of [87] for the price forecasting track of GEFCOM2014 competition in [101]. Echo state networks and kalman filter was used by [102]. Juárez et. al. used

Random forests and bagging methods [103]. They made use of lagged variables as well as calendar variables to model the Spanish electricity market data. Voronin and Partanen used a combination of ARMA, GARCH and k-nearest neighbours [71]. Tan et. al. and Liu et. al. used ARIMA and GARCH to model price data for PJM market data [104, 105]. Vilar et. al. used non parametric regression with exogenous variable and is tested on the Spanish market data [106]. Bordignon et. al. proposed various methods for combination of price forecasting models [107].

CHAPTER 3: THESIS STATEMENT

Any forecasting methodology consists of two parts - input features and forecasting technique. In the literature, the most commonly used features used to model electricity prices include historical load, price and load & renewable forecasts. A concept which is very successfully used in electricity load forecasting is "Recency Effect" [108]. The roots of recency effect can be traced from a concept in psychology, which states that most recent events are remembered the best [108]. It states that future load depends upon the conditions (specifically temperature in case of load) existing in the recent past. This effect is added to the model by including the lags (value of the previous n-hours) and moving averages (in order to capture the trend of last n-hours) of the explanatory variable.

This concept is explored extensively for load forecasting by Wang et. al. in [109]. They used a combination of lags and moving averages of temperature to the "Vanilla" benchmark model to improve the forecast accuracy by approx 18. The authors used the lag and moving average of only the temperature variable, and built a different model for lag and moving average pair used. A total of 584 models are built by varying lags from 1-72 hrs, and moving averages from 0-7 days, each model having a unique "day-hour" pair.

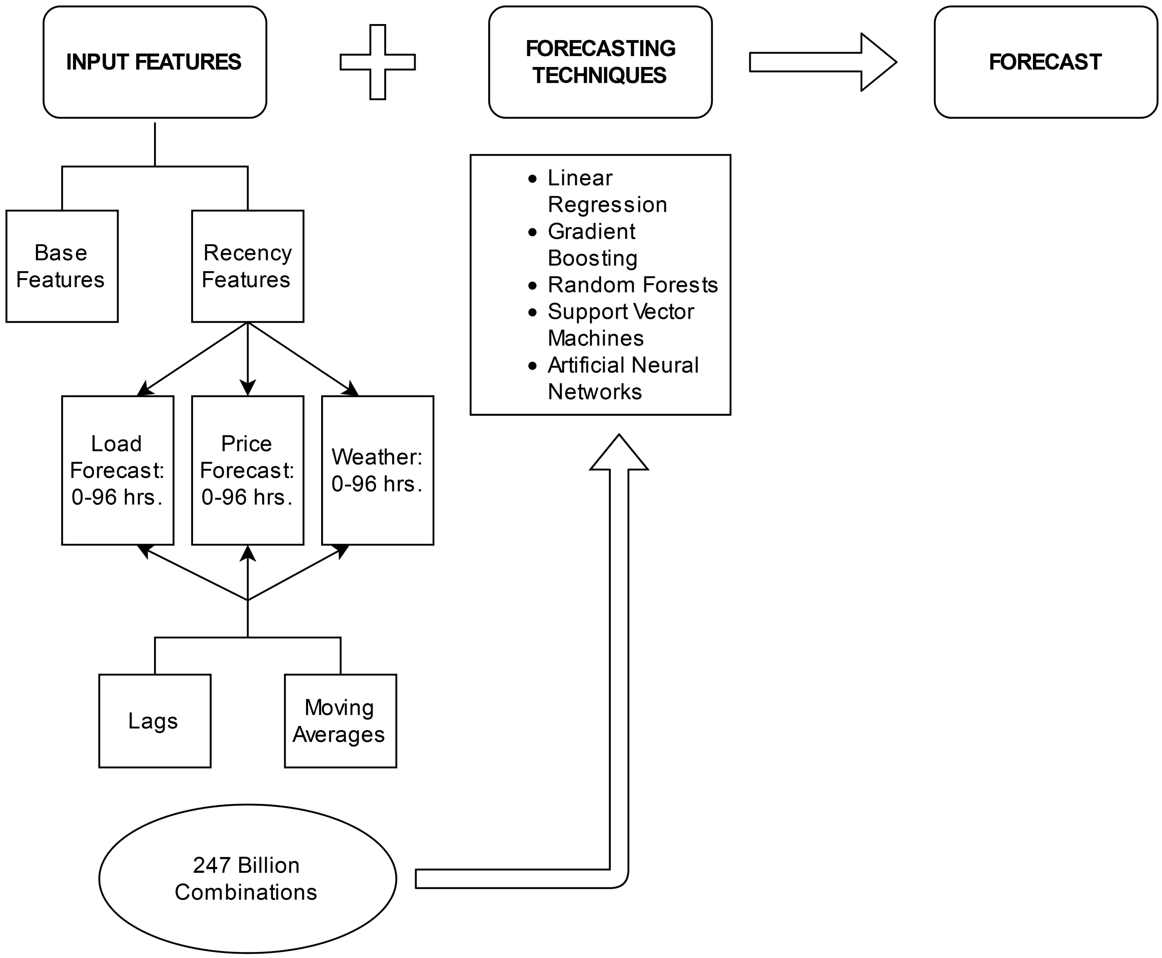

Figure 3.1: Basic structure of any forecasting model, explaining the motivation behind this work

In electricity price forecasting, the use of this effect is very limited. Panapakidis and Dagoumas make use of the lag of price and load in some of the models built in [11] but techniques for comprehensive exploration of the optimal lag and moving average pair is missing in literature.

Also other features can have a "recency effect" on the future electricity price. Hence, the lags and moving averages of the following variables can be used to model electricity prices:

1. Price: The auto-regressive component of the day ahead price is an important feature to include in the models. It accounts for the relationship of the predictor variable with itself at a previous time.

2. Load Forecast: Price is also affected by the load consumption of the previous day. Including lags and moving averages of load forecast accounts for that

3. Weather variables including temperature, wind speed and radiation: Weather is indirectly related to price. It directly affects load consumption, which in-turn affects price. Including lags and moving averages of weather variables also results in improved accuracy.

As we have 3 different sets of variables for which we need to select a pair, our search space increases by the power of 3. If we use 0-96 hrs. of lags and moving averages for these 3 variables, there can be $(96 * 8)^3$ unique combinations. This translates to a very huge search space - approx 247 billion possible combinations. Block diagram explaining this is shown in Figure 3.1.

To find a combination which gives the best (or near best) forecast accuracy, brute force techniques cannot be applied. Hence, using a metaheuristic technique - Particle Swarm Optimization - to find the optimal combination is proposed in this work.

CHAPTER 4: THEORETICAL BACKGROUND

4.1 Artificial Neural Networks

Artificial neural networks try to mimic the working of neurons in a human brain. A neural network consists of densely connected individual elements known as neurons [110]. Each neuron has four basic elements, a set of input, an output, a transfer function and a bias. These neurons can be arranged with different connections to create a highly connected network, which can learn from the input data points. All neural networks try to achieve good performance by a dense connection of simple computational elements [111]. Depending on the number of inputs, a neuron can be either single-input neuron or a multiple input neuron. Figure 4.1 shows architecture of a simple neural network with one hidden layer. In this network, the input layer is made of five input neurons, hidden layer has three neurons and the output layer has one neuron.

A neuron is a basic function of a neural network. It receives inputs, applies weights to these inputs and adds a bias. This is then passed through an activation function to get the output. While training a neural network, these weights and bias are estimated based on the training data. Because of the activation function, neural networks work well in capturing the non linear components. Figure 4.2 shows a basic neuron.

Neural networks can be trained using various techniques. Backward propagation is the most widely used algorithm for training. For the purpose of this research, conjugate gradient method is used for training.

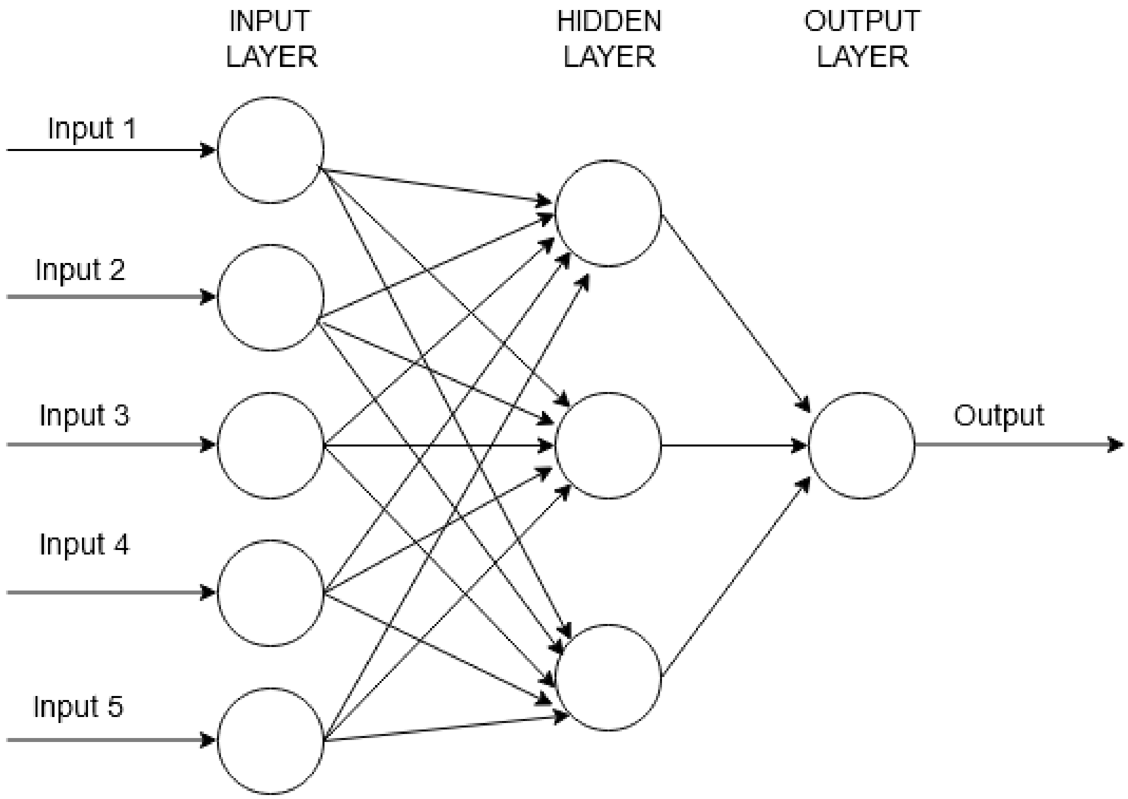

Figure 4.1: Architecture of neural network with 1 hidden layer, 1 input layer and 1 output layer

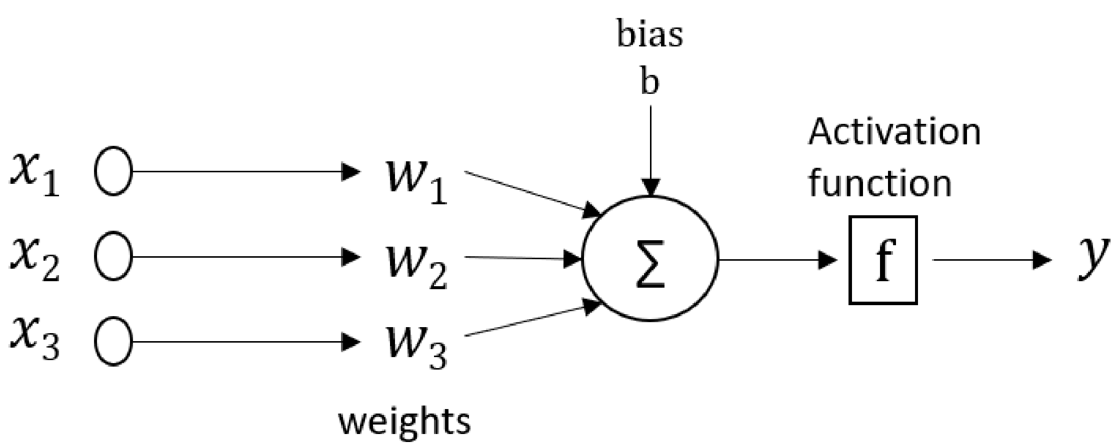

Figure 4.2: Basic structure of a neuron in a neural network

4.2 Linear Regression

Regression analysis is a statistical technique used for modeling relationship between a set of variables [112]. Linear regression tries to fit a line to a set of data points in order to draw a relation between these variables. A multiple linear regression model with n regressors can be defined as the following:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_n X_n + \epsilon \tag{4.1}$$

where β_0, ..., β_n are the parameters to be estimated, X_1, ..., X_n are the known dependent variables, ϵ is the error term, which is normally distributed random variable and Y is the dependent variable, which is being modeled [108]. A multiple linear regression model can be extended to a polynomial regression model by including the polynomial terms of the independent variables. For example, a cubic polynomial could be added

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon \tag{4.2}$$

This is equivalent to

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{4.3}$$

where $X_1 = X$, $X_2 = X^2$, $X_3 = X^3$. Linear regression has an underlying assumption that the system is well defined without any vagueness [108]. Similarly, interaction effects between different variables can also be added [112]. For example,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon \tag{4.4}$$

This is equivalent to

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{4.5}$$

where $X_3 = X_1 X_2$. Linear regression models are not limited to just quantitative variables. Qualitative variables can be included in the model [108]. these variables are also known as categorical or class variables and can be included by assigning 0 or 1 to the based on the category of the data. For example, day of the week can be added as seven new independent variables, each indicating information about the day. for Monday, all the values where the day is Monday will be 1 whereas if it is not a Monday, it will be 0.

$$X_1 = \begin{cases} 1, & \textit{if the day is Monday} \\ 0, & \textit{otherwise} \end{cases} \tag{4.6}$$

A point to be noted is that any regression model is linear in the parameters (the $\beta' s$) and not in the independent variables (the $X' s$) [112].

