TECHNISCHE UNIVERSITÄT CHEMNITZ

Chair of Communication networks

Master Thesis

A smart scheduling algorithm for a decentralized energy management system

Sunil Hebbur Srikantamurthy

Chemnitz, December 20th, 2016

Author	: **Sunil Hebbur Srikantamurthy**	
Email	: **sunil.srikantamurthy1990@gmail.com**	
Matriculation number	: **421463**	
Assessors	: **Prof. Dr.-Ing. Thomas Bauschert**	(TUC)
	Dr. Thomas Walter	(Easy Smart Grid GmbH)
Supervisor	: **Dr.-Ing. Dipl.-Math. Andreas Baumgartner**	(TUC)

Start date : July 14th, 2017
End date : December 20th, 2017

Abstract

Renewable energy harvesting has been a key area of interest for the past few decades. In the recent years, several research efforts have gone into outlining the design of the new electrical grids. Today, the existing electrical grids and architectures are undergoing several improvements to harness the maximum possible electrical energy from the renewable sources. The next generation electrical grids are coined as the "Smart Grids". These grids are intended and designed to harness the renewable energy sources, to bridge the communication between electrical utility and end users, to install the advanced metering infrastructure (AMI) and to adopt the concept of real-time pricing (RTP)[1]. This design eventually leads to reliable, resilient, effective and efficient power usage. The main motive of this thesis is to design a smart scheduling algorithm for the decentralized energy management system (DEMS). Apart from the aforementioned motives, the increase in the number of household devices demands for efficient and effective power usage. In addition to these motives, the energy demands are to be met with a possible reduced cost of consumption (CoEC), which is another challenge.

The Household devices could be categorized broadly into flexible and non-flexible devices. The flexible devices are less time-critical devices 1) The devices, whose start of operation could be delayed: such as dishwasher, water pump, washing machine (devices could also be operated at regular intervals) 2) The devices, whose start of operation could not be delayed but which can be interrupted or turned ON-OFF while operating e.g.: charging of Electrical Vehicle (EV), air conditioner and UPS/inverter. The non-flexible devices are operation time critical devices, which are user input driven, such as electric lights, oven and stove.

The DEMS adopts the "*divide and rule*" principle, i.e. it deals with monitoring and controlling the usages of each flexible device. In this work, we use DEMS principle to formulate an optimized algorithm to achieve a positive edge over the existing centralized or aggregated household energy management algorithms in terms of energy consumption cost and effective power usage. In the proposed algorithm the historic data of electricity price, device switching patterns and environmental temperature are acquired and the device switching pattern for the next 24 hours are predicted and optimized. The device's switching pattern is smartly scheduled considering the lowest electricity price, temperature (if a device is temperature dependent), possible number of interruptions and possible maximum waiting time while attempting to complete the daily load requirements. The thesis is structured into 6 sections. In SECTION I, the problem faced in traditional electrical grid and the future grid technologies are explained. SECTION II outlines the formulated DEMS algorithm along with a brief explanation of its results. SECTION III and SECTION IV cover the explanation of the algorithm in detail. The performance evaluation of the proposed algorithm is stated in SECTION V. SECTION VI gives a summary and concludes this thesis.

[1]RTP: Real-time pricing is a conceptual approach countering the existing trend of fixed electricity price for the consumption at the retailer end, irrespective of the cost of electricity generation. RTP provides the user, the information about the actual electricity price at any given time. Thereby allowing him to adjust the electricity usage by scheduling the device's operation.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AMR : Automatic meter reading.

AMI : Advanced metering infrastructure.

CoEC : Cost of energy consumption.

CEMS : Centralized energy management systems.

DEMS : Decentralized energy management systems.

DLM : Dynamic Linear Model

EMU : Central Energy Management Unit.

EV : Electrical vehicles.

HAN : Home area network.

iHEM : In-home energy management.

MDP : Markov decision process.

MAPE : Mean absolute percent error.

NAN : Neighborhood area network.

RTP : Real time pricing.

RTM : Real time monitoring.

SSM : State Space Models

STS : Stochastic scheduling.

WSN's : Wireless Sensor Networks.

WAN : Wide area network.

List of Symbols

$INTR_D^{Max}$: Maximum allowed number of interruption of a device in a day. It is a safety critical operation.

$U\_INTR_D^{Max}$: Count of interruption, updated at regular intervals e.g. sampling intervals.

W_D^{Max} : Maximum allowed number of waiting time of a device.

$U\_W_D^{Max}$: Present waited time, updated at regular intervals e.g. sampling intervals.

ζ_t : RTP of electricity at time instant *t*.

M_D^t : Device's operation modes such as stand-by, low power operation, medium power, high power operation of a device at time *t*.

P_D^t : Power consumption of a device at time *t.*

$P_D^{Total\_Day}$: Power consumption of a device for 24 hours.

$U\_P_D^{Total\_Day}$: Consumed power at present, updated at regular intervals e.g. Sampling intervals.

$C_t^{Total\_day}$: Total CoEC of a device for a 24 hours.

ω : Result or output or response variable of any predictor model, .I.e. ω_{Price} incase of electricity price, ω_{Load} in case of device's switching pattern. It is unoptimized prediction result.

ω_{Load}^* : Optimized device ON-OFF schedule for next 24 hours.

OC : Occurred data during the prediction horizon. e.g.: OC_{Load} or OC_{Price}.

η_{Power} : Efficiency of power usage.

SEN : Seasonality.

H : Historic data used to train the predictor models.

H_{Price} : Historic electricity price data used to train the predictor models.

H_{Load} : Historic load data used to train the predictor models, i.e. historic ON-OFF usage patterns of flexible device.

H_{Agg_Load} : Historic aggregated load data used to train the predictor models.

H_{Load}^{Anch} : Device's historic switching data of a device at city anchorage used to train the predictor models. Similarly H_{Load}^{LRck},H_{Load}^{Pal}, H_{Price}^{Anch}, H_{Price}^{LRck}

and H_{Price}^{Pal}.

h	: Hours of a day.
$std(X)$	: Standard Deviation of the data, *X.*
$H_{std(X)}$	: Standard deviation of the forecasted data, *X*: set of 96 samples, each sample at 15 min interval.
$[\upsilon_{p1}, \upsilon_{p2},..\upsilon_{pn}]$	: Set of *n* predictor variables applied as an input to the Predictor model.
$[\upsilon_{i1}, \upsilon_{i2},.... \upsilon_{in}]$	: Set of *n* intervention variables applied as an input to the Predictor model.
υ_i^{MoD}	: An intervention variable / vector comprising of *"Minutes Of Day"* data.
υ_p^{PWSDP}	: A predictor variable / vector comprising of *"Previous week Same Time Price"* data.
υ_p^{APWSDP}	: A predictor variable / vector comprising of *"Average Previous Week Same Day Price"* data.
υ_i^{T}	: An intervention variable / vector comprising of *"Temperature"* data.
υ_i^{RH}	: An intervention variable / vector comprising of *"Relative humidity"* data.
υ_i^{DP}	: An intervention variable / vector comprising of *"Dew point"* data.
υ_i^{WS}	: An intervention variable / vector comprising of *"Wind speed"* data.
υ_p^{PWSDL}	: A predictor variable / vector comprising of *"Previous week Same Time Load"* data.
υ_p^{APWSDL}	: A predictor variable / vector comprising of *"Average Previous Week Same Day Load"* data.
υ_i^{WiS}	: An intervention variable / vector comprising of *"Weekend is true"* data.
υ_i^{WDN}	: An intervention variable / vector comprising of *"Weeks Day Number"* data.

υ_i^{HoD} : An intervention variable / vector comprising of *"Hour Of Day"* data.

TD : The training period, at which the *H,* predictor variables and intervention variables are sampled and applied as input to the predictor model.

π^* : Optimum decision/ policy outputted from MDP at an instant. E.g.: Optimum decision for next 15 min.

μ_{MAPE}^{year} : The Mean of Prediction error/MAPE over entire year in percentage.

$\mu_{accuracy}^{year}$: The Mean of Prediction accuracies over entire year in percentage.

$H_{std(accuracy)}^{year}$: The average of standard deviation of prediction accuracies over entire year in percentage.

A : The prediction accuracy.

Acknowledgement

I would like to thank Prof. Dr.-Ing. Thomas Bauschert and Dr.-Ing. Andreas Baumgartner, for guiding me technically and being supportive morally throughout the thesis cycle. I would like to extend my warm gratitude to Dr. Thomas Walter, for providing the opportunity to work on this thesis topic. I would like to thank all for the help, guidance and cooperation.

1. Introduction

In the existing conventional electrical grids, the generated electricity is delivered to the consumers through a complex network, *electrical grids* [13]. The construction of these electrical grids started in the early 1900's, in different countries around the globe. Initially, the infrastructure or the electrical grids were laid by several local electrical utilities. These local utilities discretely supported the electrical demands of their respective geographical regions. However, in the late 1950's, due to the increase in power demands, different electrical utilities interconnected their electrical grid i.e. interconnections through the transmission systems. These utilities were depended on its own power plants or purchased the power from other utilities. The interconnections of several local electrical grids also led to the emergence of large and jointly owned electrical generation units [13]. The power plants back then usually generated electricity from non-renewable energy sources such as oil and petroleum products, coal/fossil, hydrocarbons, natural gas or nuclear sources. Presently at several geographical regions, the existing grids have reached the limit of their lifespan and should be replaced and upgraded. During this process, it is necessary to link the energy harvested from various renewable energy sources. The trend to shift to renewable energies like hydropower, biomass, wind, geothermal and solar are increasing. On the other hand, with the emerging technologies and innovative ideas, the household appliances have been under the process of optimization. This optimization is in terms of enhanced operability, simpler user interaction and several other user supportive aspects. However, this optimization is achieved at what cost? Is it at increased electrical power? Can the device seek and operate at low electricity price? Researchers have already understood the criticality of these issues for e.g.: blackout, brownouts and increase in the power demand that could be encountered in the near future.

To address several concerns as explained above, the research on the smart grid has emerged. The smart grid technically convinces to add the low power data lines along with the existing high power electrical lines. This data lines are used to connect, to communicate and to exchange the data between power sourcing and demanding entities in the electrical grid. The exchange of data, such as the daily power usage, hourly electricity-price, hourly power demand and several other parameters and information would be essential to overcome the aforementioned issues. To define "Smart Grid", it’s rather an in-definitive concept, due to its varied perceptions in the scientific community. Some researchers believe that the prepaid power usage as smart, i.e. installation of smart meters at the user end as smart, some researchers presume that harvesting renewable energy from the sources such as solar or wind in a distributed fashion as smart grid and some agree that the establishment of communication between the power generation and consumption entities as smart. However, all these still could collectively be called as a *smart grid*.

In a smart grid, several approaches have already been developed for the smart energy management. One such major approach is real-time pricing (RTP) and smart metering. RTP or dynamic pricing is a new concept, whereby either the automatic or user applied price-aware load scheduling mechanism is employed to shift the load at peak hours and to improvise the effectiveness in power generation, thereby addressing the present day issues [3]. The smart meters constitute the advanced metering infrastructure (AMI). This establishes a communication link between the electricity consumer and electrical utility. The smart meters record the electricity consumption at regular intervals i.e. at each hour or 15 minutes or other programmable intervals. The recorded load consumption information is then updated to the electrical utility. This information helps the utility to better analyze the hourly load demands and thereby, the energy demand to the energy generation efficiency could be improved. The smart meter enables features such as automatic meter reading (AMR). The author in [1] proposes a centralized energy management system (CEMS) based on RTP, in which the entire building (e.g.: a house or an office) is considered as a single entity containing a single monitoring and controlling unit. This controlling unit manages all the device operations inside the building. The algorithm aims to achieve minimum aggregated CoEC by directing the operations of several flexible devices at an instant, based on the stochastic behavior of CoEC. As in [2], Wireless Sensor Networks (WSN's) are used for an in-home energy management (iHEM) application. The schemes are based on the communication between the smart appliances, a central Energy Management Unit (EMU), a smart meter and a storage unit. When a consumer turns a device ON, a data packet is received by the EMU. The EMU communicates with the smart meter and fetches the RTP: based on the electricity price information, the EMU reschedules the operation of the device by shifting the operation to off-peak periods and thereby reducing the CoEC. Stating a real world example of smart grid applications and the communication technologies, the author in [5] provides information on the network requirements and the communication technology for varied smart grid applications. The application is used in the Home area network (HAN), Neighborhood area network (NAN) and Wide area network (WAN). The author outlines the low cost and the lower power consumption communication technologies for the future smart grid applications. However, even with all these efforts in the direction of the smart grid, there exists no state of the art algorithm or a smart grid architecture which could be globally accepted due to its varied perceptions as aforementioned.