4.3 Support Vector Machines

Support Vector Machines (SVM) were introduced by Vapnik, 1995. They are a tool for multidimensional function estimation [113]. In support vector regression, we try to estimate a function which has a maximum of ε deviation from the actual values of the target variables [114]. This means, no errors larger than ε will be tolerated. In two dimensions, we can describe support vector regression as follows: for a given set of points, $\{(x_1, y_1), ..., (x_n, y_n)\} \subset \chi \times R$, where χ is the space of input patterns [114], the function being estimated can be described as,

$$f(x) = w \cdot x + b \tag{4.7}$$

where $w \in \chi, b \in R, w \cdot x$ is the dot product in χ. To fit the data to a support vector machine regressor, we try to find the smallest w such that the error between the observed target variable and the predicted function is less than ε. This can be written as the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|w\|^2 \\ &\text{such that} \begin{cases} y_i - w \cdot x_i - b \leq \varepsilon \\ w \cdot x_i + b - y_i \leq \varepsilon \end{cases} \end{aligned} \tag{4.8}$$

This formulation for support vector machines works only if the optimization is feasible. That is, $f(x)$ actually exists. This is called a "Hard-Margin" SVM. This can only be possible when the training data is linearly separable. To overcome this, a "Soft-Margin" problem can be formulated, by adding slack variable (ξ, ξ^*) to make the optimization problem feasible [115].

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$$
$$\text{subject to } \begin{cases} y_i - w \cdot x_i - b \leq \varepsilon + \xi_i \\ w \cdot x_i + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{4.9}$$

There is a trade-off between the flatness of function $f(x)$ and the amount of deviations larger than ε that can be accepted. This can be adjusted by changing the value of C [114].

The above explained formulations can be extended to multiple dimensions. Also, to capture non linear relations, a simple pre-processing to input feature space can be performed. This is done by mapping a certain feature to a nonlinear function, usually known as a kernel. The most commonly used kernels are linear, polynomial, radial bias function, and sigmoid kernels. For the purpose of this work, SVM is implemented using the scikit_learn's svm library.

4.4 Random Forests

Random forests are essentially a combination of two techniques - Classification and Regression Trees (CART) and bagging [116]. CART is a tree based algorithm, which maps observations of a variable to conclusions, introduced by L. Breman in 1984. Figure 4.3 shows a simple classification and regression tree. CART algorithm is a growing procedure as can be seen in figure 4.3. In each iteration, based on the best split variable and value, each node is divided into two sub-nodes until the best split is found. This means it fits the data pretty well, although the prediction accuracy may not be as good due to high variance [116]. To overcome these limitations, random forests include bagging to the base CART algorithm.

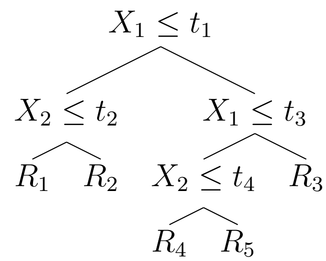

Figure 4.3: Classification and Regression Tree - CART

Algorithm 1: Random Forest

Result: Random forest tree and corresponding prediction

```
Select training data with p variables;
B= number of trees;
m = number of split variables at each split;
n_min = minimum node size;
for number of trees, B do
    Draw a sample from the training data;
    for node size = n_min do
        select m random variables out of selected p variables;
        select best spit variable out of the m variables;
        split node into 2 sub-nodes;
    end
    return ensemble of trees;
end
get prediction by the following:
Ŷ_rf^B(x) = 1/B Σ_{b=1}^{B} Ŷ_b(x);
where Ŷ_b(x) is the prediction of b_th tree;
```

$$\hat{Y}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{Y}_b(x);$$

4.5 Gradient Boosting

In boosting, multiple weak learners are combined by an iterative process in order to generate a strong learner. Gradient boosting machines (GBM) were introduced by Friedman in [117]. They are essentially an optimization problem, with an objective of finding the best additive model, which minimizes the error. It works by adding a new decision tree to the existing model, which reduces the error between the predicted data and actual data (a loss function). An essential part of the algorithm is a parameter called learning rate (α, where $0 \leq \alpha \geq 1$). At each iteration, the added decision tree's weightage to the final result reduces by this factor. With more number of small steps, the accuracy increase as compared to small number of large steps [118]. The number of subsamples selected for each iteration also has an effect on accuracy of the GBM model. Below is a simplified algorithm for GBMs.

Algorithm 2: Random Forest

Result: Gradient Boosting Machine

d = depth of the decision tree;

K = number of iterations;

α = learning rate;

η = subsample fraction;

residual $r_o = y$;

$\hat{f} = 0$;

for $k = 1, 2, ...K$ **do**

  Draw a random sample from the training data based on η;

  fit a decision tree $\hat{f}^k$ of d depth to residual r_{k-1};

  $\hat{f}(x) \leftarrow \hat{f}(x) + \alpha \hat{f}^k(x)$;

  $r_k \leftarrow r_{k-1} - \alpha \hat{f}^k(x)$;

end

4.6 Particle Swarm Optimization

Particle swarm optimization (PSO) was introduced by Kennedy and Eberhart in [119] as an evolutionary algorithm for finding near best solution to a given problem. In PSO, a number of particles are used in a search space, to find an optimum value of a given function or a problem. The movement of each particle in the search space is dependent upon its own history as well as the position of the best particle (particle with least error in the group or swarm). As the number of iterations increase, this "swarm" of particles moves, finding for the best solution in the search space of the function, just like a flock of birds finds food [120]. Very nominal computational resources are required to implement PSO. The PSO algorithm is described below based on [119].

Algorithm 3: Particle Swarm Optimization

Result: Near best solution in a given search space

initialize the number of particles, initial position of each particle and initial velocities of each particle;
initialize number of maximum iterations;
while *((current iteration < max number of iterations) or*
(gbest equals required fitness value)) **do**
    for *each particle* **do**
        calculate fitness value;
        if *current fitness value is better than particle's best fitness, pbest* **then**
            pbest = current fitness value;
            ploc = current location, x_i;
        end
        if *fitness value is better than best fitness of the population, gbest* **then**
            gbest = current fitness value;
            gloc = current location, x_i;
        end
        change velocity an position of particle based on:
        $v_i = v_i + rand() * (ploc - x_i) + rand() * (gloc - x_i)$
        $x_i = x_i + v_i$
    end
end

4.7 Forecast Evaluation

Evaluation of any forecast is done on the basis of how a forecasted value ($\hat{Y}_h$) compares with the actual observed value (Y_h). In literature, then most widely used metrics for Electricity Price forecasting is *absolute error*,

$$AE_h = |Y_h - \hat{Y}_h| \tag{4.10}$$

As forecasts are usually made for more than a single time step ahead in the future, mean of the absolute error is used ($MAE = \frac{\sum_{h=1}^{H} AE_h}{H}$). Absolute error metrics are usually hard to compare across different data sets, hence some forecasters prefer to use absolute percentage error [4],

$$APE_h = AE_h / Y_h \tag{4.11}$$

Another widely used metric is root mean squared error (RMSE) and noramalized RMSE (nRMSE).

$$RMSE_T = \sqrt{\frac{1}{T}\sum_{h=1}^{T}(Y_h - \hat{Y}_h)^2} \tag{4.12}$$

$$nRMSE_T = \frac{RMSE_T}{\bar{Y}_h} \tag{4.13}$$

Another very popular metric for forecasting, including electricity load forecasting is mean absolute percentage error (MAPE), which is the mean of all absolute percentage errors from 4.11. This is usually not used for electricity price forecasting as price can drop to near zero and in some markets go negative as well, which blows up the MAPE value (in case on 0 actual price values, MAPE becomes infinite). Hence it is not prefered in this field [4].

CHAPTER 5: DATA DESCRIPTION

The case study is performed on the data from the German Austrian markets. As a day ahead prediction is made, hourly spot prices are collected from the EPEX spot market data. Correspondingly, all other data collected is hourly data where ever available. If hourly data is not available, the frequency is changed for that data. The spot prices are dependent upon different factors. These include load forecast, renewables generation forecast (which includes on-shore and off-shore wind generation forecast and solar generation forecast), installed capacities (hydro, conventional sources, biomass, pumped hydro storage, solar, wind, uranium), import balances, price emission allowance. The weather data is also used as an input variable. The weather variable include wind-speed at 10m, temperature, direct horizontal radiation and diffuse horizontal radiation. As NWP data was not easily available, the actual data is used and only (min) 24 lags are used. This corresponds to a 24 hour ahead naive forecast of weather variables. The sources and available resolution of these variables are shown in table 5.1.

Table 5.1: Data Sources

Data	**Source**	**Resolution**
Load Forecast	ENTSO-E	15 min, day ahead
Commodity Prices	EEX/EIA	1 day
Renewable Forecast	ENTSO-E (using data from 50 Hertz, Amprion, Tennet, TransnetBW)	15 min – day ahead
Price	EPEX-spot	1 hr, day ahead
Installed capacities	50 Hertz, Amprion, Tennet, TransnetBW	1 hr
Emissions allowance price	EEX	1 day
Weather	OPSD data	1 hr

The following is the description of the available explanatory variables:

- **Load forecast variables**: This is the day ahead load forecast for the different regions of Germany. This is available from the different transmission system operators (TSOs) mentioned in Table 5.1. The available resolution of the data is 15 minutes. This data is combined for the different TSOs and re-sampled to an hourly frequency to match frequency of the price data.

- **Renewable forecast variables**: This includes solar power forecast, wind on-shore renewable production forecast, wind off-shore renewable production forecast.

- **Power generation installed capacities**: Brown coal installed capacity,hard coal installed capacity, oil installed capacity, uranium installed capacity, hydro installed capacity, biomass installed capacity, wind installed capacity, solar installed capacity, gas installed capacity, pumped storage installed capacity, seasonal storage installed capacity, other sources installed capacity, import balance capacity. All these are available as daily data. It is re-sampled to hourly data in order to make it compatible with the other variables

- **Emission allowance price**: This corresponds to the carbon tax applied on the generators in Europe.

- **Weather variables**: Temperature, wind-speed at 10 m, direct horizontal radiation and diffuse horizontal radiation.

For having an initial look into the data, the mean, standard deviation, minimum value, maximum value and 25%, 50% and 75% percentile data are measured for each variable. The available data can be broadly classified into 3 categories - forecast variables, installed capacities for conventional sources, installed capacities for non-conventional and other sources. These statistics are presented in Table 5.2 - 5.4

Table 5.2: Basic statistics for price and forecast variables

	ELSPOT PRICE DAY AHEAD (EUR/MWh)	LOAD FORECAST (MW)	SOLAR RENEWABLE FORECAST (MW)	OFFSHORE RENEWABLE FORECAST (MW)	ONSHORE RENEWABLE FORECAST (MW)	SUM RENEWABLE FORECAST (MW)
Mean	31.49	62344.68	3990.31	1307.54	8853.23	14151.07
Std	14.06	10789.74	6160.65	1014.27	6938.07	8904.75
Min	-130.09	33950.00	0.00	4.00	215.50	657.25
25%	23.95	53378.00	0.00	426.00	3698.88	6875.50
50%	30.26	62361.00	127.75	1044.63	6795.13	12396.25
75%	38.13	71664.38	6256.31	2125.00	11809.75	19887.94
Max	163.52	85599.50	27557.50	4124.50	37299.75	47702.75

Table 5.3: Basic statistics for installed capacities - conventional sources

	IMPORT BALANCE INSTALLED CAPACITY (GW)	BIOMASS INSTALLED CAPACITY (GW)	URANIUM INSTALLED CAPACITY (GW)	BROWN COAL INSTALLED CAPACITY (GW)	HARD COAL INSTALLED CAPACITY (GW)	OIL INSTALLED CAPACITY (GW)	GAS INSTALLED CAPACITY (GW)
Mean	-5.59	5.43	9.17	15.69	11.82	0.23	4.73
Std	3.35	0.62	1.71	2.34	5.67	0.11	2.91
Min	-15.85	4.33	0.00	0.00	0.00	0.00	0.00
25%	-8.01	4.75	7.79	14.43	6.84	0.19	2.45
50%	-5.91	5.57	9.75	15.93	11.87	0.21	4.12
75%	-3.46	6.06	10.48	17.39	16.69	0.22	6.18
Max	7.35	6.25	12.08	19.91	23.34	1.41	18.07

Table 5.4: Basic statistics for Installed capacities - hydro, renewables and others

	HYDRO POWER INSTALLED CAPACITY (GW)	PUMPED STORAGE INSTALLED CAPACITY (GW)	SEASONAL STORAGE INSTALLED CAPACITY (GW)	WIND INSTALLED CAPACITY (GW)	SOLAR INSTALLED CAPACITY (GW)	OTHERS INSTALLED CAPACITY (GW)
Mean	2.36	1.06	0.17	9.48	4.24	0.28
Std	0.60	0.95	0.15	7.46	6.58	0.23
Min	0.00	0.00	0.00	0.13	0.00	-0.49
25%	1.91	0.39	0.04	3.77	0.00	0.16
50%	2.33	0.71	0.11	7.38	0.10	0.26
75%	2.85	1.44	0.26	13.29	6.65	0.39
Max	3.95	6.57	0.82	39.28	29.14	1.31

Load forecast is one of the most important variable which affects the day ahead prices. Figure 5.1 shows a snapshot of the load forecast, day ahead prices and the total renewable forecast for the German-Austrian market.