To formulate and optimize algorithm considering many volatile real world aspects and parameters such as temperature, power demand, efficiency and eco-friendly power generation is a key driving factor. The scope of this thesis is at the user end, i.e. at the household appliance level. An attempt to improve the efficiency at the device level that in-turn poses as a positive influence on the entire grid. This decentralized system with RTP is one of the possible approaches, unlike the CEMS, which considers the entire house has single end entity, i.e. the aggregated load. This thesis is based on the decentralized energy management system (DEMS) where the end entity is each device in the house. These devices are individually monitored, their usage patterns are learned, analyzed, forecasted and finally, its operations are controlled efficiently and effectively considering various influences and parameters.

While considering the algorithm to be a smart scheduling algorithm for the DEMS, challenges in the implementation costs and infrastructure costs are addressed in parallel. This thesis is formulated and intended to use the existing microcontroller hardware in the devices. The proposed algorithm will be supplied as an update to the devices, thereby incurring negligible implementation costs/ hardware up-gradation costs/infrastructure costs.

The Household devices could be categorized broadly into flexible and non-flexible devices. The flexible devices are less time-critical devices

1) The devices, whose start of operation could be delayed: such as dishwasher, water pump, washing machine (devices could also be operated at regular intervals).

2) The devices, whose start of operation could not be delayed but which can be interrupted/turned ON-OFF while operating: such as charging of Electrical Vehicle (EV), air conditioner and UPS/inverter (devices which operate to maintain desired set points).

The non-flexible devices are operation time critical devices, which are user input driven, such as electrical lights, oven and stove. In this work, the devices considered are a Water pump, a Heater, and a Cooler, all of which are flexible devices. However, for Non -flexible devices the proposed algorithm should be retuned and evaluated.

The performance evaluation of the proposed algorithm is just as vital as the development of the algorithm itself. Therefore, better performance strategies and tests need to be formulated and conducted. In order to aid in the strong performance evaluation, the varied environmental conditions of different cities, the varied device interruption times, the varied maximum device waiting time and city's electricity pricing information has to be considered. Thereafter, from the results of these varied test cases, an inference has to be drawn.

2. Outline of Algorithm

The main motivation of the thesis is to formulate an efficient algorithm, which can schedule a flexible device's switching[2] operation such that the cost of energy consumption $C_t^{Total\_Day}$ incurred is reduced and the power efficiency η_{Power} is improved. The algorithm takes into account the seasonality *SEN*, the historic switching pattern of a specific devices H_{Load} and the specific city's historical data: temperature $H_{Temperature}^{City}$, electricity price H_{Price}^{City}, the maximum allowed device waiting time W_D^{Max}, the maximum device interruptions possible in a day $INTR_D^{Max}$, the total power requirement in a day $P_D^{Total\_Day}$, RTP ζ_t, next 24 h CoEC $C_t^{Total\_Day}$ and the total device execution time left T_{Exe}.

In *Figure 1*, an overview of the proposed DEMS algorithm is presented. In this work, the DEMS is designed as a 3 stage algorithm, similar to the 3 stage algorithm with CEMS in [6]: Real Time Monitoring (RTM), Stochastic Scheduling (STS) and Real Time Control (RTC). However, we concentrate on a decentralized approach, i.e. a single device control rather than a centralized or a simultaneous multiple device control as in [6]. The algorithm predicts the next 24 h switching patterns ω_{Load} and thereafter it optimizes these patterns, ω_{Load}^*. The predicted load patterns for the next day, i.e. the switching patterns are shifted to arrive at the new and optimized switching patterns. These shifts are based on the temperature at time *t=0* (if the device is temperature dependent), the electricity price ζ_t at time *t=0* and the next 1 h price ζ_t^h and other device specifications: $INTR_D^{Max}$, W_D^{Max}, $P_D^{Total\_Day}$, *SEN*, T_{Exe} and $C_t^{Total\_Day}$. The complexity of optimizing ω_{Load} is handled by the Markov Decision Process (MDP). The entire algorithm is run periodically on a daily basis so that the optimized switching schedule ω_{Load}^* for the next 24 h is available beforehand. SECTION III deals with the prediction phase. SECTION IV states the MDP formulation in detail, i.e. optimization phase. SECTION V covers the performance evaluation. Finally, the optimal algorithm and the supporting results are stated in SECTION VI.

Real Time Monitoring (RTM) or Phase 1:

This phase is considered to be the prediction phase. The device switching patterns are monitored periodically at a rate of 15 minutes. However, this interval rate could be further reduced to 5 minutes or even lesser, considering the growth trend in the computational power. The switching patterns are the ON-OFF patterns of a device. In phase 1, the device's next 24 h switching patterns are predicted. These predicted switching patterns are un-optimized, in which the probability of energy usage inefficiency (over power usage) η_{Power} and the CoEC is high.

[2] The device is assumed to be in ON state, when the device is performing the intended action and OFF state when it is not. In both the cases the device is in standby mode i.e. the microcontroller is active. E.g.: A Water Pump, washes the cloths when it is ON and not when it is OFF, however in both the cases the device remains powered in standby mode.

The process of using H_{Load} for forecasting ω_{Load}^{t+P} is called *training,* where P is the desired forecast horizon. The considered training period range is one week to several years, i.e. *one week ≤ TD ≤ several years*. However, the efforts are made to identify which TD would fit relatively best, i.e. either the short duration $TD \approx$ *one week*, where the usage dynamics are captured better or longer duration $TD \geq$ *one Month*, where there are chances of averaging the data and thereby, discarding the usage dynamics and the SEN to a considerable extent.

The historical data of the device is considered as training data, i.e. device switching patterns H_{Load}, the temperature of the chosen city $H_{Temperature}$ and the corresponding electricity pricing information H_{price} . The historical data are acquired at a sampling rate of 15 minutes, i.e. 96 samples in 24 h. TD of one week is considered to be an ideal training period, since it covers and learns the user dependent usage dynamics better. Accordingly, as an example, one week of H_{Load}, H_{Price} and $H_{Temperature}$ at 15 minutes sampling interval is used as training data set in phase 1 to predict ω_{Load} and ω_{Price} at a sampling interval of 15 minutes. Today, there exits several state of the art weather forecasters. In order to reduce the computational effort, the forecasted temperature $\omega_{Temperature}$ data for the next 24 h is fetched from one of those stable online sources and are directly passed on to phase 2. Hence, only the device's switching pattern ω_{Load} and electricity price ω_{Price} data is forecasted using the predictive models.

The proposed algorithm is simulated, analyzed and evaluated in MATLAB. MATLAB consists of toolboxes supporting several prediction algorithms or predictor models such as ARIMA, Neural Network (NN), state space algorithms, Hybrid Fuzzy logic and time series algorithms - ANFIS, Linear regression and nonlinear regression. To remove the ambiguities in selecting a relatively good predictor model for the smart scheduling algorithm for the DEMS, each prediction models have to be formulated and the results of each model are to be evaluated. An effort is also put into identifying the optimal training period. In addition, the different devices (E.g.: Water pump, Heater and Cooler) in combination with different cities are chosen for a quality performance evaluations and thereafter to arrive at a relatively good predictor model irrespective of varied test cases.

Stochastic Scheduling (STS) or Phase 2:

The objective of this phase is to optimize the device switching patterns, ω_{Load}. The process of optimization involves altering/shifting the device's switching pattern based on the influencing factors. This shifting has to be smart, in order to account for the uncertainties such as the seasonality SEN, the maximum allowed device waiting time W_D^{Max}, the maximum device interruptions possible in a day $INTR_D^{Max}$, the total power requirement in a day $P_D^{Total\_Day}$, RTP ζ_t, next 24 h CoEC $C_t^{Total\_Day}$ and the total device execution time left T_{Exe}. These uncertainties have to be included in the algorithm to obtain the optimized schedule ω_{Load}^* [6]. The property of randomness in any variable, when observed over any time is called *stochasticity* and scheduling the device switching in accordance with the stochastic behavior is called *stochastic scheduling*.

Markov decision processes are a useful mathematical framework within probability theory to make optimal decisions π^*. The MDP is an extension of Markov chains. The basic assumption to Markov theory is the memoryless, i.e. it is assumed that the next step could be predicted by considering solely the current step, unlike the chain of historic events. In Markov chains, it is assumed that the future and the past states are independent and the next state is only driven by the present event or state (i.e. Memory-less property). The Markov chains are discrete states *(S_t, S_{t+1},...,S_{t+n})*, regardless of the nature of time *t*, being discrete *t=1, 2...n* or continuous $t \geq 0$. An MDP is a Markov process with *m* states *S=* S_1,S_2,...,S_m and *n* possible actions per state *a=* a_1,a_2...a_n, where *n* may or may not be equal to *m*. Since MDP operates in discrete time steps or states, it can be formulated to decide the next possible optimal policy[3] π^*. The π^* obtained as a result of the MDP is the next 15 minute's ON-OFF state of the device, i.e. since the sampling interval is 15 minutes, the π^* directs the device to remain turned OFF or ON for the next 15 minutes. At each time step, the device's present state ω_{Load}^{t}, temperature, electricity price ω_{Price}^{t}, updated max waiting time $U\_W_D^{Max}$, updated max interruptions of device $U\_INTR_D^{Max}$ and updated total power usage left for the day $U\_P_D^{Total\_Day}$ are applied as inputs to the MDP. In order to arrive at the optimal switching policy for the next 24 *h*, ω_{Load}^{*}= [π_1^*,π_2^*, π_3^*,..,π_{96}^*], the MDP is iterated for 96 times ($\frac{24\,h}{15\text{Min}}$). Since there exists a requirement to obtain the status of the device, electricity price and temperature at each time step, their status is the predicted in phase 1, i.e. ω_{Load}, ω_{Price} and $\omega_{Temperature}$. At the end of phase 2, the device switching patterns are optimized. The optimized schedule is provided as an input to phase 3 where it physically switches the device. In SECTION IV the MDP formulation and the corresponding results are explained in detail. The performance evaluation of the MDP is described in SECTION V.

Real Time Control (RTC) or Phase 3:

Once the next 24 *h* switching pattern is optimized in phase 2, in phase 3 the microcontroller in the device switches the device accordingly. This thesis assumes devices with only 2 operational states, i.e. ON and OFF state. However, in future works, the algorithms can be designed for the devices operating at varied power modes M_D^t: stand-by, low, medium and high power modes.

[3]Policy: In case of an MDP, a policy means choosing an optimal action among the several possible actions. In our use case the MDP is iterated for 96 times, so policy in here means a series of ON - OFF actions (for an entire day), which is an optimal switching schedule for a flexible device to incur reduced CoEC, $C_t^{Total\_Day}$.