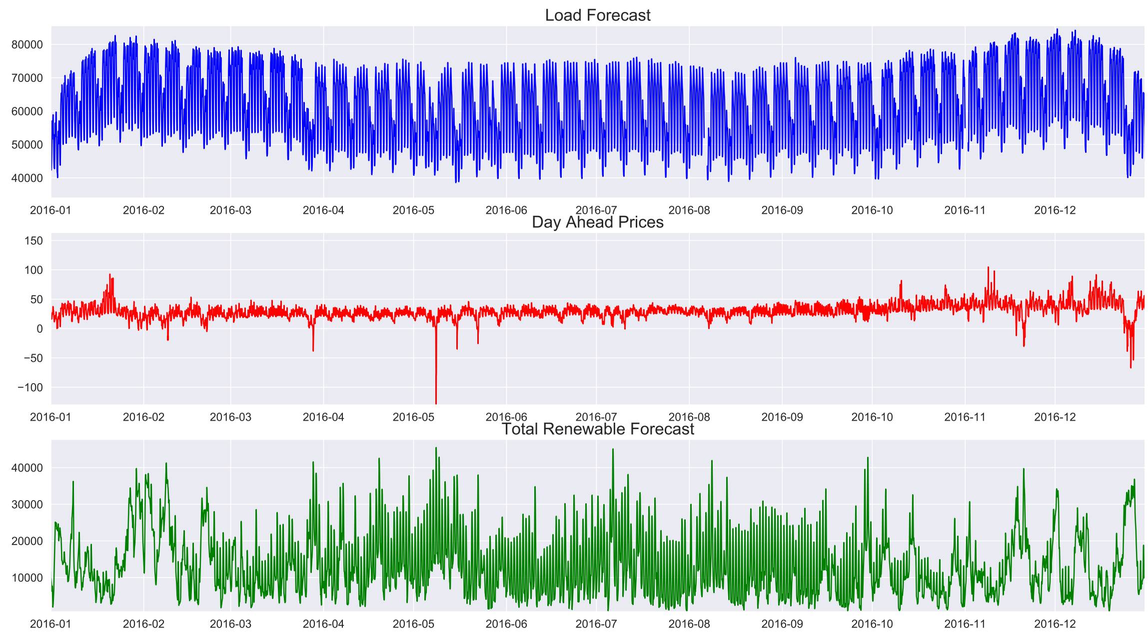

Figure 5.1: Load forecast (in MW), price (Eur/MWh) and renewable forecast (MW) for the training period

For these variables, to see the existing patterns between these three variables, a scatter matrix is formed. This gives a comprehensive view into the relationship between the day ahead price and load forecast, renewable forecast. This view can also give insights into the type of distribution of each variable (shown by the diagonal axis in the scatter matrix plot, i.e. price vs price plot, load forecast vs load forecast plot and renewable forecast vs renewable forecast plot. Figure 5.2 visualizes this. "elspot_price_day_ahead" is the price variable, "load_fc" is the load forecast variable and "sum_de_at_renewable_fc" is the total renewable forecast of the German-Austrian market. It can be observed from this figure that as the load demand increases, the price also increases and as the renewable production increases, price decreases as a general trend.

To get more interesting insights into the data, box-whisker plots are drawn. Box-whisker plots are non-parametric and are a good way to visualized data without making any assumptions about the distribution of the data set. Each box-whisker plot shows the following elements:

- Minimum
- Maximum

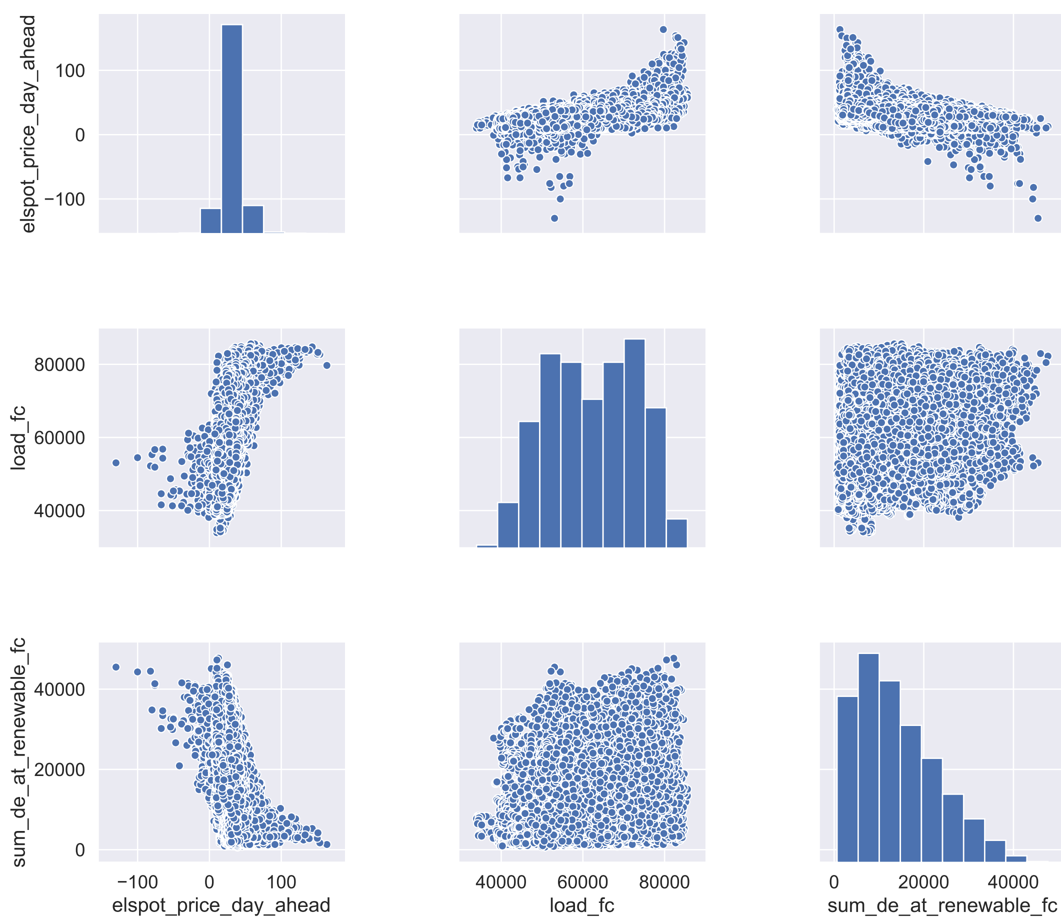

Figure 5.2: Pairwise scatter plot of day ahead price, load forecast and renewable forecast

- Median or 50th percentile
- First quartile or 25th percentile

- Third quartile or 75th percentile
- Interquartile ranage or IQR

Box plots are generated for the day ahead price variable. This is aggregated into different groups based on month of year, hour of day and day of week in order to see if any pattern exists between the price and these calendar variables. Figure 5.3 shows these plots.

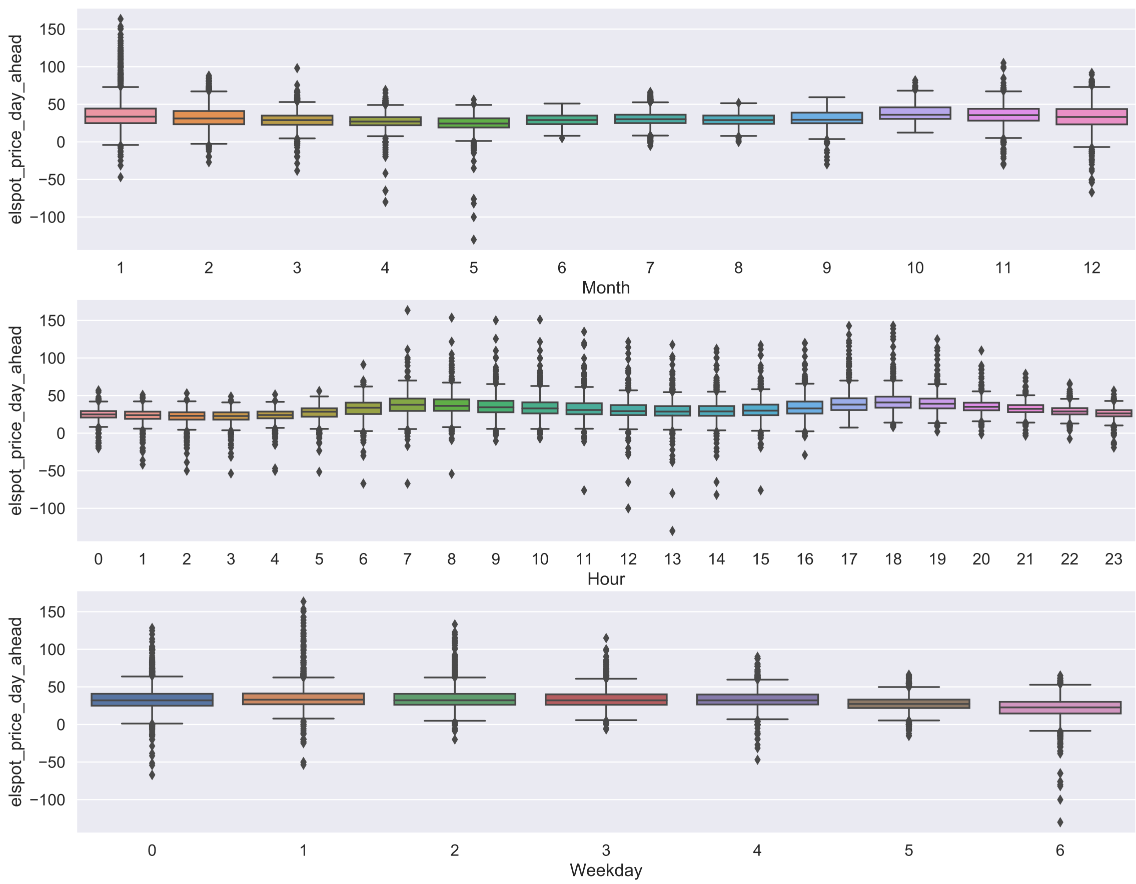

Figure 5.3: Box plots of target day ahead price with calendar variables - month, weekday and hour (elspot_price_day_ahead is the price variable)

Another way in which the correlations between different variables can be visualized by generating a correlation matrix. The pearson correlation ratio is calculated for each pair of variables. This is then plotted as a heatmap in figure 5.4.

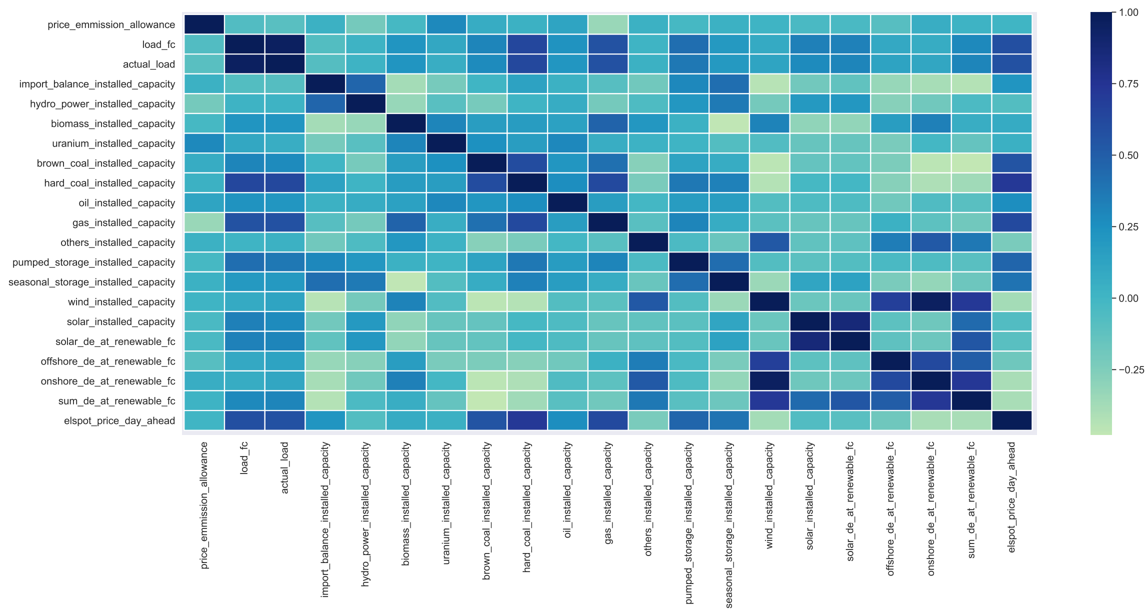

Figure 5.4: Correlation matrix of all the variables

CHAPTER 6: METHODOLOGY AND RESULTS

Short term electricity price forecasting is of prime importance for daily operations of the power markets. Various modeling methods, including multi-agent models(for qualitative forecasts), fundamental models (based on physical and economic relations), reduced form models (for modeling main characteristics of price instead of accurate hourly forecast), statistical models (regression, time series techniques like ARIMA, ARMAX, GARCH, exponential smoothing) and computational intelligence models (ANN, SVM, etc.) are used for this purpose.

This chapter explores the statistical models and computational intelligence models and a unique combination of these. An evolutionary algorithm is used for optimizing the input features for the various models. The first section gives the overview of hardware and software used for implementation of these techniques. Second section gives the methods used for building the base models with their respective results. Third section provides a peak into how "recency effect" affects forecast accuracy. The fourth and final section explains the implementation of particle swarm optimization for selection of optimal combinations of recency variables.