Flowchart: Outline of Algorithm

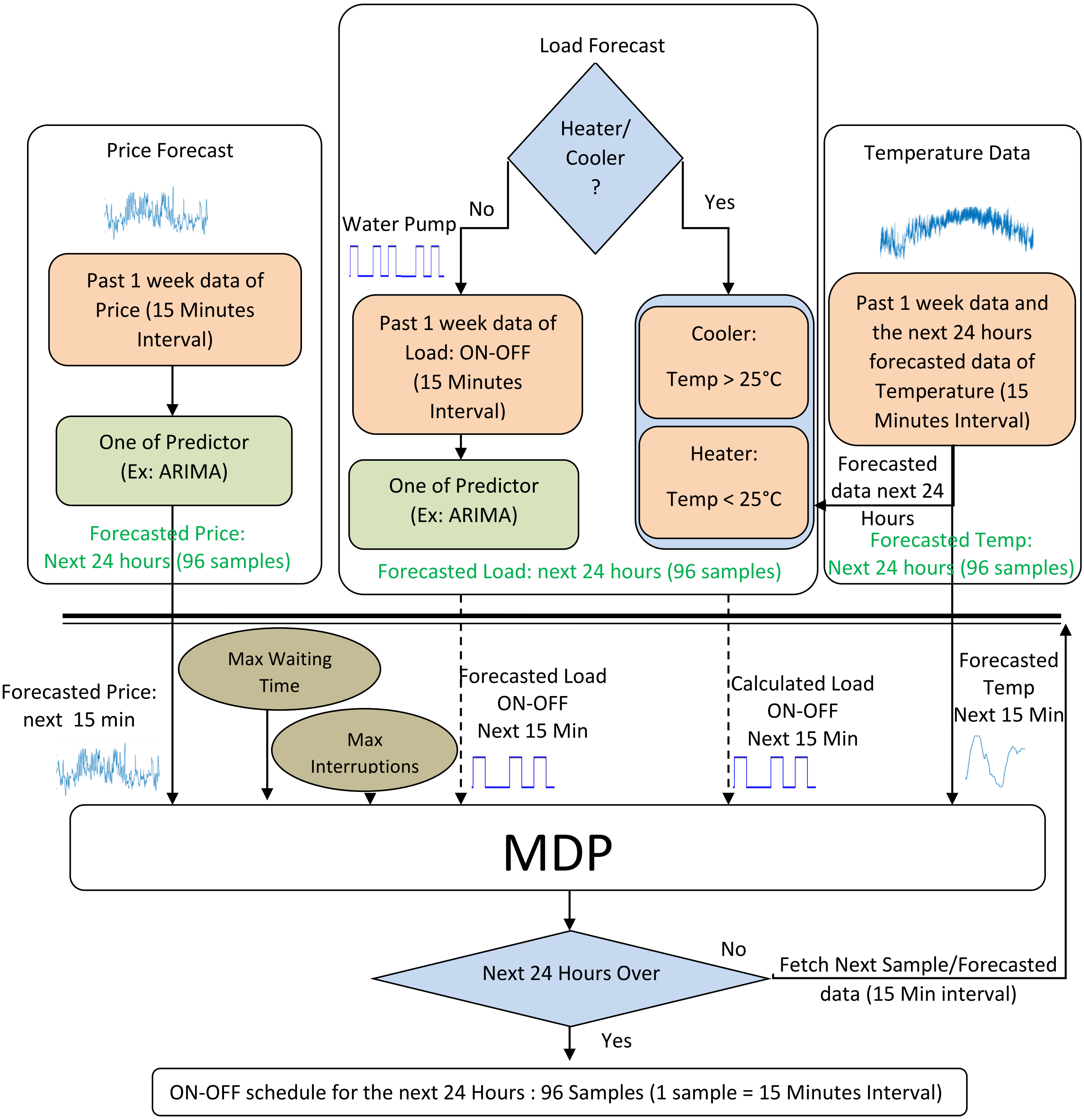


Figure 1. *An overview of the proposed DEMS algorithm*

3. Phase 1: Real time monitoring / Prediction phase

The motive of phase 1 is to predict the next 24 h device's switching patters ω_{Load} and the electricity price pattern ω_{Price} . This section states the following

1. The need of phase 1 or the prediction phase.
2. A brief description of the chosen predictor models.
3. The list of historic data or the inputs applied to each predictor model, in order to improve the prediction accuracy.
4. The considered test conditions and the simulation results of phase 1 in detail.
5. Conclusion.

3.1 Need for prediction

The algorithm is simulated and evaluated in MATLAB. The prediction phase or Phase 1 employs various predictor models to predict ω_{Load} and ω_{Price} information. In Markov theory, it is assumed that the past is independent of the future, only the present state is sufficient to estimate the next step. The main objective of phase 2 is to optimize the ON-OFF scheduling policy for the next 24 h, ω^*_{Load}, which gives 96 decision points [π^*_1, π^*_2, π^*_3,...,π^*_{96}], one for each 15 minute interval. To choose the optimal action at each decision point, π^*_x with $1 \leq x \leq 96$, the MDP requires the real-time electricity price, real-time temperature and device's switching state information at that specific instant of time as input. It is not possible to fetch these real time data in a simulated environment. Therefore, the future states comprising of the mentioned real-time information are predicted, e.g. a device's switching pattern is predicted with its samples present at 15 minutes interval, $\omega_{Load} = [\omega^{t=0}_{Load}, \omega^{t=15}_{Load}, \dots, \omega^{t=1440}_{Load}]$. This predicted information is applied as an input to the MDP at its corresponding time step.

3.2 Predictor models

The six chosen predictor models are listed in *Table 1*. From the listed predictor models, it is essential to identify one predictor model which provides a relatively best prediction accuracy A. Hence, with a similar training data set or a historic data $H = [H_{Load}, H_{Price}, \upsilon_p \mid \upsilon_i]$, the different predictor models are trained and corresponding responses ω are evaluated.

Table 1. *List of Predictor models*

Sl.No	Name
1	Linear Regression
2	Non Linear Regression
3	ARIMA
4	ANFIS
5	Neural Network Fitting
6	State Space Model

As discussed in SECTION II the prediction models are employed to predict 2 aspects:

1) For the prediction of the electricity price for the next 24 h, ω_{Price}.

2) For the prediction of the device switching pattern for the next 24 h, ω_{Load} .

1) For the prediction of the electricity price for the next 24 h, ω_{Price}: The algorithm predicts the electricity price information irrespective of the chosen device type (Water pump/Heater/Cooler). The outcome of the predictor model is a response variable[4], which is the electricity price for the next 24 h, ω_{Price}. The historic price data is received as an input to the predictor model with a sampling interval of 15 minutes $H_{Price} = [H_{Price}^{t=0}, H_{Price}^{t-15}, H_{Price}^{t-30}, \dots, H_{Price}^{t-n}]$, where H_{Price}^{t-n} is the n^{th} oldest sample of historical data. For that purpose, x predictor variables [$\upsilon_{p1}, \upsilon_{p2}, \dots, \upsilon_{px}$] and y intervention variables [$\upsilon_{i1}, \upsilon_{i2}, \dots, \upsilon_{iy}$] are considered to estimate the response variable ω accurately. The list of independent variables [$\upsilon_{p1}, \upsilon_{p2}, \dots, \upsilon_{px}, \upsilon_{i1}, \upsilon_{i2}, \dots, \upsilon_{iy}$] are to be carefully selected since a large number of predictor variables do not imply a prediction with a better accuracy. Any unnecessary information should be avoided. The list of selected predictor variables is $[\upsilon_i^{MoD}, \upsilon_p^{PWSTP}, \upsilon_p^{APWSDP}, \upsilon_i^{T}, \upsilon_i^{RH}, \upsilon_i^{DP}, \upsilon_i^{WS}]$.

a) *"Minutes Of Day"* υ_i^{MoD}:

This variable captures the minute at which the historic electricity price data H_{Price} is sampled. Since the sampling interval is 15 minutes, there exists 96 samples in a day, i.e. a day consists of 1440 minutes. Accordingly, 15,30,45,..., 1440 are the set of sample values for every 24 h. Since the considered *TD* is one week, the set of sample values is repeated for seven times. This variable is categorized as Interventional variable or Intervention indicator, since its value is neither an actual nor a derived value of H_{Price}. The υ_i^{MoD} is a set of n sample values, each sample derived from the corresponding sample values of $H_{Price.}$

$$\upsilon_i^{MoD} = [15, 30, \dots, 1440, 15, 30, \dots, 1440, \dots, 15, 30, \dots, 1440]$$

- b) *"Previous week Same Time Price"* υ_p^{PWSTP}:

In order to learn the electricity price patterns of same day of previous week, this predictor variable is included. This variable fetches the previous week's price data at the same day-of-week and the time-of-day as the present sampling time t. Term n denotes a week before the last day of *TD.*

$$Min = 24\ h \bullet 60.$$

[4] The input variables to the predictor model are called the *predictor variable* υ_p and the output of predictor model is called a *response variable*. Some of the predictor variables with an additional information that help estimate the response variables better are called the *intervention indicators* or the *intervention variables* υ_i.

$$\upsilon_p^{PWSTP} = [H_{Price}^{t-(15+(Min \bullet 7))}, H_{Price}^{t-(30+(Min \bullet 7))}, \dots, H_{Price}^{t-(1440+(Min \bullet 7))}, \\ \dots, \\ H_{Price}^{t-(15+(Min \bullet n))}, H_{Price}^{t-(30+(Min \bullet n))}, \dots, H_{Price}^{t-(1440+(Min \bullet n))}]$$

c) *"Average Previous Week Same Day Price"* υ_p^{APWSDP}:

In order to learn the average electricity price value μH_{Price} of the same day of the previous week, this predictor variable is included. This variable captures the computed average of the previous week's price data on the same day-of-week as the present sampling day. Term *n* denotes a week before the last day of *TD.*

Min = 24 h • 60.

$$\upsilon_p^{APWSDP} = [\mu H_{Price}^{t-(Min \bullet 7)}, \mu H_{Price}^{t-(Min \bullet 7)}, \dots, \mu H_{Price}^{t-(Min \bullet 7)}, \\ \dots, \\ \mu H_{Price}^{t-(Min \bullet n)}, \mu H_{Price}^{t-(Min \bullet n)}, \dots, \mu H_{Price}^{t-(Min \bullet n)}]$$

Where,

$$\mu H_{Price}^{t-(Min \bullet 7)} = Mean(H_{Price}^{t-(15+(Min \bullet 7))} + H_{Price}^{t-(30+(Min \bullet 7))} + \dots + H_{Price}^{t-(1440+(Min \bullet 7))})$$

d) *"Temperature"* υ_i^T, *"Relative humidity"* υ_i^{RH}, *"Dew point"* υ_i^{DP} *and "Wind speed"* υ_i^{WS}:1

These sets of variables, capture the temperature and the other environmental influences of the chosen city. To overcome the situations in which the H_{Price} or its derivatives, e.g. υ_p^{APWSDP}, alone may not be good enough to estimate ω_{Price}, these *y* intervention variables [$\upsilon_{i1}, \upsilon_{i2}, \dots, \upsilon_{iy}$] are included. The power generated in a power plant is influenced by the environmental uncertainty, e.g.: the power generated by a windmill is influenced by the wind speed. These variables are expected to aid in a better prediction and it is a necessary element of the proposed algorithm. For this reason, these variables are categorized as Interventional variables.

2) For the prediction of the device switching pattern for the next 24 h, ω_{Load}: The predictor models are employed to predict a flexible and a temperature independent device's switching pattern. The flexible devices which are operated at fixed interval, are the devices under consideration, e.g.: a Water pump. However, for the devices operating based on the temperature uncertainties, e.g.: a Cooler or a Heater, the device switching pattern is not predicted but it is determined by comparing the forecasted temperature value (for next 24 *h*) with 25° C. A set of *x* predictor variables [$\upsilon_{p1}, \upsilon_{p2}, \dots, \upsilon_{px}$] and *y* intervention variables [$\upsilon_{i1}, \upsilon_{i2}, \dots, \upsilon_{iy}$] are considered to estimate the ω_{Load} accurately. The

list of independent variables [$\upsilon_{p1}, \upsilon_{p2}, \ldots, \upsilon_{px}, \upsilon_{i1}, \upsilon_{i2}, \ldots, \upsilon_{iy}$] are to be carefully selected since a large number of predictor variables do not imply a prediction with a better accuracy A. Any unnecessary information should be avoided. The list of selected variables are [$\upsilon_i^{MoD}, \upsilon_i^{WiS}, \upsilon_i^{HoD}, \upsilon_i^{WDN}, \upsilon_p^{PWSTL}, \upsilon_p^{APWSDL}, \upsilon_i^{T}, \upsilon_i^{RH}, \upsilon_i^{DP}, \upsilon_i^{WS}$].

a) *"Minutes of Day"* υ_i^{MoD}:

This variable captures the minute at which the device's historical switching data H_{Load} is sampled. Since the sampling interval is 15 minutes, there exists 96 samples in a day, i.e. a day consists of 1440 minutes. Accordingly, 15,30,45,...,1440 are the set of sample values for every 24 h. Since the considered TD is one week, the set of sample values is repeated for seven times. This variable is categorized as an Interventional variable or an Intervention indicator, since its value is neither an actual nor a derived value of H_{Load}. The υ_i^{MoD} is a set of n sample values, each sample derived from the corresponding sample values of H_{Load}..

$$\upsilon_i^{MoD} = [15, 30, \ldots, 1440, 15, 30, \ldots, 1440, \ldots, 15, 30, \ldots, 1440]$$

b) *"Weekend is true"* υ_i^{WiS}:

This variable captures whether the day is considered a weekend or a weekday, i.e. Saturday and Sunday is indicated by the value *'1'* and weekdays from Monday to Friday are denoted by *'0'*. This predictor variable helps the predictor model to observe the weekend and weekday behaviors in H_{Load} and influence the ω_{Load} accordingly. The υ_i^{WiS} is set of n sample values, each corresponding to n sample values of H_{Load}. E.g. Consider a training data set, which is starting from Monday, the υ_i^{WiS} is given by

$$\upsilon_i^{WiS} = [0,0,0,0,0,1,10, \ldots, 1,0, \ldots, 1]$$

c) *"Weeks Day Number"* υ_i^{WDN}:

This variable captures the day's index in a week. Sunday is considered as the start of the week. Sunday is indicated by the value '*1*' and in sequence, Saturday is denoted by *'7'*. This allows the predictor model to learn the relationship between a particular day-of-week and the corresponding H_{Load}. The υ_i^{WDN} is a set of n sample values, each corresponding to n sample values of H_{Load}. E.g. Consider a training data set, which is starting from Monday, the υ_i^{WDN} is given by

$$\upsilon_i^{WDN} = [2,3,4, \ldots, 7,1,2, \ldots, 7,1]$$

d) *"Hour of Day"* υ_i^{HoD}:

This variable captures the hour at which the device's historical switching data H_{Load} is sampled. Since the sampling interval is 15 minutes, there exist four samples every hour. Accordingly, 1,1,1,1,2,2,2,2,3,...,24,24 are the set of sample values for 24 *h*. Since the considered *TD* is one week, the set of sample values is repeated for seven times. This variable is categorized as an Interventional variable or an intervention indicator, since its value is neither an actual nor a derived value of H_{Load}. The υ_i^{HoD}, is a set of *n* sample values, each corresponding to *n* sample values of H_{Load}.

$$\upsilon_i^{HoD} = [1,1,1,1,2,2,2,2,3, \dots ,24,1,1,1,1,2, \dots ,24,1,1,1,1, 2, \dots ,24]$$

e) *"Previous week Same Time Load"* υ_p^{PWSTL}:

In order to learn the device's switching patterns of the same day of the previous week, this predictor variable is included. This variable fetches the previous week's switching data at the same day-of-week and the time-of-day as the present sampling time *t*. Term *n* denotes a week before the last day of *TD.*

Min = 24 h • 60.

$$\upsilon_p^{PWSTL} = [H_{Load}^{t-(15+(Min\bullet 7))}, H_{Load}^{t-(30+(Min\bullet 7))}, \dots , H_{Load}^{t-(1440+(Min\bullet 7))}, \dots, H_{Load}^{t-(15+(Min\bullet n))}, H_{Load}^{t-(30+(Min\bullet n))}, \dots , H_{Load}^{t-(1440+(Min\bullet n))}]$$

f) *"Average Previous Week Same Day Load"* υ_p^{APWSDL}:

In order to learn the average number of switching μH_{Load}, of the same day of the previous week, this predictor variable is included. This variable captures the computed average of previous week's load data at the same day-of-week as the present sampling day. Term *n* denotes a week before the last day of *TD.*

Min = 24 h • 60.

$$\upsilon_p^{APWSDL} = [\mu H_{Load}^{t-(Min\bullet 7)}, \mu H_{Load}^{t-(Min\bullet 7)}, \dots , \mu H_{Load}^{t-(Min\bullet 7)}, \dots, \mu H_{Load}^{t-(Min\bullet n)}, \mu H_{Load}^{t-(Min\bullet n)}, \dots , \mu H_{Load}^{t-(Min\bullet n)}]$$

Where,

$$\mu H_{Load}^{t-(Min\bullet 7)} = Mean(H_{Load}^{t-(15+(Min\bullet 7))} + H_{Load}^{t-(30+(Min\bullet 7))} + \cdots + H_{Load}^{t-(1440+(Min\bullet 7))})$$

g) *"Temperature"* υ_i^{T} , *"Relative humidity"* υ_i^{RH}, *"Dew point"* υ_i^{DP} *and "Wind speed"* υ_i^{WS} :

These sets of variables, capture the temperature and the other environmental influences of the chosen city. To overcome the situations in which the H_{Load} or its derivatives, e. g. υ_p^{APWSDL}, alone may not be good enough to estimate ω_{Load}, these y intervention variables [$\upsilon_{i1}, \upsilon_{i2}, \ldots, \upsilon_{iy}$] are included. The power consumption is directly proportional to the environmental uncertainty, e.g.: The cold or warm weather influences the usage of the heater. These variables are expected to aid in a better prediction and it is a necessary element of the proposed algorithm. For this reason, these variables are categorized as an Interventional variable or an Intervention indicator.

Note: In this thesis, an attempt is made to predict the aggregated power consumption of the household as well. For the prediction of the aggregated power consumption for the next 24 h, $\omega_{Power\_Consumption}$, the list of selected predictor or intervention variables are same as that of the variables in device switching prediction and the term *load* refers to the aggregated power consumption [$\upsilon_i^{MoD}, \upsilon_i^{WiS}, \upsilon_i^{HoD}, \upsilon_i^{WDN}, \upsilon_p^{PWSTL}, \upsilon_p^{APWSDL}, \upsilon_i^{T}, \upsilon_i^{RH}$, $\upsilon_i^{DP}, \upsilon_i^{WS}$] instead of device's switching pattern.

A. Linear Regression

In statistical modeling, the regression analysis helps to estimate the relation between the variables. A regression is a set of technique that helps in understanding the relation between the dependent variables and the independent variables. The dependent variables are the response variable and the independent variables are the predictor variables and the intervention variables. Understanding the relationship between one independent variable and one dependent variable is called a *simple linear regression*. On the other hand, when more than one independent variable is involved, the process is called as *multiple linear regression* [9].

Let x_a, x_b, x_c,..., x_p be the p independent variables or the features influencing the dependent variable, y. It is assumed that the dependency y on x_a, x_b, x_c,..., x_p are linear. Let the assumed linear model be

$$y = \beta_0 + \beta_1 x + \epsilon \qquad [1]$$

Where,

β_0 : y intercept or coefficient
β_1 : slope or coefficient
ϵ : error

Let us consider an example of the advertising data on different media [14]. *Figure 22* shows the respective budgets and sales. Efforts are made to identify the synergy of sales among these advertising data.

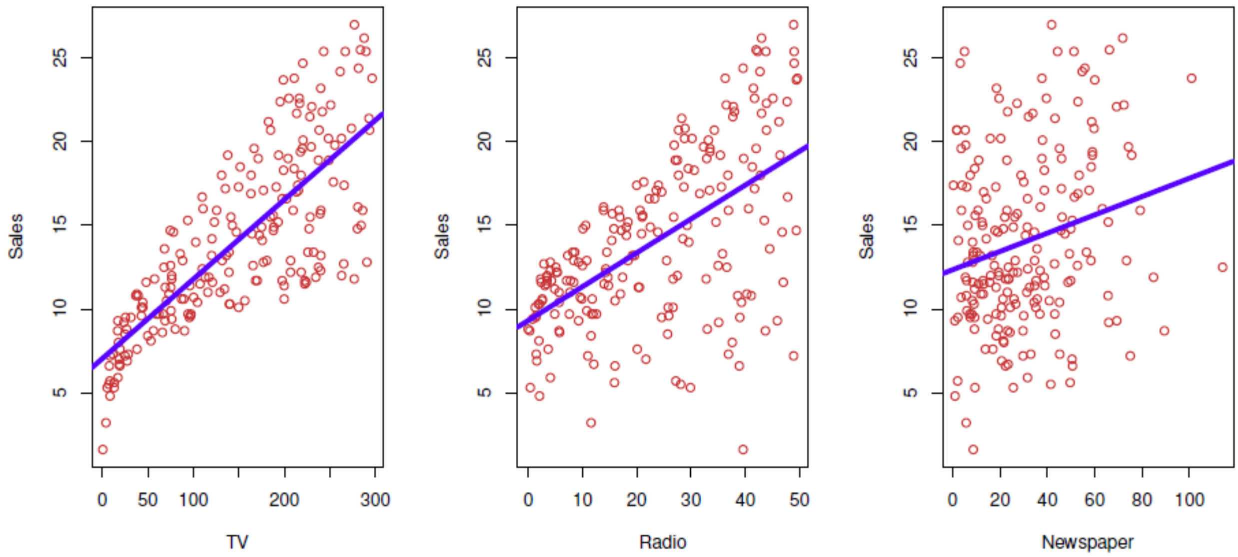


***Figure 2**. Advertising budgets and sales in different media [14].*

If some estimates of the coefficients (β_0 and β_1) are available beforehand, the future response values or the sales can be estimated by

$$\widehat{y_i} = \widehat{\beta_0} + \widehat{\beta_1} x_i + \epsilon \qquad [2]$$

Where the *hat* denotes the estimated value. As stated in [14], the response value y_i is based on x_i. The residual sum of error can be defined as.

$$RSS = e_1^2 + e_2^2 + \cdots e_n^2 \qquad [3]$$

Where e represent the i^{th} residual $e_i = y_i - \hat{y}_i$. RSS is considered as the *residual sum of errors.* The least square approach is applied to minimize the RSS in order to capture the best linear fit.

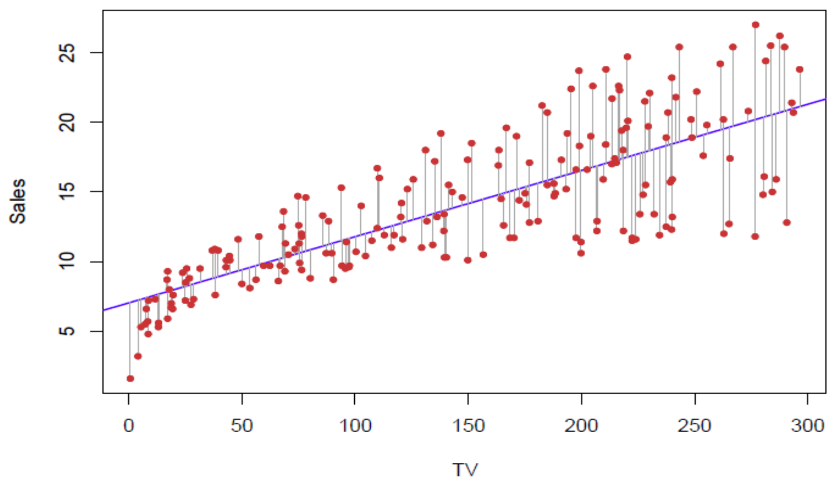


***Figure 3**. Least square fit for regression of sales on TV [14]*

In the proposed algorithm, the multiple linear regression analysis is adopted, i.e. analyzing the relationship between one dependent variable and multiple independent variables. As a first step, the predictor model is trained with the historical data set/data points, i.e. y is defined as a linear function of x_a, x_b, x_c,..., x_p and by adding some Gaussian noise ε_i [14]. The noise indicates the fact that, the estimated data points will not fit the model perfectly. β, β_1,..., β_p are the coefficient or the weighting factor.

$$y = \beta_a x_a + \beta_b x_b + \beta_c x_c + ... + \beta_p x_p + \varepsilon_i \quad [4]$$

The input data is represented as a $x \times p$ matrix, where each row corresponds to a data point and each column corresponds to a predictor variable or a feature. Since each output y_i is just a single number, the collection is represented as an n-element column vector y. As stated in [14], the linear model can be expressed as

$$y = X\beta + \varepsilon_i \;\; or \;\; y = f(X_i, \beta) + \varepsilon_i \quad [5]$$

Where β is a p-element vector of coefficients, and ε is an n-element vector, where each element ε_i, is normal with mean 0 and variance σ^2. This leads to the optimization problem [9,14,29].

$$y = \min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 \quad [6]$$

Where $\min_\beta$ refers to 'minimizing over β'. This refers to a least square linear regression problem. X_i refers to the row i of the matrix X. *Regress* is the linear fit model, used in this work.