6.1 Hardware and Software Description

6.1.1 Hardware

The system used for implementation of all the models consisted of a 12 core Intel(R) Core(TM) i7-6850K CPU with 46 GB RAM. The system also had 2 NVIDIA TITAN V GPUs with CUDA Version 10.2 installed. Operating system installed on the system was Ubuntu 18.04.4 LTS (Linux kernel 5.3).

6.1.2 Software

Only open-source software was used for building the models. Python 3.6.9 was the main language used with Spyder 4.1 IDE for code development. The following python packages were used:

- **numpy**: a scientific computing package in python [121]. This was used for mathematical and matrix operations. Also, this is the backbone of machine learning library sci-kit learn.

- **Pandas**: A basic data analysis and manipulation tool for python [122]. It was used for building the input data set before forecast modeling

- **Sci-Kit Learn**: A machine learning library with efficient tools for predictive modeling [123]. It was used for building linear regression, random forests, support vector machine models.

- **XGBoost**: An optimized distributed gradient boosting library [124]. The python API from XGBoost was used for building gradient boosting models.

- **neupy**: A python library for building artificial neural networks with Tensorflow backend [125]. The conjugate gradient algorithm implementation was used for training the ANN models built for this work.

- **PySwarms**: A python toolkit for implementing swarm intelligence and evolutionary algorithms [126].

6.2 Base Models

6.2.1 Model Description

The investigated base models include linear regression, gradient boosting, random forests, support vector regression, artificial neural networks using adam optimizer for learning and artificial neural network using conjugate gradient learning. These models are selected from the literature. The most successfully implemented models are used for this work.

These base models have various sets of parameters which need to be tuned. Each parameter is selected for the current data set using a grid search to find the best set of parameters. Given below are the models used along with the set of parameters used for each model.

1. **Linear Regression (LR)**: Ordinary least square method is used for parameter estimation of the linear regression model. The model is built using Python's SciKit Learn package [123].

2. **Gradient Boosting (GB)**: For building the gradient boosting model, various parameters need to be considered. The loss function to be minimized is selected to be least squares, the number of estimators are 100, learning rate is 0.1 and the subsample size is chosen to be 1. For building this model, XGBoost library is used [127].

3. **Random Forest (RF)**: Random forest model is built using Python's SciKit Learn package. 100 estimators are used. Mean squared error is used to measure the accuracy of each split (as explained in Section 4.4). Each tree is expanded until all sub-nodes are pure or all sub-nodes have less than 2 samples.

4. **Support Vector Machines**: Python's scikit learn implementation is used to build svm model [123]. It internally uses the famous libsvm library [128]. To capture non-linear relationships, a radial bias function kernel is used on input data as a pre-processing step. The epsilon value,for which there is no penalty in training loss is 0.1. The regularization parameter C is 1 and the tolerance for stopping is 10^{-3}.

5. **Artificial Neural Network (ANN)**: ANN model is built with learning done using conjugate gradient. NeuPy library in python is used for using Conjugate

Gradient algorithm [125]. The architecture of the model was selected by hit and trial method. Two hidden layers are used. the first hidden layer contains 100 neurons with sigmoid activation function. The second hidden layer is made up of 150 neurons with ReLU activation function. The number of epochs for training are 250 and the loss function minimised for training is root mean squared error.

6.2.2 Feature Engineering

Chapter 5 gives a comprehensive description of all the available variables. Apart from the available variables, some new features are created. These new features include square and cube of the load forecast variable, calendar variables like weekday, month, hour and holiday indicators. The following are all the variables used for the base models:

- **Load forecast variables**: Hourly load forecast, square of hourly load forecast, cube of hourly load forecast
- **Renewable forecast variables**: Solar power forecast, wind onshore renewable production forecast, wind off-shore renewable production forecast
- **Power generation installed capacities**: Brown coal installed capacity,hard coal installed capacity, oil installed capacity, uranium installed capacity, hydro installed capacity, biomass installed capacity, wind installed capacity, solar installed capacity, gas installed capacity, pumped storage installed capacity, seasonal storage installed capacity, other sources installed capacity, import balance capacity. All these are available as daily data. It is re-sampled to hourly data in order to feed to the base models
- **Emission allowance price**
- **Calendar variables**: Weekday indicator, month indicator, hour indicator, holiday indicator for each state of Germany

- **Weather variables**: Temperature, wind-speed at 10 m, direct horizontal radiation and diffuse horizontal radiation. As numerical weather prediction data is not readily available, history of the weather is used with a 24hr lag to replicate a weather prediction. That is, previous day's actual weather is considered to be weather prediction for the day to be predicted.

Table 6.1: RMSE, nRMSE and MAE values for base model with weather data included (24hr lags for weather data to represent weather forecasts)

Error/Model	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE	8.34	7.29	8.28	9.79	**6.86**
nRMSE	0.288	0.251	0.286	0.338	**0.237**
MAE	6.47	5.53	6.24	7.56	**4.96**

6.2.3 Results

Table 6.1 shows the RMSE, nRMSE and MAE values for the six base models. For this set of features, ANN with conjugate gradient performs the best with least error when compared to other techniques. A season-wise analysis of error is done for the base models. The error is recorded in tables 6.2 - 6.5. It shows that except for winter, ANN has the least error. For winter season, LR model has the least error with only a very small improvement in nRMSE value.

Table 6.2: Error values for winter season (December, January and February)

Error/Model	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE	**8.78**	9.04	9.28	11.90	8.84
nRMSE	**0.296**	0.305	0.313	0.401	0.298
MAE	6.76	7.07	7.27	9.62	**6.71**

Table 6.3: Error values for spring season (March, April, May)

Error/Models	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE	9.53	8.14	10.96	11.77	**7.47**
nRMSE	0.403	0.344	0.463	0.497	**0.315**
MAE	6.80	6.19	8.82	9.18	**5.08**

Table 6.4: Error values for summer season (June, July, August)

Error/Models	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE	8.33	5.11	5.96	6.91	**4.94**
nRMSE	0.304	0.187	0.218	0.253	**0.181**
MAE	7.28	3.95	4.38	5.59	**4.08**

Table 6.5: Error values for autumn season (September, October, November)

Error/Models	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE	6.38	6.20	5.72	7.50	**5.46**
nRMSE	0.181	0.176	0.162	0.213	**0.155**
MAE	5.01	4.92	4.48	5.84	**3.99**

The performance of these models is visualized in Figure 6.1 and Figure 6.2. A snapshot (10th May 2016 to 14th May 2016) of the actual price value and forecasted values are shown. It is observed that the ANN with conjugate gradient learning and linear regression work well for that day as well as for the whole year. This corresponds to the error values shown in the tables above.

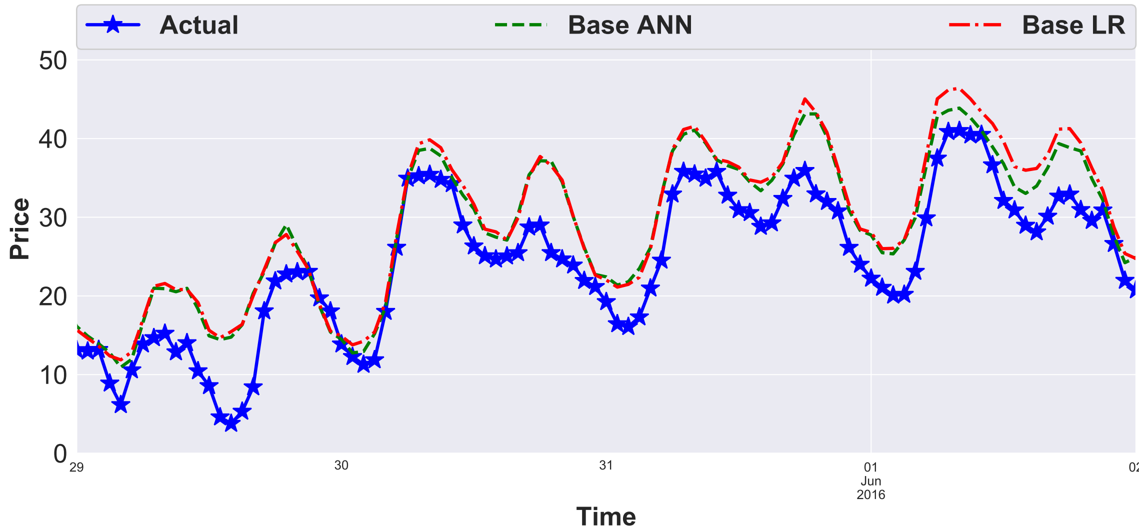

Figure 6.1: LR and ANN forecast vs actuals

6.3 Adding Recency Effect

In load forecasting, recency effect is used very effectively. Wang et. al. use a combination of lags and moving averages of temperature to the "Vanilla" benchmark model to improve the forecast accuracy by approx 18% [109]. The roots of recency effect can be traced from a concept in psychology, which states that most recent events are remembered the best [108]. In [109], the authors use the lag and moving average of only the temperature variable, and build a different model for lag and moving average pair used. A total of 584 models are built by varying lags from 1-72 hrs, and moving averages from 0-7 days, each model having a unique "d-h" pair.

This concept can be extended to the price forecasting realm and is not yet explored in literature for electricity price forecasting. Panapakidis and Dagoumas make use of the lag of price and load in some of the models built in [11] but techniques for comprehensive exploration of the optimal lag and moving average pair is missing in literature.

Another set of lags and moving averages can be added for the weather variables and the load forecast variable.

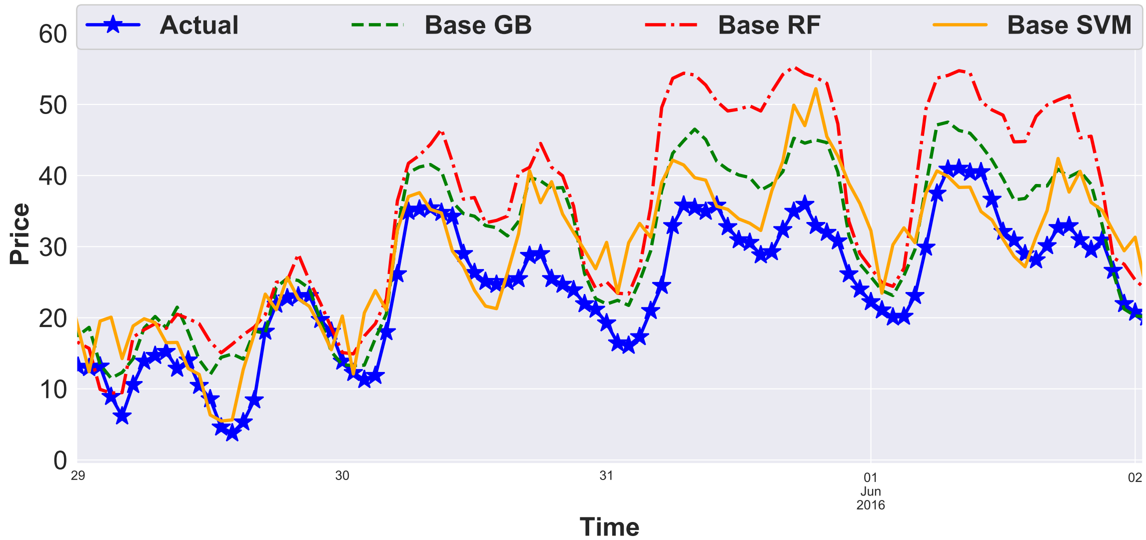

Figure 6.2: GB, RF and SVM forecast vs actuals

1. Price: The auto-regressive component of the day ahead price is an important feature to include in the models. It accounts for the relationship of the predictor variable with itself at a previous time.

2. Load Forecast: Price is also affected by the load consumption of the previous day. Including lags and moving averages of load forecast accounts for that

3. Weather variables including temperature, wind speed and radiation: Weather is indirectly related to price. It directly affects load consumption, which inturn affects price. Including lags and moving averages of weather variables also results in improved accuracy.

In [109], the search space obtained is a modest one as only pair of lags and moving averages is selected for a single variable. They build a linear regression model for each "d-h" pair by varying lags from 1-72 and moving averages from 0-7. This means 584 (73*8) linear regression models are built. This task can be done relatively quickly with modern computers.

In our case, as we have 3 different sets of variables for which we need to select a pair, our search space increases by the power of 3. That is, there can be $(73 * 8)^3$

unique combinations of lag and moving average values for the 3 sets of variables. Also, if we decide to use moving average with an hourly increment instead of daily increment, we further increase the search space. This calls for implementation of new techniques, other than brute force, to find the optimum or the near optimum value.

6.3.1 Effect of Recency on Electricity Price Forecasting

To see the effect of adding recency effect to EPF, lag and moving average of the 3 variables mentioned above (price, load forecast and weather variables) are added. 24 hr lag values are used for all three variables and 2 day (or 48 hr) moving averages are added to the base models.

Table 6.6: Error with recency effect - initial look with 24hr lags and 48hr moving averages for price, load forecast and weather variables

Error/Model	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE	6.83	**6.74**	7.23	10.01	6.88
nRMSE	0.24	**0.23**	0.25	0.35	0.24
MAE	5.06	**4.99**	5.31	7.81	4.95

There is a significant improvement in the accuracy of the base models. Table 6.6 shows the error values for each model after adding 24 hr lag and 48 hr moving average of the mentioned variables. In this case, gradient boosting model performs the best. Even though, linear regression shows the most improvement form the base model. Table 6.7 shows the percentage change in the RMSE/nRMSE and MAE values. The negative value indicates a reduction of error, while positive value shows an increase in error.

Table 6.7: Percentage change in error from base models after adding recency effect

Model -> Error % Change	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE/nRMSE	**-18.11%**	-7.46%	-12.77%	2.14%	0.28%
MAE	**-21.75%**	-9.82%	-14.87%	3.40%	-0.33%

Figure 6.3 - Figure 6.7 show the comparison of the base models with recency models. in each of the models, inclusion of recency effect is able to reduce the over forecast that was done by the base models. The greatest reduction is seen in LR, which is also evident from the percentage change in error. The figures also show how much each model has improved from the base model.