B. Nonlinear Regression

A nonlinear regression is a type of regression analysis, in which the response variable is modeled by a function, which is a nonlinear combination of the model parameters and is dependent on the independent variables [9]. Unlike the linear model with $y = mx+c$ fit, the nonlinear model tries to fit the curvy data points using an exponential model or a power model. As stated in [9], a nonlinear regression model can be written as

$$Y_n = f(x_i, \theta) + \varepsilon_i \quad [7]$$

Where x_i corresponds to the predictor variables, f corresponds to the nonlinear function of p predictor variables. However, the generic nonlinear regression model equation is similar to the linear regression model, *Equation 5*, and the difference is specified by mentioning θ for the parameters in the nonlinear model. The term ε_i corresponds to a Gaussian noise. A nonlinear functions for several data sets are represented in a matrix form, $\eta(\theta)$, and $x_n = x \times p$ matrix, where each row corresponds to a data point and each column corresponds to a predictor variable or a feature. Since each output y_n is just a single number, the collection is represented as an n-element column vector y. As in [9], the nonlinear model can be expressed as

$$Y_n = \eta(\theta) + \varepsilon_i \quad [8]$$

This predictive model is present in ANOVA (analysis of variance) toolbox of MATLAB. *fitnlm*, the nonlinear fit model. This predictor model accepts 3 independent variables. Accordingly, the three predictor variables $x1$, $x2$ and $x3$ are.

1) For the prediction of the electricity price for the next 24 h
   a) *"Minutes Of Day"* υ_i^{MoD}
   b) *"Previous week Same Time Price"* υ_p^{PWSDP}
   c) *"Average Previous Week Same Day Price"* υ_p^{APWSDP}

2) For the prediction of the device switching patterns for the next 24 *h*.
   a) *"Minutes Of Day"* υ_i^{MoD}
   b) *"Previous week Same Time Load"* υ_p^{PWSDL}
   c) *"Average Previous Week Same Day Load"* υ_p^{APWSDL}

C. ARIMA

An Auto Regression Integrated Moving Average (ARIMA) model is one example of a time series regression model. A time series is a set of observations of a variable measured at regular intervals of time. The ARIMA model is used to analyze the *univariate time series* data *y(t)* [8,15,29]. The univariate time series data refer to a random variable's observation, captured at equal intervals *t*. The ARIMA model extrapolates the analyzed components of the time series data to estimate the future data, $y_{t+1}, y_{t+2},...,y_{t+n}$ [8,15]. The components in the time series data are the trends, seasonality, cyclical behavior and the other random behaviors. The analysis of data in linear and nonlinear regression analysis is via cross-correlation, i.e. the correlation is found between two or more different signals: between the independent variables $x_1,x_2,...,x_n$ and the response variable *y.* The ARIMA model analyzes the given data by finding the auto-correlation, i.e. the correlation of a signal's present sample y_t and with the various delayed copies of itself $y_{t-1}, y_{t-2},...,y_{t-n}$. ARIMA as a model is a combination of Autocorrelation methods (AR), differencing of non-stationary time series data (I) and Moving Averages (MA) [15].

Auto Regression / Autocorrelation model (AR model):

An AR model is one, in which y_t depends on its own past values $y_{t-1}, y_{t-2},...,y_{t-n}$.

$$y_t = f(y_{t-1}, y_{t-2},..., y_{t-n}, \varepsilon_t) \quad [9]$$

An AR model depending on *no* past value is denoted by AR (0). An AR model depending on *one* past value is denoted by AR (1). In generic, an AR model which depends on its *p* past value is denoted by AR (*p*) [15].

$$y = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \beta_3 y_{t-3} + ... + \beta_p y_{t-p} + \varepsilon_t \quad [10]$$

$$y = \sum_{i=0}^{p} \beta_i y_{i-1} + \varepsilon_t \quad [11]$$

Where,

β_a: Auto correlation at lags 1, 2,...,p.
ε_t : Residual error at time t.

Moving Averages model (MA model):

An MA model is one, in which y_t depends on its past random error value $\varepsilon_{t-1}, \varepsilon_{t-2},...,\varepsilon_{t-n}$. The error terms are assumed to be white noise with zero mean [15].

$$yt = f(\varepsilon_{t-1}, \varepsilon_{t-2},..,\varepsilon_{t-n}, \varepsilon_t) \qquad [12]$$

An MA model depending on *no* past value is denoted by MA (0). An MA model depending on *one* past value is denoted by MA (1). In generic, an MA model which depends on its *q* past value is denoted by MA (*q*) [15].

$$y = \phi_0\varepsilon_t + \phi_1\varepsilon_{t-1} + \phi_2\varepsilon_{t-2} + \phi_3\varepsilon_{t-3} + \dots + \phi_q\varepsilon_{t-q} \qquad [13]$$

$$y = \sum_{i=0}^{q} \phi_i \varepsilon_{i-1} \qquad [14]$$

Where,

ϕ_i: *Residual error at lags 1, 2,...,p.*

ε_t : *Residual error at time t.*

Auto Regression Moving Average Model (ARMA model):

A scenario in which a time series data may be represented as a mix of both AR and MA models is referred as ARMA model, i.e. the resulting model is of order *(p, q)* with *p* AR terms and *q* MA terms [15].

$$y = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \beta_3 y_{t-3} + \dots + \beta_p y_{t-p} + \varepsilon_t + \phi_1\varepsilon_{t-1} + \phi_2\varepsilon_{t-2} + \phi_3\varepsilon_{t-3} + \dots + \phi_q\varepsilon_{t-q} \qquad [15]$$

This form of the model assumes a time series data to be stationary, i.e. the joint probability of the data is time-invariant, thereby, this property implies that the parameters such as mean, variance and covariance are invariant of time, which in reality is rarely possible. The stationarized series are relatively easy to predict [8,15,25]. Typically the trends, cycle and random-walking contribute to the non-stationary behavior of the data set. A time series, which is non-stationary can be made stationary by differencing. The time series which has an inconsistent mean, variance and autocorrelation over time (season to season or period to period) and has these statistics changes between the periods and between the seasons constant is said to be *non-stationary* or *difference-stationary* [25]. This non-stationary behavior is removed by including a different order of differencing (e.g. 1^{st} order,2^{nd} order,...) into the model. These statistical differences obtained is untransformed to obtain the predicted original series [25]. A series which is differentiated once is denoted by *I(1)*. Similarly, if a time series is differentiated for *d* times, it is termed as *I(d)*. Collectively AR, MA and I form the triplet *(p,d,q)* i.e. ARIMA model [15].

In the econometrics toolbox of the MATLAB, the implementation of an Autoregressive Integrated Moving Average (ARIMA) model is present. The ARIMA model can be applied in cases where the data shows evidence of being non-stationary. ARIMA forecasts a value as a linear combination of historical data and past errors. The ARIMA modeling process/ ARIMA procedure is divided into 3 stages, i.e. the univariate time series estimation and forecasting are carried out by using a methodology such as Box-Jenkins (B-J) [8,15].

1) Identification stage. : The autocorrelation (ACF)[5] methods are employed to find the patterns from the past data.

2) Estimation and diagnostic stage: At this stage, the coefficients are estimated and the best fit model is diagnosed. E.g if ARMA is selected, tool diagnoses between AR (0)/AR (1)... and adopts the best among them.

3) Forecasting stage: For future time slots the time series components are extrapolated.

Chaotic Time Series prediction using ANFIS

The time series has properties such as cyclic, non-stationary, nonlinearity and chaotic [16]. Among these properties, the usual property found in reality is chaotic-time series [16]. These chaotic time series can be either discrete chaotic time series $f(x_n) = x_{n+1}$ or continuous chaotic time series $F(x(t)) = \frac{dx(t)}{dt}$ [17], where x is the random variable of a chaotic system, whose projection is extremely difficult to analyze and predict. In the chaotic time series prediction theory, the aim is to use the known past $x|_{(t-n)}$ and present scalar value $x|_t$ to predict the future value $x|_{(t+p)}$. There exists several intelligent systems such as neural network, fuzzy system and genetic algorithm etc., to predict the chaotic time series. In this thesis, the prediction accuracy of a hybrid system is investigated, i.e. of an Adaptive Neuro-Fuzzy Inference System or an Adaptive Network-based Fuzzy Inference System (ANFIS). ANFIS integrates both pattern learning property of the neural networks and the decision making / the reasoning property of the fuzzy logic principles. ANFIS is one of the special structures of the Neuro-Fuzzy Networks (FNNs). ANFIS is a model free estimator. ANFIS is composed of two parts, i.e. antecedents and conclusion. The input and output are connected by fuzzy rule base in network form [16]. A two input ANFIS structure is shown in *Figure 4.*

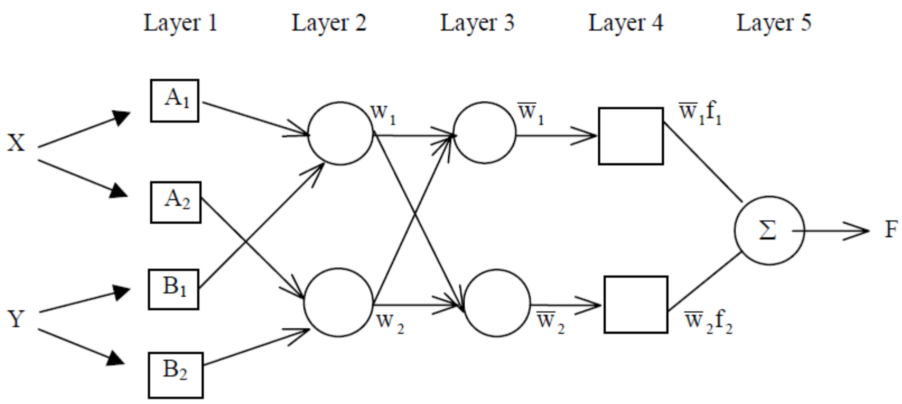


Figure 4. *Two input ANFIS structure [16]*

[5]ACF : Autocorrelation function refers to the observations of time series data that are related to each other.