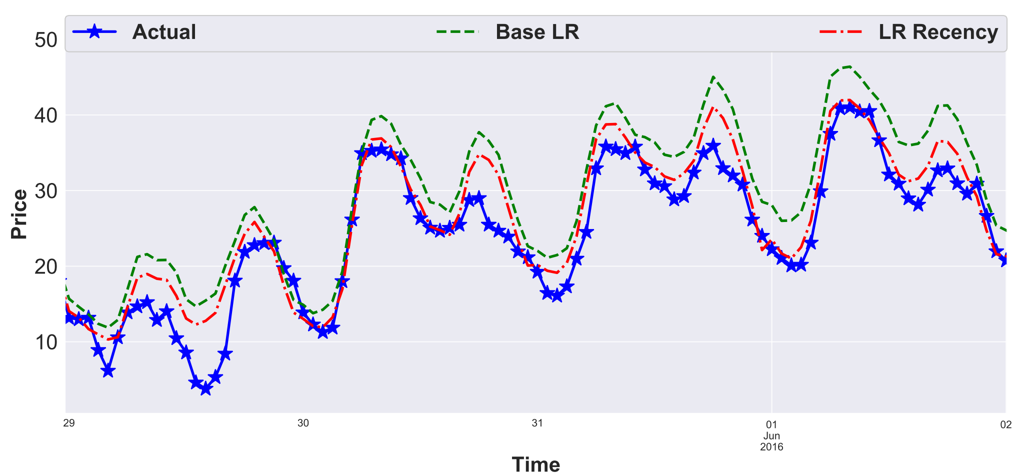

Figure 6.3: Comparison of Base LR model and LR with Recency

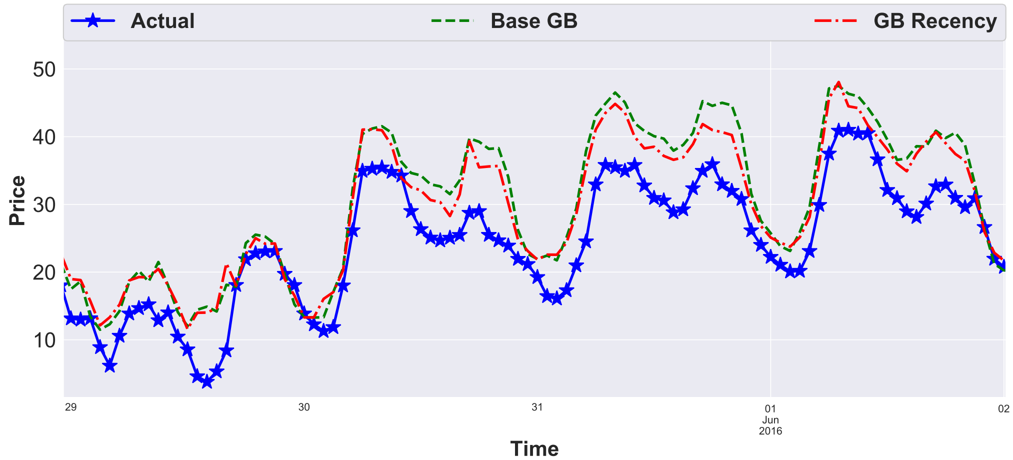

Figure 6.4: Comparison of Base GB model and GB with Recency

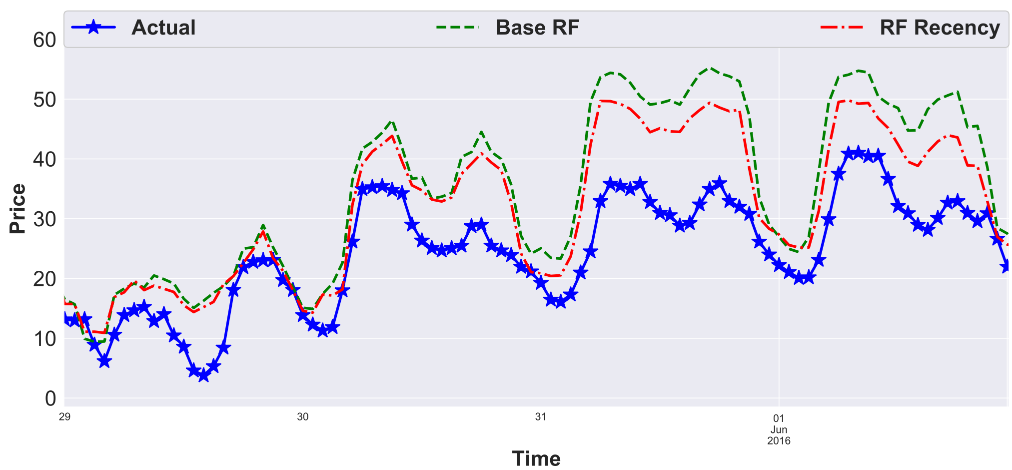

Figure 6.5: Comparison of Base RF model and RF with Recency

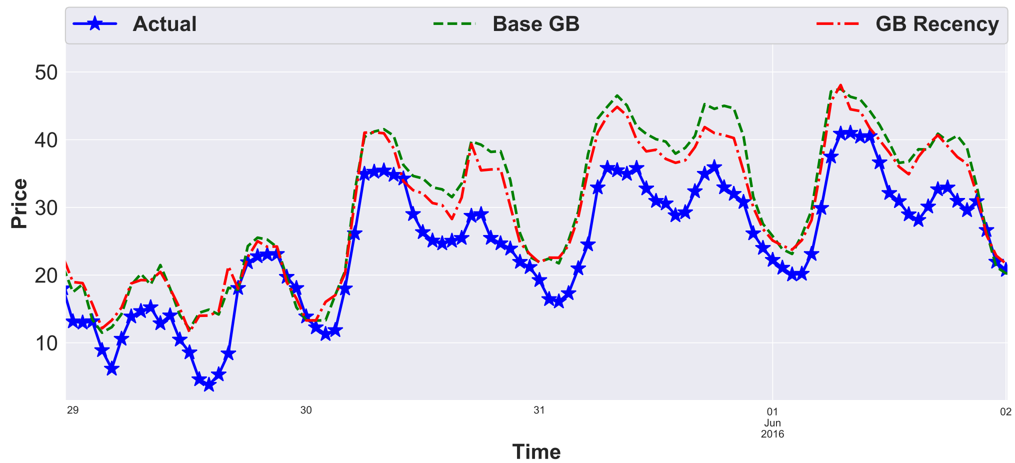

Figure 6.6: Comparison of Base RF model and RF with Recency

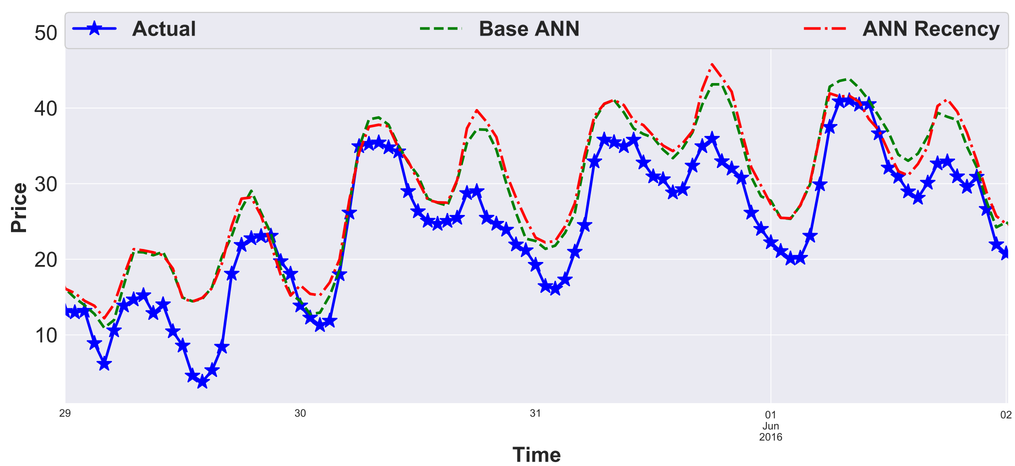

Figure 6.7: Comparison of Base ANN model and ANN with Recency

6.4 Optimal Feature Selection using Particle Swarm Optimization

6.4.1 Experimental Setup

Given the huge search space to find optimal pairs of lags and moving averages of 3 variables, a brute force methodology can not be applied because of computational limitations. To resolve this, meta-heuristic approach was followed.

Meta-heuristics are a class of procedures usually used to find a good enough solution to a problem. Unlike optimization algorithms or iterative (brute force) procedures, Meta heuristic algorithms do not guarantee a global optimal solution. Instead, they find good solutions with nominal computational power. Some examples of meta-heuristic algorithms include genetic algorithm, particle swarm optimization, ant colony optimization, chemical reaction optimization, simulated annealing, etc.

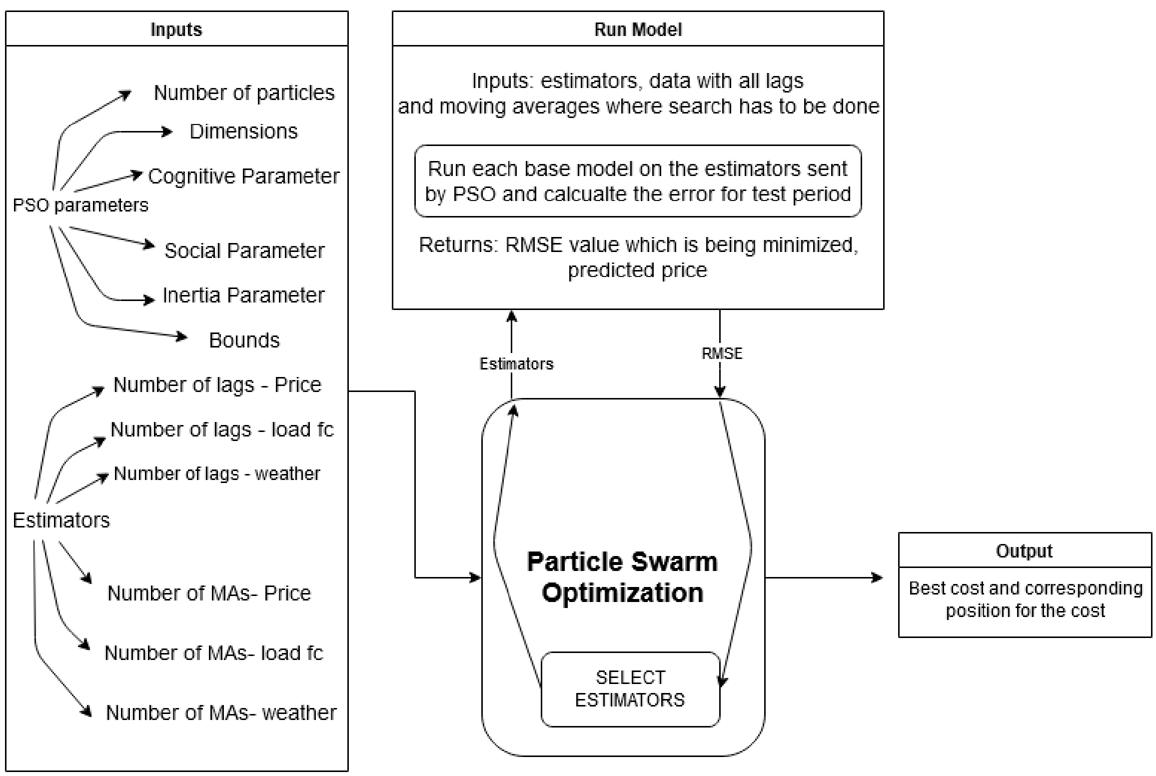

Figure 6.8: Block diagram explaining the working of PSO in current setup

For this research, particle swarm optimization was selected due to its simplicity and its ability to multi-process. Detailed explanation of the algorithm is given in Chapter

4. The lags and MAs of the 3 variables are given as an input to the PSO algorithm. The initial positions of these are chosen at random. These estimators (which the PSO algorithm will try to find the best position for) are used to call a custom function, which runs the base models (individually), by adding these estimator values as lags and moving averages to the model. This function returns the RMSE of the model, which is being minimized. The PSO updates the value of the estimators on each iteration and multiple models are built in each iteration (known as particles). The algorithm returns the best position and cost for that position of estimators. Figure 6.8 gives an overview of this method.

6.4.2 Results

Table 6.8: Error for 2016 with optimized features using PSO

Error/Models	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE	**5.39**	6.47	6.99	9.49	6.14
nRMSE	**0.186**	0.223	0.241	0.327	0.211
MAE	**3.75**	4.86	5.17	7.48	4.46

Table 6.9: Percentage change in error between base models and recency with PSO models

Model\ Error % Change	**LR**	**GB**	**RF**	**SVM**	**ANN**
RMSE/nRMSE	**-35.35%**	-11.27%	-15.64%	-3.16%	-10.81%
MAE	**-41.97%**	-12.19%	-17.17%	-1.03%	-10.16%

Table 6.8 gives the error obtained after selecting the best combination of lags and moving averages for the variables mentioned above. On comparison with Table 6.1, it can be seen that the linear regression model gives the best improvement in

performance. It shows more than 35% improvement from the base model. The other models also improve but with a relatively less percentage. GB shows approx. 12% improvement, RF shows 16% improvement, SVM shows the least - 3% improvement and ANN improves by 10% for the year 2016. Percentage change in error is given in Table 6.9. Note that these changes are from the base models.

The lag and moving average values selected by the PSO algorithm in each case is shown in Table 6.10.

Table 6.10: Lag and Moving average values selected by PSO.

Model	**Lag**			**Moving Average**		
	Price	**Load Forecast**	**Weather**	**Price**	**Load Forecast**	**Weather**
LR	64	49	66	25	23	40
GB	24	62	48	32	89	35
RF	24	44	47	26	20	27
SVM	52	63	46	24	65	95
ANN	24	49	48	25	26	40

6.4.3 Analysis of Results

Each model shows some improvement in error when compared to the base models. Given below are some observations from the improved models.

- **Linear Regression**: The model shows the most improvement in overall error. The model built with PSO shows to reduce the over forecasts that were done by the base linear regression model. Though, sometimes, the model seems to under forecast the troughs as well. This is evident from Figure 6.9.

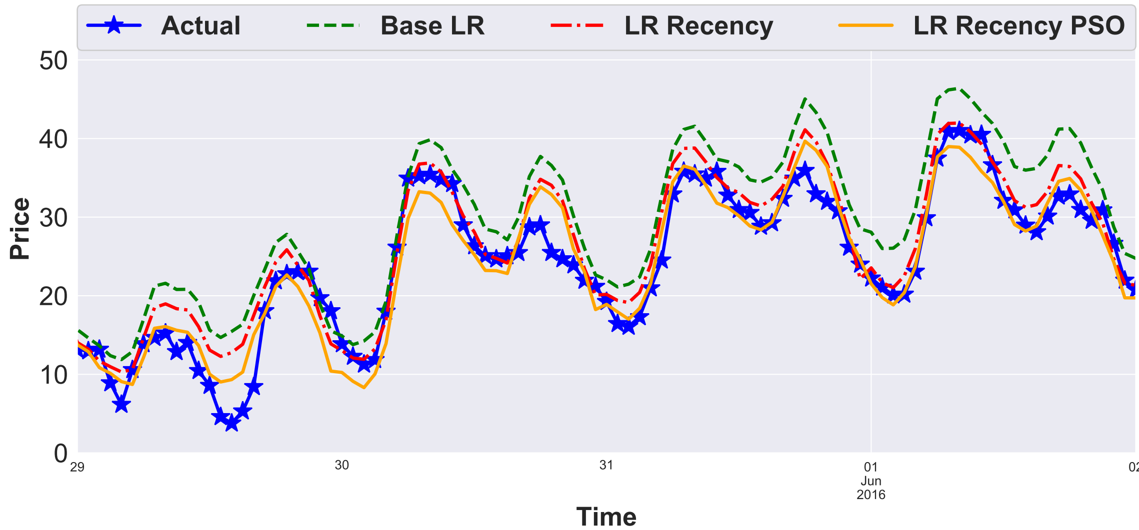

Figure 6.9: Comparison of Base LR model and LR with Recency

- **Gradient Boosting**: This model looks to capture the smaller crests and troughs better when compared to its base model. Figure 6.10 shows the comparison of the 3 models.

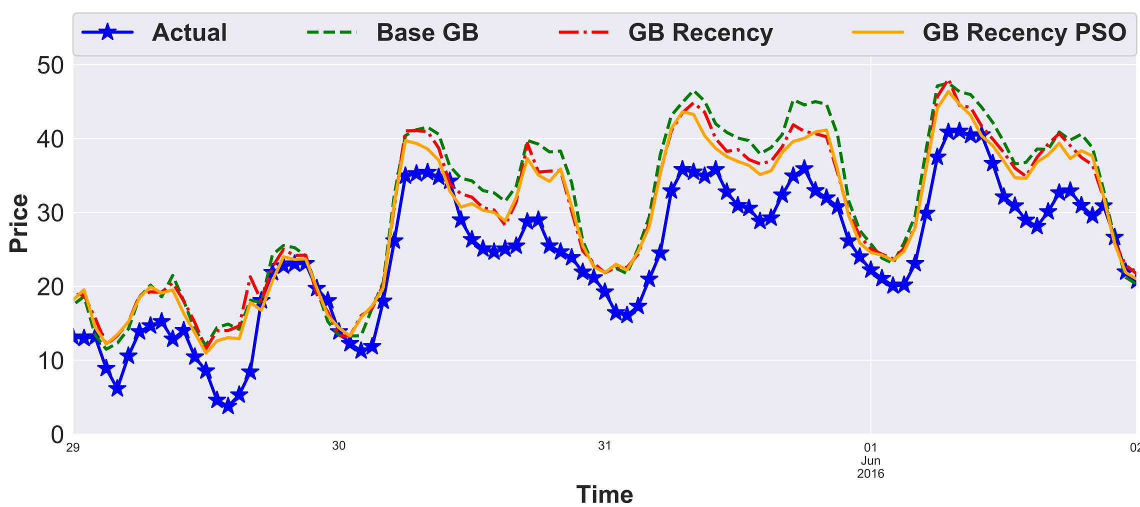

Figure 6.10: Comparison of Base GB model and GB with Recency

- **Random Forest**: This has the second best improvement when compared with the base model. Random forest captures the shape of the curve in a pretty good way, even though it fails to capture the "trend" in the price variable as can be seen in Figure 6.11.

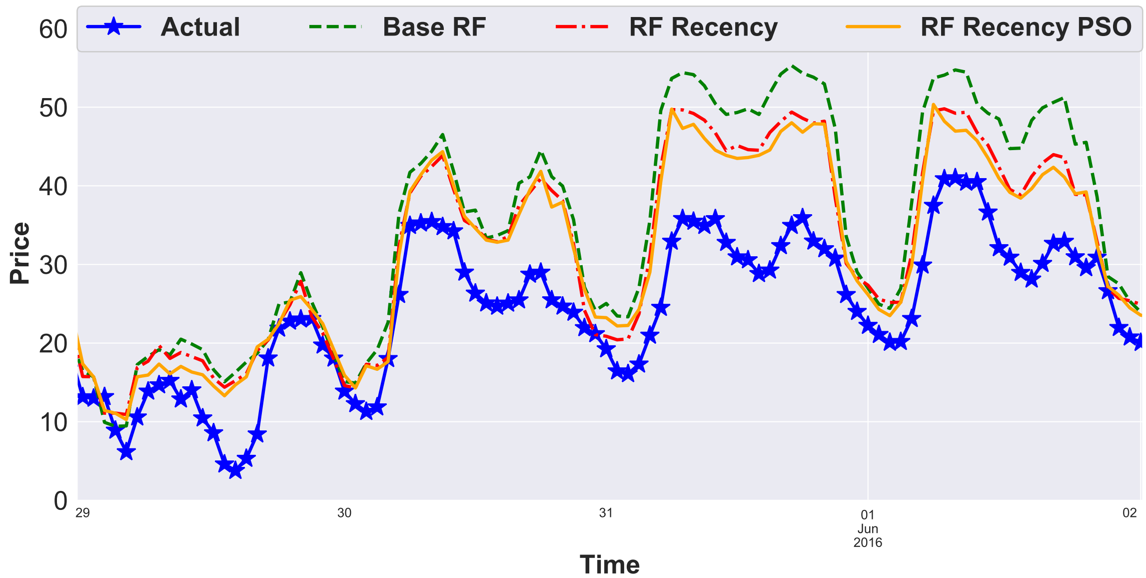

Figure 6.11: Comparison of Base RF model and RF with Recency

- **Support Vector Machines**: This shows the least improvement.

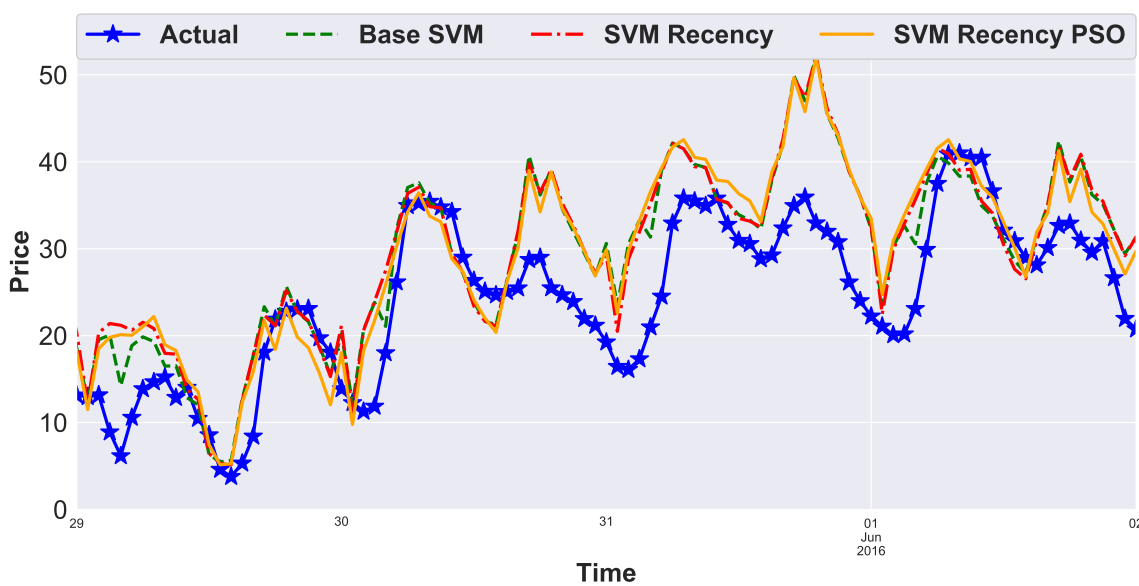

Figure 6.12: Comparison of Base RF model and RF with Recency

- **Artificial Neural Networks**: ANN also shows some improvement and is almost similar to the performance of linear regression except it does not under forecast the troughs. Figure 6.13 shows this.

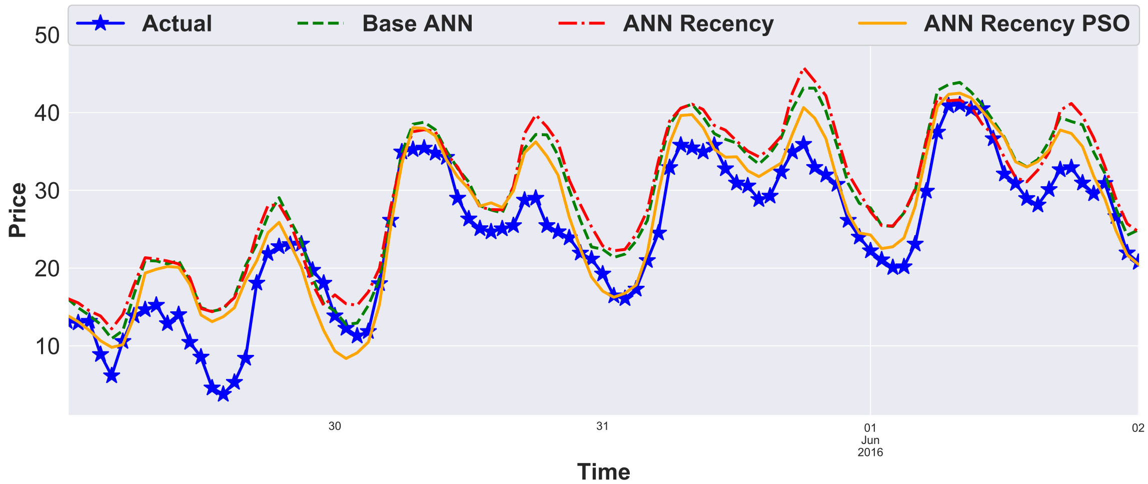

Figure 6.13: Comparison of Base ANN model and ANN with Recency

CHAPTER 7: CONCLUSIONS

7.1 Discussions

Electricity price forecasts are a fundamental part of the decision making process of energy companies. Utility companies are, in particular, vulnerable to the volatility of electricity prices as they cannot pass these down to the retail customers. Accurate forecasts can help companies make decisions which may lead to avoiding huge losses, and possibly even save them from bankruptcy. Good forecasts also help them maximize their profits as they can adjust their strategies and consumption/production profile according to the forecasts. In this thesis, advanced machine learning techniques including gradient boosting, random forests, support vector machines and artificial neural networks are explored. The focus is on selection of appropriate features from a huge search space, which cannot be solved using brute force method. To overcome this, a meta-heuristic algorithm - particle swarm optimization is employed for this task.

Use of PSO for selection of optimal features, shows a significant improvement in forecast accuracy, especially for linear regression model. It can be noted that linear regression improves the best because it is a statistical technique. It does not have any hyper parameters that have to be tuned with a change in the set of input features. This is not the case with other machine learning techniques. For example, gradient boosting has various parameters like learning rate, number of estimators, subsample size etc. which would need to be changed if the set of input features change. If these would also be included as the estimators of PSO, the already huge search space would increase and PSO would require more iteration, and hence more time to come to a near optimal solution. In the literature, there is a lot of work which build "hybrid models" by using a meta heuristic algorithm to tune the hyper parameters.

The PSO algorithm is a great work around for a computationally infeasible problem. It also has multiprocessing capabilities, because of which, multiple models can be built in parallel, to leverage the computation resources available. This can reduce the search time by a huge amount of time.

7.2 Future Work

As mentioned in the conclusions, the improvement of the machine learning models is comparatively less than improvement in the linear regression model, work can be done on improvement of the performance of these techniques. One possible reason for sub-optimal performance improvement in case of techniques other than linear regression can be un-optimized hyperparameters of these models. Along with the feature selection by PSO, hyper parameter tuning can be done for each particle in PSO algorithm. This can be possible by running a grid search for the parameters. But this can lead to an exponential increase in the search space. To overcome this, hyper parameter tuning can aso be done using some metaheuristic technique.