The circular nodes in the ANFIS architecture indicate that they are fixed, whereas the square nodes indicate that they have parameters to be learnt [16]. During the training, in the *forward pass,* the input vector propagates through each layer and in *backward pass,* the errors are sent back. The ANFIS learns the rules and the membership function from the data [27]. The *rules* are if-then rules that are determined for quantitative reasoning. The *Membership Function (MF)* is a characteristic function of a subset *A*, that indicates the degree of membership of an elements *x* of any set *X,* in the fuzzy set *A,* and is denoted by $\mu_X(x)$, where $A \epsilon X$. The two input fuzzy rules are [27]

$$If\ x\ is\ A_1\ and\ y\ is\ B_1\ then\ f_1 = p_1x+q_1y+r_1 \quad [16]$$

$$If\ x\ is\ A_2\ and\ y\ is\ B_2\ then\ f_2 = p_2x+q_2y+r_2 \quad [17]$$

The output of each node in layer 1 is

$$O_{1,i} = \mu_{Ai}(x)\ \ for\ \ i = 1,2 \quad [18]$$

$$O_{1,i} = \mu_{Bi}(y)\ \ for\ \ i = 3,4 \quad [19]$$

Where,

$O_{1,i}(x)$: *The membership degree of the x and y*

The layer 2 computes the fuzzy AND of the antecedents.

$$O_{2,i} = w_i = \mu_{Ai}(x)\mu_{Bi}(y)\ ,\ i=1,2 \quad [20]$$

The layer 3 normalizes the MFs.

$$O_{3,\mathrm{i}} = \overline{w_i} = \frac{w_i}{w_1 + w_2}, i = 1,2 \quad [21]$$

The layer 4 computes the consequent of the fuzzy rule.

$$O_{4,i} = \overline{w_i}\, f_i = \overline{w_i}(p_i\, x + q_i\, y + r_i) \quad [22]$$

The layer 5 sums up the layer 4 output.

$$f = \overline{w_1}\, f_1 + \overline{w_2}\, f_2 \quad [23]$$

The *Imperialistic Competitive Algorithm (ICA)* and the *Least Square Estimation (LSE)* are used in ANFIS to train and determine the antecedent's and consequent's parameters (a_i, b_i and c_i), such that the ANFIS estimated output matches the original training data [16].

Prediction algorithm: A chaotic time series although exhibits stochasticity in general, in its phase-space reconstruction with appropriate embedding dimensioning D and time delay τ, a quasi-periodic attractor[6] in phase space can be obtained [16].

[6]A set of numerical values, towards which the system tend to evolve and which are quasi-periodic in behavior.

Step 1: Create a matrix with time series data. Let $x_1,x_2,...,x_N$ be the chaotic time series. An embedded phase vector $u(i)$ in the D dimensional phase space R^D is

$$u(i) = [x(i)\ x(i\text{-}\tau) \dots x(i\text{-}(D\text{-}1)\tau], \quad [24]$$

Where,

$$i \in [1+(D-1)\tau, N]$$

Step 2: Apply the time series matrix to the ANFIS structure. The matrix has the embedding phase vector as columns and chaotic time series as rows.

$$U = [u^T(i)\ u^T(i\text{-}1) \dots u^T(i+m)] \quad [25]$$

Where,

$$i \in [1+(D-1)\tau, N\text{-}m\text{-}k]$$

$$k : \textit{Number of prediction steps}$$

Step 3: The ANFIS is trained with the input matrix by hybrid-iterative procedures such as ICA and LSE algorithms [16]. When the error between the estimated and the original data is within the acceptable limits, the process is stopped.

An example of the most widely used chaotic time series is Mackey-Glass time series is shown in *Figure 5*, whose differential equation is given by

$$\frac{dx(t)}{dt} = \frac{0.2x(-t-\tau)}{1+x^{10}+(t-\tau)} \quad [26]$$

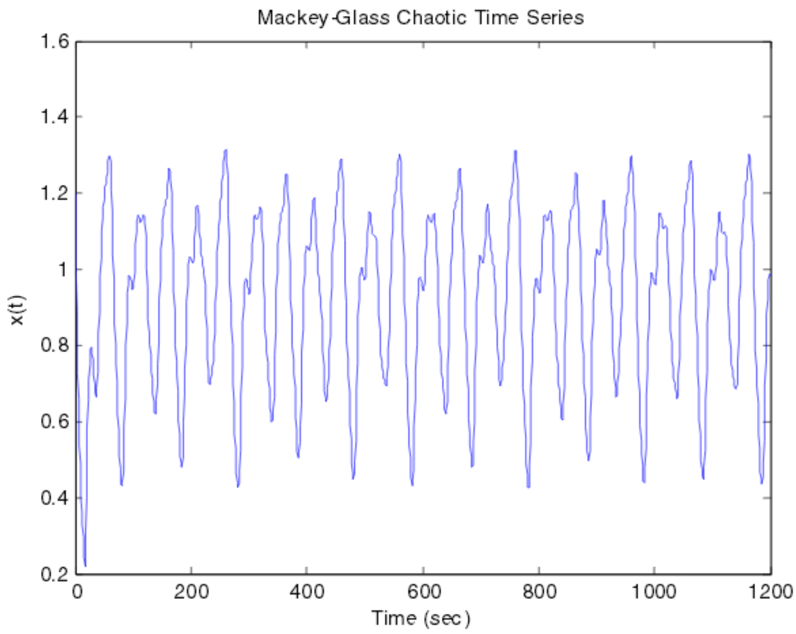


***Figure 5**. Chaotic time series [28]*

The future time series values $x|_{(t+p)}$ is predicted by considering the present and the past chaotic time series, i.e. the training data set [18]. In the proposed algorithm, the assumption is $\tau = 6$ and $D = 4$.

$$[x(t-18), x(t-12), x(t-6), x(t) ; x(t+P)] \quad [27]$$

These datasets could be applied as inputs to the pattern recognition and the decision making algorithm such as ANFIS for predicting *x(t+p)* with a high accuracy. MATLAB provides the ANFIS estimator in the standard *fuzzy logic toolbox*. *Genfis1* on receiving the training data outputs a fuzzy interface system, then *evalfis* perform fuzzy inference calculations to predict the next state space *x(t+P)*.

D. Neural Network fitting

Although the soft computing has evolved in the recent years, making a machine think like a human has been an unprecedented challenge. To resolve the computing gap that existed to mimic the human brain, several researches were carried out to emulate the neural activity of the human brain, thereby, the human prowess. These researches have led to the field of Artificial Neural Networks (ANNs) [7,11,19].

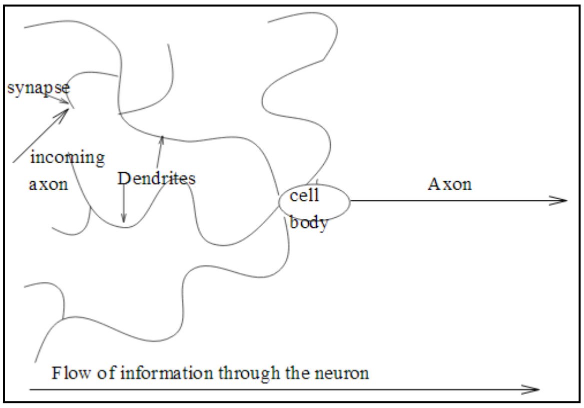


Figure 6a. *A simplified neuron [19]*

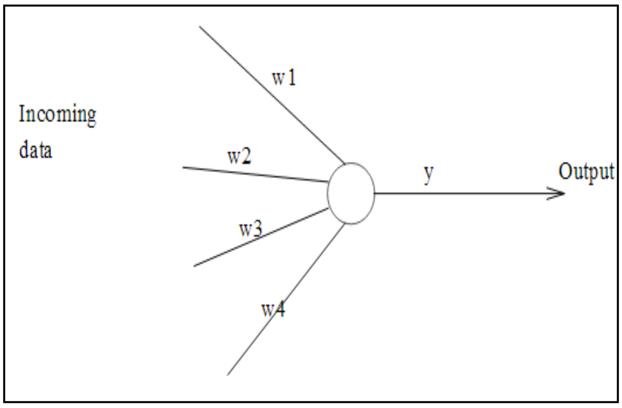


Figure 6b.*Artifical neuron [19]*

The *Figure 6a.* shows the simplified neural network. If the electrical impulses in the cell body are greater than a certain threshold, a signal/information is sent to another neuron via dendrites and synapses. The efficiency of synapses differs as per the lifetime of neurons. The *Figure 6b.* shows the artificial neuron. The ANNs are a collection of several such artificial neurons. The inputs are denoted by *x.* The synapses are modeled by weight vector *w.* Therefore the output of this neuron is

$$y = f(\textstyle\sum_i w_i x_i) = f(w^T x) \quad [28]$$

The ANN is represented in *Figure 7.* The underlying neural network consists of several interlaced hidden neuron layers. Based on changes in the environment or in the inputs, the neural network trains/learns and adapts to the change by changing the weights *w* in the network.

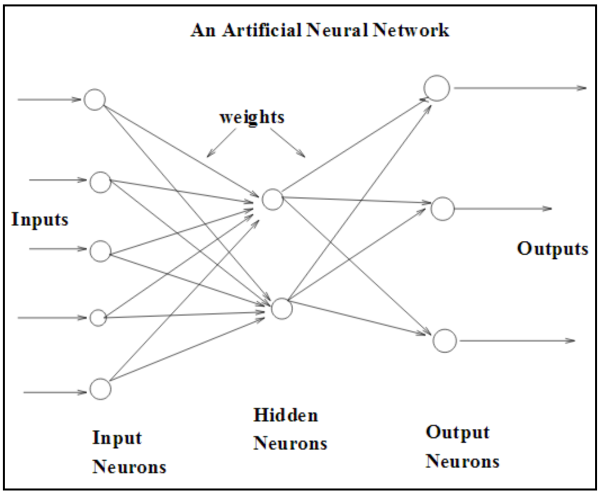
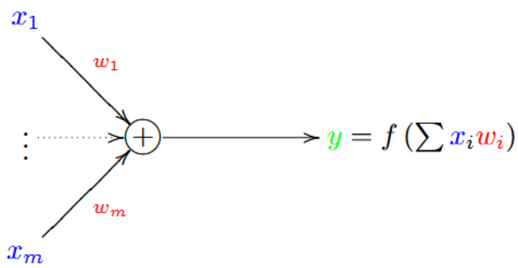


Figure 7. *ANN [19]*

A model of single neuron [26]



1. Let $x_1, x_2, \ldots, x_m$ be the m input values or synapses. They do not compute anything but pass the values to processing nodes.
2. Let $w_1, w_2, \ldots, w_m$ be the weights corresponding to each synapse.
3. The input values are multiplied by their weights and summed.

$$v = w_1 x_1 + w_2 x_2 + \ldots + w_m x_m = \sum_{i-1}^{m} w_i x_i \qquad [29]$$

4. The output function is weighted sum. $y = f(v)$. This function is called *activation function*.

$$f(v) = a + v = a + \sum_{i-1}^{m} w_i x_i \qquad [30]$$

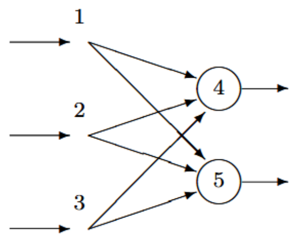
Where, a is called the *bias*, i.e. bias a is the intercept and the weights, $w_1, \ldots, w_m$ is the slope. A neural network may have hidden nodes, which are not connected to the environment directly. These nodes are organized as layers to form hidden layers.

5. A cost function is used to compute the error between the true output and the estimated output. One such cost function is linear mean square regression.

$$c(y, f(x)) = | y - f(x)|^2 \quad [31]$$

Feed-Forward Neural Networks

A collection of neurons connected together in a network can be represented by a directed graph [26]



1. Nodes represent the neurons, and arrows represent the links between them.
2. Each node has its number, and a link connecting two nodes will have a pair of number (e.g. (1,4)).
3. Networks without a feedback loop are called a feed forward network.

Training

The process of fine-tuning the weights w_{ij} of the network to match the desired output is called *training* [26]. The training algorithm used in this thesis is Supervised learning. In Supervised learning, the network is supplied with the input data and the expected output. The outline of the supervised algorithm is

1. Set all the weights w_{ij} to random values.
2. Feed the network with input data $x_1, x_2, \ldots, x_m$.
3. Compute the network output $f(v)$.
4. Change the weight $w_{11}, w_{12}, \ldots, w_{mn}$ of the nodes.
5. Repeat from step 2 to 4 until the computed cost or error is small.

MATLAB provides a standard toolbox for prediction with a multilayered feed-forward artificial neural network. NN is more of a black box capable of learning hidden dependencies, which is not possible to explicitly represent with any equation based model such as regression analysis. Available data are divided into three sets: learning set, validating set and testing set. These sets could be overlapping and do not have to be continuous. The learning set is a sequence that is shown to the neural network during the learning phase. The network is adapted to it to achieve required outputs (in other words, weights in the network are changed based on this set). The difference to the required output is measured using the validating set and this difference is used to validate whether the learning of the network can be finished. The last set, testing set, is then used to test whether the network is able to work also on the data that were not used in the previous process [11].