Another extension to this research can be exploration of other meta heuristic techniques. Some example of meta heuristic techniques that can be tried are genetic algorithm, firefly algorithm, ant colony optimization, cuckoo search algorithm etc.

The effect of proposed methodology can be tested in other domains. For example, similar lag and moving average combination selection can be done for wind and solar power forecasting.

REFERENCES

[1] R. Raineri and H. Rudnick, "Restructuring the chilean electric industry: From monopoly to competition," *Power Economics*, vol. 30, 1998.

[2] R. Weron, *Modeling and forecasting electricity loads and prices: A statistical approach*, vol. 403. John Wiley & Sons, 2007.

[3] V. Kaminski, *Energy markets*. Risk Books, 2012.

[4] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International journal of forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.

[5] P. L. Joskow, "California's electricity crisis," *oxford review of Economic Policy*, vol. 17, no. 3, pp. 365–388, 2001.

[6] T. Hong, "Crystal Ball Lessons in Predictive Analytics." `https://www.energycentral.com/c/um/crystal-ball-lessons-predictive-analytics`, 2015. [Online; accessed 19-October-2019].

[7] H. Weigt and C. von Hirschhausen, "Price formation and market power in the german wholesale electricity market in 2006," *Energy policy*, vol. 36, no. 11, pp. 4227–4234, 2008.

[8] W. Lise, B. F. Hobbs, and S. Hers, "Market power in the european electricity market—the impacts of dry weather and additional transmission capacity," *Energy Policy*, vol. 36, no. 4, pp. 1331–1343, 2008.

[9] J. Viehmann, "Risk premiums in the german day-ahead electricity market," *Energy policy*, vol. 39, no. 1, pp. 386–394, 2011.

[10] P. Zweifel, A. Praktiknjo, and G. Erdmann, *Energy economics: theory and applications*. Springer, 2017.

[11] I. P. Panapakidis and A. S. Dagoumas, "Day-ahead electricity price forecasting via the application of artificial neural network based models," *Applied Energy*, vol. 172, pp. 132–151, June 2016.

[12] D. Singhal and K. Swarup, "Electricity price forecasting using artificial neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 33, pp. 550–555, Mar. 2011.

[13] D. Keles, J. Scelle, F. Paraschiv, and W. Fichtner, "Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks," *Applied Energy*, vol. 162, pp. 218–230, Jan. 2016.

[14] A. Khosravi, S. Nahavandi, and D. Creighton, "Quantifying uncertainties of neural network-based electricity price forecasts," *Applied Energy*, vol. 112, pp. 120–129, Dec. 2013.

[15] C. Silvano, G. Giulia, P. Linda, and R. Marco, "Modeling and forecasting of electricity spot-prices: Computational intelligence vs classical econometrics," *AI Communications*, no. 3, pp. 301–314, 2014.

[16] S. Voronin and J. Partanen, "Forecasting electricity price and demand using a hybrid approach based on wavelet transform, ARIMA and neural networks: Forecasting electricity price and demand," *International Journal of Energy Research*, vol. 38, pp. 626–637, Apr. 2014.

[17] S. Anbazhagan and N. Kumarappan, "A neural network approach to day-ahead deregulated electricity market prices classification," *Electric Power Systems Research*, vol. 86, pp. 140–150, May 2012.

[18] S. Anbazhagan and N. Kumarappan, "Day-ahead deregulated electricity market price forecasting using neural network input featured by DCT," *Energy Conversion and Management*, vol. 78, pp. 711–719, Feb. 2014.

[19] G. Dudek, "Multilayer perceptron for GEFCom2014 probabilistic electricity price forecasting," *International Journal of Forecasting*, vol. 32, pp. 1057–1060, July 2016.

[20] M. Cerjan, M. Matijaš, and M. Delimar, "Dynamic Hybrid Model for Short-Term Electricity Price Forecasting," *Energies*, vol. 7, pp. 3304–3318, May 2014.

[21] D. Huang, H. Zareipour, W. D. Rosehart, and N. Amjady, "Data Mining for Electricity Price Classification and the Application to Demand-Side Management," *IEEE Transactions on Smart Grid*, vol. 3, pp. 808–817, June 2012.

[22] Y.-Y. Hong and C.-P. Wu, "Day-Ahead Electricity Price Forecasting Using a Hybrid Principal Component Analysis Network," *Energies*, vol. 5, pp. 4711–4725, Nov. 2012.

[23] C. Unsihuay-Vila, A. Zambroni de Souza, J. Marangon-Lima, and P. Balestrassi, "Electricity demand and spot price forecasting using evolutionary computation combined with chaotic nonlinear dynamic model," *International Journal of Electrical Power & Energy Systems*, vol. 32, pp. 108–116, Feb. 2010.

[24] N. Kunwar, Y. K., and R. Kumar, "Area-Load Based Pricing in DSM Through ANN and Heuristic Scheduling," *IEEE Transactions on Smart Grid*, vol. 4, pp. 1275–1281, Sept. 2013.

[25] M. Shafie-khah, M. P. Moghaddam, and M. Sheikh-El-Eslami, "Price forecasting of day-ahead electricity markets using a hybrid forecast method," *Energy Conversion and Management*, vol. 52, pp. 2165–2169, May 2011.

[26] X. Chen, Z. Y. Dong, K. Meng, Y. Xu, K. P. Wong, and H. W. Ngan, "Electricity Price Forecasting With Extreme Learning Machine and Bootstrapping," *IEEE Transactions on Power Systems*, vol. 27, pp. 2055–2062, Nov. 2012.

[27] L. d. S. Coelho and A. A. Santos, "A RBF neural network model with GARCH errors: Application to electricity price forecasting," *Electric Power Systems Research*, vol. 81, pp. 74–83, Jan. 2011.

[28] W.-M. Lin, H.-J. Gow, and M.-T. Tsai, "Electricity price forecasting using Enhanced Probability Neural Network," *Energy Conversion and Management*, vol. 51, pp. 2707–2714, Dec. 2010.

[29] W.-M. Lin, H.-J. Gow, and M.-T. Tsai, "An enhanced radial basis function network for short-term electricity price forecasting," *Applied Energy*, vol. 87, pp. 3226–3234, Oct. 2010.

[30] Lei Wu and M. Shahidehpour, "A Hybrid Model for Day-Ahead Price Forecasting," *IEEE Transactions on Power Systems*, vol. 25, pp. 1519–1530, Aug. 2010.

[31] M. Shahidehpour and L. Wu, "A hybrid model for integrated day-ahead electricity price and load forecasting in smart grid," *IET Generation, Transmission & Distribution*, vol. 8, pp. 1937–1950, Dec. 2014.

[32] N. Amjady, A. Daraeepour, and F. Keynia, "Day-ahead electricity price forecasting by modified relief algorithm and hybrid neural network," *IET Generation, Transmission & Distribution*, vol. 4, no. 3, p. 432, 2010.

[33] S. Anbazhagan and N. Kumarappan, "Day-Ahead Deregulated Electricity Market Price Forecasting Using Recurrent Neural Network," *IEEE Systems Journal*, vol. 7, pp. 866–872, Dec. 2013.

[34] F. Keynia, "A new feature selection algorithm and composite neural network for electricity price forecasting," *Engineering Applications of Artificial Intelligence*, vol. 25, pp. 1687–1697, Dec. 2012.

[35] S. Lahmiri, "Comparing Variational and Empirical Mode Decomposition in Forecasting Day-Ahead Energy Prices," *IEEE Systems Journal*, vol. 11, pp. 1907–1910, Sept. 2017.

[36] A. R. Gollou and N. Ghadimi, "A new feature selection and hybrid forecast engine for day-ahead price forecasting of electricity markets," *Journal of Intelligent & Fuzzy Systems*, vol. 32, pp. 4031–4045, May 2017.

[37] V. Sharma and D. Srinivasan, "A hybrid intelligent model based on recurrent neural networks and excitable dynamics for price prediction in deregulated electricity market," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 1562–1574, May 2013.

[38] J. Lago, F. De Ridder, P. Vrancx, and B. De Schutter, "Forecasting day-ahead electricity prices in Europe: The importance of considering market integration," *Applied Energy*, vol. 211, pp. 890–903, Feb. 2018.

[39] S. Chakravarty and P. Dash, "Dynamic filter weights neural network model integrated with differential evolution for day-ahead price forecasting in energy market," *Expert Systems with Applications*, vol. 38, pp. 10974–10982, Sept. 2011.

[40] P. Mandal, A. K. Srivastava, T. Senjyu, and M. Negnevitsky, "A new recursive neural network algorithm to forecast electricity price for PJM day-ahead market," *International Journal of Energy Research*, vol. 34, pp. 507–522, May 2010.

[41] C. Wan, Z. Xu, Y. Wang, Z. Y. Dong, and K. P. Wong, "A Hybrid Approach for Probabilistic Forecasting of Electricity Price," *IEEE Transactions on Smart Grid*, vol. 5, pp. 463–470, Jan. 2014.

[42] Z. Yang, L. Ce, and L. Lian, "Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods," *Applied Energy*, vol. 190, pp. 291–305, Mar. 2017.

[43] N. A. Shrivastava and B. K. Panigrahi, "A hybrid wavelet-ELM based short term price forecasting for electricity markets," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 41–50, Feb. 2014.

[44] M. Rafiei, T. Niknam, and M.-H. Khooban, "Probabilistic Forecasting of Hourly Electricity Price by Generalization of ELM for Usage in Improved Wavelet Neural Network," *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 71–79, Feb. 2017.

[45] N. Anand Shrivastava and B. Ketan Panigrahi, "Point and prediction interval estimation for electricity markets with machine learning techniques and wavelet transforms," *Neurocomputing*, vol. 118, pp. 301–310, Oct. 2013.

[46] D. Wang, H. Luo, O. Grunder, Y. Lin, and H. Guo, "Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm," *Applied Energy*, vol. 190, pp. 390–407, Mar. 2017.

[47] N. Chaâbane, "A hybrid ARFIMA and neural network model for electricity price prediction," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 187–194, Feb. 2014.

[48] L. Xiao, W. Shao, M. Yu, J. Ma, and C. Jin, "Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting," *Applied Energy*, vol. 198, pp. 203–222, July 2017.

[49] M. K. Kim, "Short-term price forecasting of Nordic power market by combination Levenberg-Marquardt and Cuckoo search algorithms," *IET Generation, Transmission & Distribution*, vol. 9, pp. 1553–1563, Oct. 2015.

[50] N. Amjady and F. Keynia, "Application of a new hybrid neuro-evolutionary system for day-ahead price forecasting of electricity markets," *Applied Soft Computing*, vol. 10, pp. 784–792, June 2010.

[51] N. Amjady and F. Keynia, "A new prediction strategy for price spike forecasting of day-ahead electricity markets," *Applied Soft Computing*, vol. 11, pp. 4246–4256, Sept. 2011.

[52] J. Lago, F. De Ridder, and B. De Schutter, "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms," *Applied Energy*, vol. 221, pp. 386–405, July 2018.

[53] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, pp. 2546–2554, 2011.

[54] O. Abedinia, N. Amjady, M. Shafie-khah, and J. Catalão, "Electricity price forecast using Combinatorial Neural Network trained by a new stochastic search method," *Energy Conversion and Management*, vol. 105, pp. 642–654, Nov. 2015.

[55] H. Ebrahimian, S. Barmayoon, M. Mohammadi, and N. Ghadimi, "The price prediction for the energy market based on a new method," *Economic Research-Ekonomska Istraživanja*, vol. 31, pp. 313–337, Jan. 2018.

[56] P. Bento, J. Pombo, M. Calado, and S. Mariano, "A bat optimized neural network and wavelet transform approach for short-term price forecasting," *Applied Energy*, vol. 210, pp. 88–97, Jan. 2018.

[57] W. Gao, A. Darvishan, M. Toghani, M. Mohammadi, O. Abedinia, and N. Ghadimi, "Different states of multi-block based forecast engine for price and load prediction," *International Journal of Electrical Power & Energy Systems*, vol. 104, pp. 423–435, Jan. 2019.

[58] J. Wang, F. Liu, Y. Song, and J. Zhao, "A novel model: Dynamic choice artificial neural network (DCANN) for an electricity price forecasting system," *Applied Soft Computing*, vol. 48, pp. 281–297, Nov. 2016.

[59] N. Singh, S. R. Mohanty, and R. Dev Shukla, "Short term electricity price forecast based on environmentally adapted generalized neuron," *Energy*, vol. 125, pp. 127–139, Apr. 2017.

[60] L. Peng, S. Liu, R. Liu, and L. Wang, "Effective long short-term memory with differential evolution algorithm for electricity price prediction," *Energy*, vol. 162, pp. 1301–1314, Nov. 2018.

[61] J. Che and J. Wang, "Short-term electricity prices forecasting based on support vector regression and Auto-regressive integrated moving average modeling," *Energy Conversion and Management*, vol. 51, pp. 1911–1917, Oct. 2010.

[62] X. Yan and N. A. Chowdhury, "Mid-term electricity market clearing price forecasting: A hybrid LSSVM and ARMAX approach," *International Journal of Electrical Power & Energy Systems*, vol. 53, pp. 20–26, Dec. 2013.

[63] X. Yan and N. A. Chowdhury, "Mid-term electricity market clearing price forecasting: A multiple SVM approach," *International Journal of Electrical Power & Energy Systems*, vol. 58, pp. 206–214, June 2014.

[64] N. A. Shrivastava, A. Khosravi, and B. K. Panigrahi, "Prediction Interval Estimation of Electricity Prices Using PSO-Tuned Support Vector Machines," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 322–331, Apr. 2015.