E. State Space Model

The State Space Models (SSM) along with Kalman filters serves as a good tool to analyze and forecast the time series data. This section describes the linear Gaussian state space model or the dynamic linear model (DLM). The state space model comprises of two aspects, i.e. the state process and the observations or output [28]. These *state process* or *unobserved signal* x_t, given the input $u_t = [u_1,...,u_n]$, in state space model, are considered to be hidden or latent and are Markovian [28], i.e. it is assumed that the future and past states are independent and conditional on the present state. The *observations*, y_t are independent and dependence among these observations are formulated by linear Gaussian state space model [28].

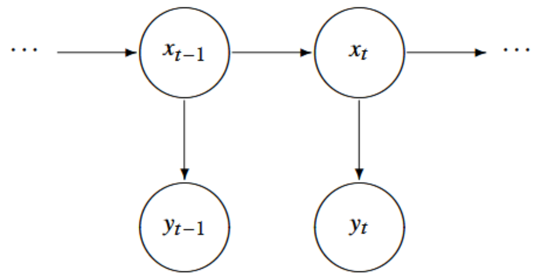


Figure 8. *State space model [28]*

The linear Gaussian state space model has a first order state equation, which is p-dimensional auto regressive vector [28].

$$x_t = \phi x_{t-1} + w_t,\ w_t \sim N(0,Q) \qquad [32]$$

where,

w_t : The process noise of dimension px1 with '0' mean and covariance Q.

State vector x_t is not observed directly but it's linearly transformed and noise added version is given by an observation equation [28], which is q-dimensional (larger or smaller than state vector)

$$y_t = A_t x_t + v_t,\ v_t \sim N(0,R) \qquad [33]$$

where,

A_t : A qxp measurement or observation matrix.
v_t :The measurement noise of dimension qx1 with '0' mean and covariance R.

Let u_t be the input vector *rx1*, the mathematical model is given by

$$x_t = \phi x_{t-1} + \Upsilon u_t + w_t \qquad [34]$$

$$y_t = A_t x_t + \Gamma u_t + v_t \qquad [35]$$

Where, Υ is *pxr* and Γ is *qxr* matrix and either of them can be zero matrix. The motive of the state space model is to estimate the hidden unobserved data x_t, given the observed data y_t. After defining the mathematical model, the x_t is estimated and fine-tuned by *Kalman filter,* which is denoted by x_t^s [28]. The x_t^s is the linear filtered version of y_t.

$$x_t^t = \sum_{S=1}^{t} B_s\, y_s \qquad [36]$$

Where,

B_s : Suitably chosen smoothening matrix of order p x q .

Several related time series that has dynamic interactions can be forecasted using the STATESPACE procedure. MATLAB provides the feature of estimation using state space models under the linear model identification of the system identification toolbox. State space models support the data such as time or frequency domain data, real or complex data or even single or multiple output.

3.3 Prediction of power consumption and electricity price for household (aggregated power consumption)

The historic data *H= [$H_{Power\_Consumption}$, H_{Price}, υ_p | υ_i]* is provided to train each predictor model. The *TD* also plays a vital role as described in SECTION II. As a first step, the type of load considered is the aggregated power consumption of the house. The aggregated power consumption at specific instant of time t is the summed up energy consumption of the household devices (e.g.: Water pump P_{WP}^t, heater P_{Heater}^t, cooler P_{Cooler}^t, TV P_{TV}^t etc.) which are operating at specific point in time t, $P_{Total}^t = P_{WP}^t + P_{Heater}^t + P_{Cooler}^t + \cdots + P_{TV}^t$. As a first task, the centralized energy management system *CEMS* was simulated, since it is assumed that, there exists one monitoring unit per house for all household appliances.

Step 1: All aforementioned 6 predictor models are employed.

Step 2: Each predictor model is trained with two year's worth (*TD = two years)* of *H = [$H_{Power\_Consumption}$, H_{Price}, υ_p | υ_i]*. The predictor models are employed to predict ω_{Price} and $\omega_{Agg\_Power\_Consumption}$. However, the temperature forecast υ_p^T is directly fetched from the online weather forecasters. As an Initial step, a set of predictor variables are chosen and the corresponding power consumption and electricity price prediction accuracy is observed. In successive steps, by trial and error the different predictor variables are chosen such that such a way that it improves the prediction accuracy. Accordingly, the predictor variables and the intervention variables chosen for all but except the nonlinear regression predictor model are

1) For the prediction of the electricity price for the next 24 h, ω_{Price}:
   a) *"Minutes Of Day"* υ_i^{MoD}
   b) *"Previous week Same Time Price"* υ_p^{PWSDP}
   c) *"Average Previous Week Same Day Price"* υ_p^{APWSDP}
   d) *"Temperature"* υ_i^T, *"Relative humidity"* υ_i^{RH}, *"Dew point"* υ_i^{DP} *and "Wind speed"* υ_i^{WS}

2) For the prediction of the aggregated power consumption for the next 24 *h*, $\omega_{Agg\_Power\_Consumption}$:
   a) *"Minutes Of Day"* υ_i^{MoD}
   b) *"Previous week Same Time Load"* υ_p^{PWSDL}
   c) *"Average Previous Week Same Day Load"* υ_p^{APWSDL}
   d) *"Weekend is true"* υ_i^{WiS}
   e) *"Weeks Day Number"* υ_i^{WDN}
   f) *"Hour Of Day"* υ_i^{HoD}
   g) *"Temperature"* υ_i^{T}, *"Relative humidity"* υ_i^{RH}, *"Dew point"* υ_i^{DP} *and "Wind speed"* υ_i^{WS}

In the nonlinear regression model the count of independent variables $\upsilon_p \mid \upsilon_i$ that can be applied as input is 3. The independent variables $\upsilon_p \mid \upsilon_i$ that result in an improved *A*, are chosen by trial and error, i.e. by monitoring the curve fit of the response variable *y* with occurred data of next 24 *h* and checking the MAPE. The chosen predictor variables and intervention variables are

1) For the prediction of the electricity price for the next 24 h, ω_{Price}:
   a) *"Minutes Of Day"* υ_i^{MoD}
   b) *"Previous week Same Time Price"* υ_p^{PWSDP}
   c) *"Average Previous Week Same Day Price"* υ_p^{APWSDP}

2) For the prediction of the aggregated power consumption for the next 24 *h*, $\omega_{Agg\_Power\_Consumption}$:
   a) *"Minutes Of Day"* υ_i^{MoD}
   b) *"Previous week Same Time Load"* υ_p^{PWSDL}
   c) *"Average Previous Week Same Day Load"* υ_p^{APWSDL}

Step 3: Once the predictor models are trained with the corresponding independent variables for the prediction of the device's switching or electricity price, they predict $\omega_{Agg\_Power\_Consumption}$ or ω_{Price}, respectively. The sampling interval is 15 minutes. Therefore, the historical time series data *H* and the forecasted 24 *h* data ω are available at 15 minutes sampling intervals, i.e. making a total of 96 sample points in every 24*h*.

$$\omega_{Agg\_Load} = [\omega^{t}_{Agg\_Power\_Consumption}, \omega^{t+15}_{Agg\_Power\_Consumption}, \dots, \omega^{t+1440}_{Agg\_Power\_Consumption}]$$

$$\omega_{Price} = [\omega^{t}_{Price}, \omega^{t+15}_{Price}, \omega^{t+30}_{Price}, \dots, \omega^{t+1440}_{Price}]$$

Step 4: Once the prediction is completed, the prediction accuracy *A* has to be evaluated. *A = (1-MAPE)•100*, where MAPE is expressed in percentage and is given by *Equation 37.*

$$M = \frac{100}{\mathrm{n}} \sum_{\mathrm{k}=0}^{\mathrm{n}} \left| \frac{A_t - F_t}{A_t} \right| \qquad [37]$$

Where,

A_t : *Actual value.*
F_t : *Forecast value.*
M : *Prediction accuracy.*

3.4 Results

The test condition for the effective evaluation of the predictors models are as follows

I. The predictor models adopted in this work are
   a) ARIMA
   b) Nonlinear regression
   c) Linear regression
   d) ANFIS
   e) Neural network
   f) State space model

II. Place of consideration: A small two-storied office in a city, Anchorage is considered, which is having a power consumption of 60 kWh - 80 kWh in weekdays and 30 kWh - 50 kWh in weekends.

III. Duration of training data *TD*: Two year's worth of $H_{Agg\_Power\_Consumption}$, temperature $H_{Temperature}$ and price H_{Price} information's were applied to the predictor models as inputs, i.e. as a set of *x* predictor variables and *y* interventional variables [$\upsilon_{p1}, \upsilon_{p2},\ldots,\upsilon_{px}, \upsilon_{i1}, \upsilon_{i2},\ldots,\upsilon_{iy}$]. The considered training year is 2015 and 2016 (start date: Jan 2nd 2015 and end date: Jan 01st, 2017) and the considered forecast date is Jan 2nd of 2017. Jan 2nd 2017, was selected as the forecast date, since Jan 01 is a holiday. However, the predictor models would have predicted successfully for Jan 01st as well, if an additional predictor variable addressing the special holidays, υ_p^{SH}, in a year were included in the design.

IV. Sampling interval: As per the requirement, the device is controlled and monitored at an interval of 15 minutes, therefore, the $H = [H_{Power\_Consumption}, H_{Price}, \upsilon_p \mid \upsilon_i]$ data are acquired at a rate of 15 minute interval, giving rise to 96 samples each day.

$$\omega_{Price/Agg\_Power\_Consumption} = [\omega^{t}_{Price/Agg\_Power\_Consumption}, \omega^{t+15}_{Price/Agg\_Power\_Consumption}, \ldots, \omega^{t+1440}_{Price/Agg\_Power\_Consumption}]$$

Output:

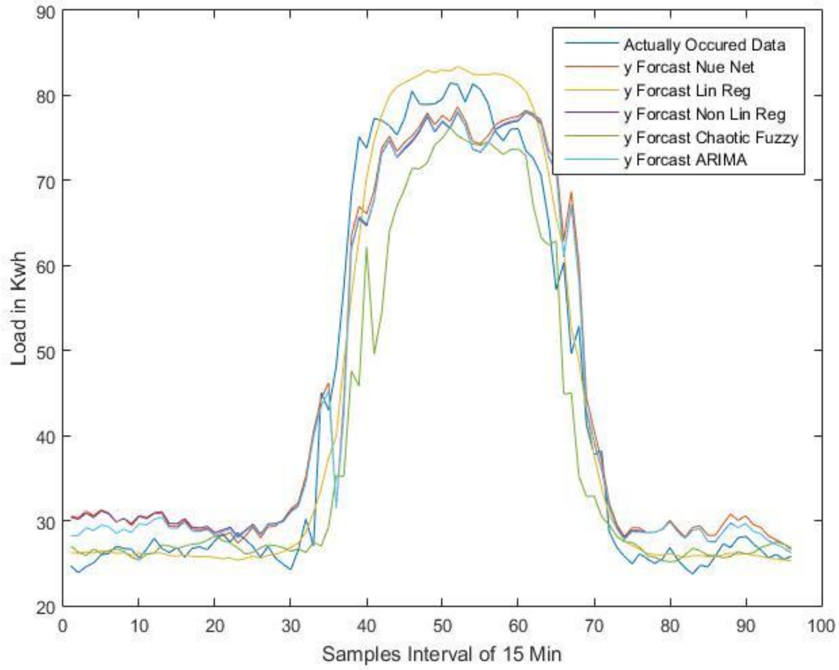
Next 24 h aggregated power consumption and electricity price i.e. $\omega_{Agg\_Power\_Consumption}$ and ω_{Price} is forecasted at 15 minutes sampling interval. For both the prediction aspects, *Table 2* shows the prediction error of each predictor model. The considered training duration is two years and the chosen forecast date is Jan 2nd, 2017. In case of the power consumption prediction, the neural network is predicting relatively good with less prediction error. In case of electricity price prediction, ANFIS model is predicting relatively good with less prediction error. The SSM is consuming approximately 9 hours to estimate the future states and has relatively poor accuracy, hence discarded from the work.