[65] L. Saini, S. Aggarwal, and A. Kumar, "Parameter optimisation using genetic algorithm for support vector machine-based price-forecasting model in National electricity market," *IET Generation, Transmission & Distribution*, vol. 4, no. 1, p. 36, 2010.

[66] H. Shayeghi, A. Ghasemi, M. Moradzadeh, and M. Nooshyar, "Simultaneous day-ahead forecasting of electricity price and load in smart grids," *Energy Conversion and Management*, vol. 95, pp. 371–384, May 2015.

[67] H. Shayeghi and A. Ghasemi, "Day-ahead electricity prices forecasting by a modified CGSA technique and hybrid WT in LSSVM based scheme," *Energy Conversion and Management*, vol. 74, pp. 482–491, Oct. 2013.

[68] D. Niu, D. Liu, and D. D. Wu, "A soft computing system for day-ahead electricity price forecasting," *Applied Soft Computing*, vol. 10, pp. 868–875, June 2010.

[69] J. Zhang and Z. Tan, "Day-ahead electricity price forecasting using WT, CLSSVM and EGARCH model," *International Journal of Electrical Power & Energy Systems*, vol. 45, pp. 362–368, Feb. 2013.

[70] N. Amjady and F. Keynia, "Electricity market price spike analysis by a hybrid data model and feature selection technique," *Electric Power Systems Research*, vol. 80, pp. 318–327, Mar. 2010.

[71] S. Voronin and J. Partanen, "Price Forecasting in the Day-Ahead Energy Market by an Iterative Method with Separate Normal Price and Price Spike Frameworks," *Energies*, vol. 6, pp. 5897–5920, Nov. 2013.

[72] J. Zhang, Z. Tan, and S. Yang, "Day-ahead electricity price forecasting by a new hybrid method," *Computers & Industrial Engineering*, vol. 63, pp. 695–701, Nov. 2012.

[73] X. Yan and N. A. Chowdhury, "Mid-term electricity market clearing price forecasting utilizing hybrid support vector machine and auto-regressive moving average with external input," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 64–70, Dec. 2014.

[74] P. Sarikprueck, W.-J. Lee, A. Kulvanitchaiyanunt, V. C. P. Chen, and J. Rosenberger, "Novel Hybrid Market Price Forecasting Method With Data Clustering Techniques for EV Charging Station Application," *IEEE Transactions on Industry Applications*, vol. 51, pp. 1987–1996, May 2015.

[75] T. Pinto, T. M. Sousa, I. Praça, Z. Vale, and H. Morais, "Support Vector Machines for decision support in electricity markets' strategic bidding," *Neurocomputing*, vol. 172, pp. 438–445, Jan. 2016.

[76] A. Ghasemi, H. Shayeghi, M. Moradzadeh, and M. Nooshyar, "A novel hybrid algorithm for electricity price and load forecasting in smart grids with demand-side management," *Applied Energy*, vol. 177, pp. 40–59, Sept. 2016.

[77] M. Alamaniotis, D. Bargiotas, N. G. Bourbakis, and L. H. Tsoukalas, "Genetic Optimal Regression of Relevance Vector Machines for Electricity Pricing Signal Forecasting in Smart Grids," *IEEE Transactions on Smart Grid*, vol. 6, pp. 2997–3005, Nov. 2015.

[78] T. Papadimitriou, P. Gogas, and E. Stathakis, "Forecasting energy markets using support vector machines," *Energy Economics*, vol. 44, pp. 135–142, July 2014.

[79] E. E. Elattar, "Day-ahead price forecasting of electricity markets based on local informative vector machine," *IET Generation, Transmission & Distribution*, vol. 7, pp. 1063–1071, Oct. 2013.

[80] J. P. S. Catalao, H. M. I. Pousinho, and V. M. F. Mendes, "Hybrid Wavelet-PSO-ANFIS Approach for Short-Term Electricity Prices Forecasting," *IEEE Transactions on Power Systems*, vol. 26, pp. 137–144, Feb. 2011.

[81] A. Motamedi, H. Zareipour, and W. D. Rosehart, "Electricity Price and Demand Forecasting in Smart Grids," *IEEE Transactions on Smart Grid*, vol. 3, pp. 664–674, June 2012.

[82] H. Pousinho, V. Mendes, and J. Catalão, "Short-term electricity prices forecasting in a competitive market by a hybrid PSO-ANFIS approach," *International Journal of Electrical Power & Energy Systems*, vol. 39, pp. 29–35, July 2012.

[83] J. Catalão, H. Pousinho, and V. Mendes, "Short-term electricity prices forecasting in a competitive market by a hybrid intelligent approach," *Energy Conversion and Management*, vol. 52, pp. 1061–1065, Feb. 2011.

[84] G. Osório, J. Matias, and J. Catalão, "Electricity prices forecasting by a hybrid evolutionary-adaptive methodology," *Energy Conversion and Management*, vol. 80, pp. 363–373, Apr. 2014.

[85] P. Gaillard, Y. Goude, and R. Nedellec, "Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting," *International Journal of Forecasting*, vol. 32, pp. 1038–1050, July 2016.

[86] F. Serinaldi, "Distributional modeling and short-term forecasting of electricity prices by generalized additive models for location, scale and shape," *Energy Economics*, vol. 33, no. 6, pp. 1216–1226, 2011.

[87] J. Nowotarski and R. Weron, "Computing electricity spot price prediction intervals using quantile regression and forecast averaging," *Computational Statistics*, vol. 30, no. 3, pp. 791–803, 2015.

[88] K. F. Wallis, "Combining density and interval forecasts: a modest proposal," *Oxford Bulletin of Economics and Statistics*, vol. 67, pp. 983–994, 2005.

[89] M. Lei and Z. Feng, "A proposed grey model for short-term electricity price forecasting in competitive power markets," *International Journal of Electrical Power & Energy Systems*, vol. 43, pp. 531–538, Dec. 2012.

[90] N. Haldrup, F. S. Nielsen, and M. Ø. Nielsen, "A vector autoregressive model for electricity prices subject to long memory and regime switching," *Energy Economics*, vol. 32, pp. 1044–1058, Sept. 2010.

[91] H. C. Wu, S. C. Chan, K. M. Tsui, and Y. Hou, "A New Recursive Dynamic Factor Analysis for Point and Interval Forecast of Electricity Price," *IEEE Transactions on Power Systems*, vol. 28, pp. 2352–2365, Aug. 2013.

[92] G. Dorini, P. Pinson, and H. Madsen, "Chance-Constrained Optimization of Demand Response to Price Signals," *IEEE Transactions on Smart Grid*, vol. 4, pp. 2072–2080, Dec. 2013.

[93] P. Mandal, Haque, J. Meng, Srivastava, and R. Martinez, "A novel hybrid approach using wavelet, firefly algorithm, and fuzzy ARTMAP for day-ahead electricity price forecasting," *IEEE Transactions on Power Systems*, vol. 28, pp. 1041–1051, May 2013.

[94] Y. Dong, J. Wang, H. Jiang, and J. Wu, "Short-term electricity price forecast based on the improved hybrid model," *Energy Conversion and Management*, vol. 52, pp. 2987–2995, Aug. 2011.

[95] J. Nowotarski, E. Raviv, S. Trück, and R. Weron, "An empirical comparison of alternative schemes for combining electricity spot price forecasts," *Energy Economics*, vol. 46, pp. 395–412, Nov. 2014.

[96] J. Alvarez-Ramirez and R. Escarela-Perez, "Time-dependent correlations in electricity markets," *Energy Economics*, vol. 32, pp. 269–277, Mar. 2010.

[97] C. H. Jin, G. Pok, Y. Lee, H.-W. Park, K. D. Kim, U. Yun, and K. H. Ryu, "A SOM clustering pattern sequence-based next symbol prediction method for day-ahead direct electricity load and price forecasting," *Energy Conversion and Management*, vol. 90, pp. 84–92, Jan. 2015.

[98] W. Wang and X. Liu, "Fuzzy forecasting based on automatic clustering and axiomatic fuzzy set classification," *Information Sciences*, vol. 294, pp. 78–94, Feb. 2015.

[99] A. Karsaz, H. R. Mashhadi, and M. Mirsalehi, "Market clearing price and load forecasting using cooperative co-evolutionary approach," *International Journal of Electrical Power & Energy Systems*, vol. 32, pp. 408–415, June 2010.

[100] K. A. De Jong and M. A. Potter, "Evolving complex structures via cooperative coevolution.," in *Evolutionary Programming*, pp. 307–317, 1995.

[101] K. Maciejowska and J. Nowotarski, "A hybrid model for GEFCom2014 probabilistic electricity price forecasting," *International Journal of Forecasting*, vol. 32, pp. 1051–1056, July 2016.

[102] E. Crisostomi, C. Gallicchio, A. Micheli, M. Raugi, and M. Tucci, "Prediction of the Italian electricity price for smart grid applications," *Neurocomputing*, vol. 170, pp. 286–295, Dec. 2015.

[103] I. Juárez, J. Mira-McWilliams, and C. González, "Important variable assessment and electricity price forecasting based on regression tree models: classification and regression trees, Bagging and Random Forests," *IET Generation, Transmission & Distribution*, vol. 9, pp. 1120–1128, Aug. 2015.

[104] Z. Tan, J. Zhang, J. Wang, and J. Xu, "Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models," *Applied Energy*, vol. 87, no. 11, pp. 3606–3610, 2010.

[105] H. Liu and J. Shi, "Applying arma–garch approaches to forecasting short-term electricity prices," *Energy Economics*, vol. 37, pp. 152–166, 2013.

[106] J. M. Vilar, R. Cao, and G. Aneiros, "Forecasting next-day electricity demand and price using nonparametric functional methods," *International Journal of Electrical Power & Energy Systems*, vol. 39, no. 1, pp. 48–55, 2012.

[107] S. Bordignon, D. W. Bunn, F. Lisi, and F. Nan, "Combining day-ahead forecasts for british electricity prices," *Energy Economics*, vol. 35, pp. 88–103, 2013.

[108] T. Hong *et al.*, "Short term electric load forecasting.," 2010.

[109] P. Wang, B. Liu, and T. Hong, "Electric load forecasting with recency effect: A big data approach," *International Journal of Forecasting*, vol. 32, no. 3, pp. 585–597, 2016.

[110] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural network design*. Martin Hagan, 2014.

[111] R. Lippmann, "An introduction to computing with neural nets," *IEEE Assp magazine*, vol. 4, no. 2, pp. 4–22, 1987.

[112] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*, vol. 821. John Wiley & Sons, 2012.

[113] V. Vapnik, S. E. Golowich, and A. J. Smola, "Support vector method for function approximation, regression estimation and signal processing," in *Advances in neural information processing systems*, pp. 281–287, 1997.

[114] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[115] C. Cortes and V. Vapnik, "N, support vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–295, 1995.

[116] J. Mei, D. He, R. Harley, T. Habetler, and G. Qu, "A random forest method for real-time price forecasting in new york electricity market," in *2014 IEEE PES General Meeting| Conference & Exposition*, pp. 1–5, IEEE, 2014.

[117] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[118] S. Touzani, J. Granderson, and S. Fernandes, "Gradient boosting machine for modeling the energy consumption of commercial buildings," *Energy and Buildings*, vol. 158, pp. 1533–1543, 2018.

[119] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[120] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[121] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[122] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020.

[123] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[124] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[125] Y. Shevchuk, "Neupy: neural networks in python," 2019.

[126] L. J. V. Miranda, "PySwarms, a research-toolkit for Particle Swarm Optimization in Python," *Journal of Open Source Software*, vol. 3, 2018.

[127] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016.

[128] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

APPENDIX A: PSEUDO CODE FOR BASE MODELS

This appendix consists of pseudo code for base models.

Algorithm 4: Base Models

Result: Electricity Price Forecast

Read input data - price and other explanatory variables;

Add holidays of different regions of Germany;

Add calendar variables - hour of day, week of year, month of year;

Add square and cube of load forecast and price variables;

Split Y(price) and X(other independent variables);

Split Y and X into train (year 2015) and test (year 2016) set;

Train models based on methodology in Chapter 4;

Predict values for test set based on trained model;

Obtain error metrics as described in Section 4.7

APPENDIX B: PSEUDO CODE FOR RECENCY MODELS

This appendix consists of pseudo code for recency models.

Algorithm 5: Recency Models

Result: Electricity Price Forecast

Read input data - price and other explanatory variables;
Add holidays of different regions of Germany;
Add calendar variables - hour of day, week of year, month of year;
Add square and cube of load forecast and price variables;
Add 24 hr. lag and 48 hr. moving average for price, load forecast and weather;
Split Y(price) and X(other independent variables);
Split Y and X into train (year 2015) and test (year 2016) set;
Train models based on methodology in Chapter 4;
Predict values for test set based on trained model;
Obtain error metrics as described in Section 4.7

APPENDIX C: PSEUDO CODE FOR SELECTION OF OPTIMAL COMBINATION FOR RECENCY USING PSO

This appendix consists of pseudo code for optimal combination selection using PSO.

Algorithm 6: Optimal Selection of lag and moving average combinations

Result: Best combination of lags and moving averages for given variables

```
Read input data - price and other explanatory variables;
Add holidays of different regions of Germany;
Add calendar variables - hour of day, week of year, month of year;
Add square and cube of load forecast and price variables;
Initialize the parameters and estimators for PSO as in Algorithm 3;
while current iteration < max number of iterations do
    for each particle do
        Add lags and moving averages of price, load forecast and weather
          variables based on PSO estimators;
        Split Y(price) and X(other independent variables);
        Split Y and X into train (year 2015) and test (year 2016) set;
        Train models based on methodology in Chapter 4;
        Predict values for test set based on trained model;
        Obtain and return error metrics as described in Section 4.7;
        Update position of estimators according to Algorithm 3;
    end
end
```