Table 2 *MAPE values computed for the aggregated power consumption prediction and electricity price prediction for different predictor models*

Sl.No	Name	MAPE value for Power consumption prediction	MAPE value for Price prediction
1	Linear regression	10.97	14.83
2	Non Linear regression	11.35	30.07
3	ARIMA	10.05	14.02
4	ANFIS	8.60	13.60
5	Neural Network fitting	5.82	15.15
6	State Space Model	22.10 SSM takes approx. 9 h to predict the next 24 h power consumption pattern	Excluded, since SSM takes approx. 9 h to predict the next 24 h power consumption pattern

Figure 9 shows the forecasted $\omega_{Agg\_Power\_Consumption}$ and ω_{Price} curves (predicted power consumption in kWh versus sampling time in the next 24 h, electricity price in cents/Watts versus next 24 h respectively). All the six predictor outputs, e.g.: the power consumptions $[\omega_{Agg_{Power_{Consumption}}}|_{ARIMA}, \ldots, \omega_{Agg\_Power\_Consumption}|_{NN}]$ are plotted on the same axes to give a better overview. The power consumption data are plotted on the axes, samples (96 intervals) versus electricity consumption in kWh. The electricity price data are plotted on the axes, price in cents versus hours.

Since the main motive is to monitor and control the switching pattern at the device level instead of the aggregated power consumption data, the individual device data are considered and proceeded in this work. Accordingly, a Water pump, a Heater and a Cooler are considered.



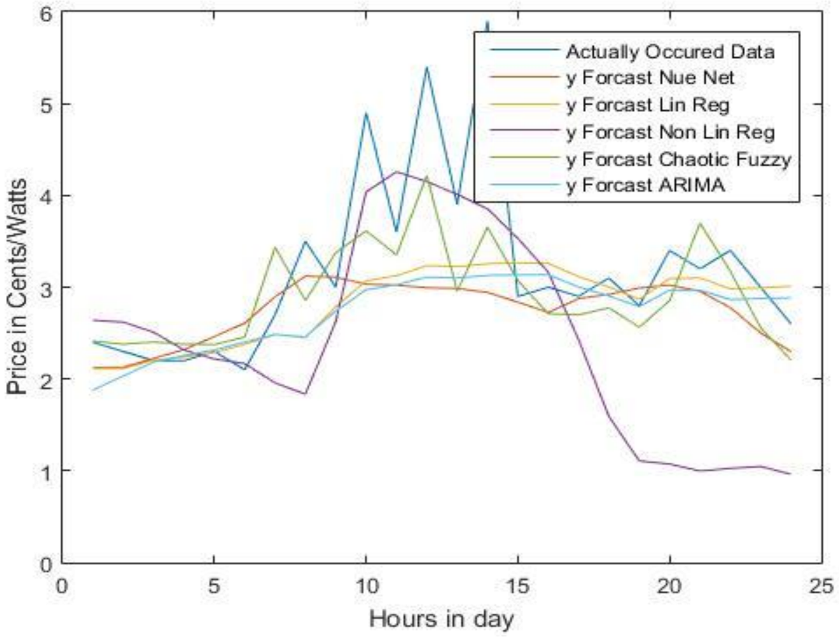


Figure 9. *The aggregated power consumption prediction and electricity price prediction for different predictor models*

3.5 Prediction of device switching pattern and electricity price for Single Device

In this work, the centralized and the decentralized energy management approaches are seen from the household appliance perspective. Hence, in DEMS the individual device is considered to be the end entity. Accordingly, in this thesis a device's switching pattern is considered rather than the aggregated household power consumption. The chosen flexible devices are a Water pumps, a Heater and a cooler.

Step 1: All aforementioned 6 predictor models are employed.

Step 2: Each predictor model is trained with one week (*TD=One week)* of historical data $H = [H_{Load} + H_{Price} + \upsilon_p \mid \upsilon_i]$. This time window would help the predictor models to learn the device's switching or electricity price dynamics better compared to a long term *TD*. The predictor models are applied to predict ω_{Price} and ω_{Load}. However, the temperature forecast υ_p^T is taken from the online weather forecasters. The Water pump's switching pattern is forecasted, the temperature dependent heater and cooler switching patterns are determined from the temperature forecast, i.e. if the temperature is higher than the nominal ambient temperature (25°C), the Heater is considered OFF and the cooler is considered ON and if the temperature is lower than the aforementioned value, then vice versa. The independent variables are chosen in such a way, that it improves the prediction accuracy A. Accordingly, the predictor variables and the intervention variables chosen are the same as that of the list mentioned in *Section 3.3*, however the present case is for single device switching pattern.

Step 3: Once the predictor models predicts ω_{Load} and ω_{Price}. The sampling interval is 15 minutes. Therefore, the time series historical data H and the forecasted data ω are available at 15 minutes sampling interval, i.e. making a total of 96 sample points in every 24 h.

$$\omega_{Load} = [\omega_{Load}^{t}, \omega_{Load}^{t+15}, \omega_{Load}^{t+30}, \dots, \omega_{Load}^{t+1440}]$$

$$\omega_{Price} = [\omega_{Price}^{t}, \omega_{Price}^{t+15}, \omega_{Price}^{t+30}, \dots, \omega_{Price}^{t+1440}]$$

Step 4: Once the ω_{Load} and ω_{Price} has been forecasted, the prediction accuracy has to be evaluated. The Mean absolute percentage error (MAPE) or the Mean absolute percentage deviation (MAPD), serves as a good measuring tool for prediction accuracy, which is expressed in percentage and is given by *Equation 37.*

Step 5: The output of the best predictor models is fed as input to phase 2. In phase2, the forecasted switching pattern ω_{Load} is optimized to ω_{Load}^{*} using MDP. This will be explained in detail in section IV.

3.6 Results

The test condition for the effective evaluation of the predictor models are

I. The predictor models adopted in this work are

a) ARIMA

b) Nonlinear regression

c) Linear regression

d) ANFIS

e) Neural network

f) State space models

II. Place of consideration: To capture the regional influence on the algorithm, the devices in 3 different cities or locations are considered. The considered locations are ANCHORAGE, PALMADELE and LITTLE ROCK of USA. At each location Water pump's next 24 *h* switching pattern is predicted and heater/cooler's next 24 *h* switching pattern is determined according to the ambient temperature. The process of evaluation is explained in detail in SECTION IV.

III. Duration of training data *TD*: One week of device's historical switching pattern H_{Load}, temperature $H_{Temperature}$ and price H_{Price} of a chosen location were applied to the predictor models as input, i.e. as a set of x predictor variables and y interventional variables $[\upsilon_{p1},\upsilon_{p2},\ldots,\upsilon_{px},\upsilon_{i1},\upsilon_{i2},\ldots,\upsilon_{iy}]$. Irrespective of the season of the year and chosen location, the predicted device switching pattern is having a better *A,* when trained with one week's historical data compared to two years of historic data.

IV. Sampling interval: As per the requirement, the device is controlled and monitored at a rate of 15 minutes, i.e. $H = [H_{Load}, H_{Price}, \upsilon_p \mid \upsilon_i]$ data are acquired at a rate of 15 minutes, giving rise to 96 samples each day. Similarly, the expected prediction output samples are also spaced 15 minutes apart.

$$\omega_{Price/Load} = [\omega^{t}_{Price/Load}, \omega^{t+15}_{Price/Load}, \omega^{t+30}_{Price/Load}, \ldots, \omega^{t+1440}_{Price/Load}]$$

Output:

Next 24 *h* device ON-OFF pattern for Water pump for the chosen location ω_{Load} and electricity price ω_{Price} is forecasted at 15 minute interval.

3.7 Predicted and determined device switching patterns

I. Water pump's predicted ON-OFF patterns

Figure 10 shows the $\omega_{Load}|_{ARIMA}$and the prediction error curves. As seen, the prediction error plot is of the order 10^{-15}, which is almost negligible. It can be inferred that, the prediction accuracy is nearly ≈100%. However, the precise mean accuracy results w.r.t an entire year is provided in SECTION IV. The prediction error is calculated by

Prediction error = Occurred data - Predicted data

II. Predicted Electricity Price

Figure 11 shows the $\omega_{Price}|_{ARIMA}$ (Device: Any Device, Location: chosen location - ANCHORAGE) and its prediction error curves. The price prediction accuracy of ARIMA is approximately 40%. However, the precise mean accuracy results w.r.t an entire year is provided in SECTION IV.

III. Calculated Heater ON-OFF pattern

Figure 12 shows the heater's switching and temperature curves. The device will be ON if the temperature is greater than 77° Fahrenheit or else will remain in OFF state.
Note: 77° Fahrenheit = 25° Celsius

IV. Calculated Cooler ON-OFF pattern

Figure 13 shows the cooler's switching and temperature curve. The device will be ON if the temperature is lesser than 77° Fahrenheit or else will remain in OFF state.

Note: 77° Fahrenheit = 25° Celsius

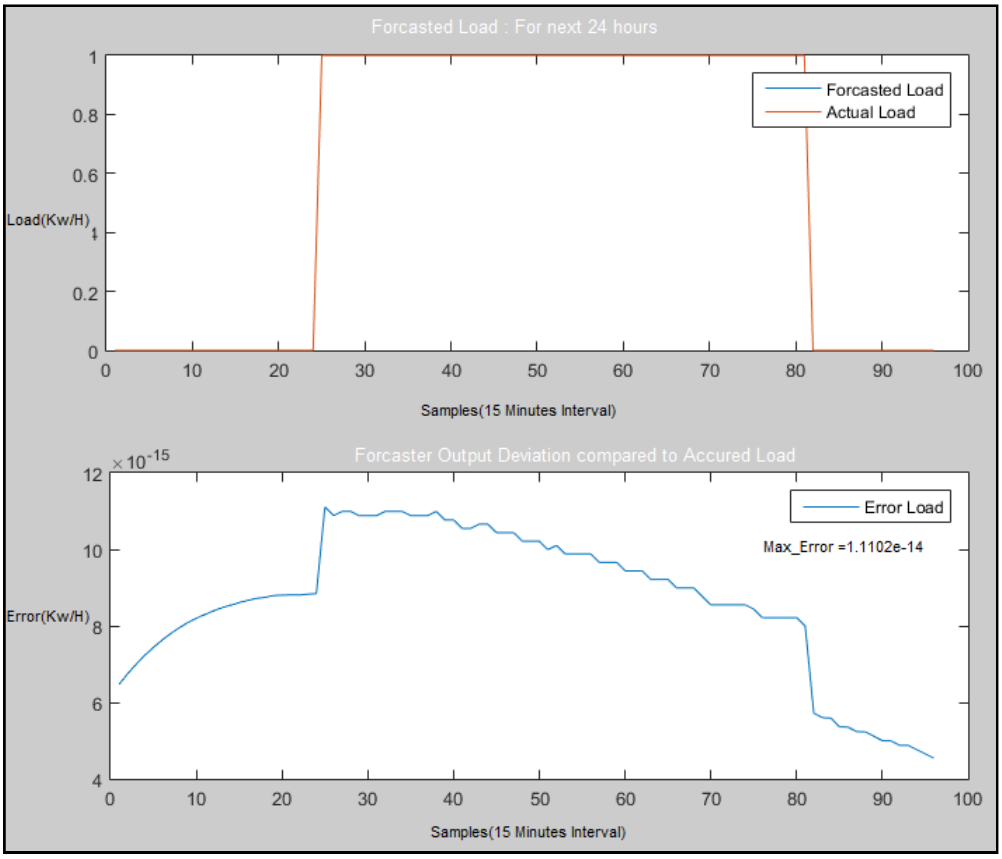


Figure 10. *Water pump's switching prediction and prediction error curves for the next 24 h*

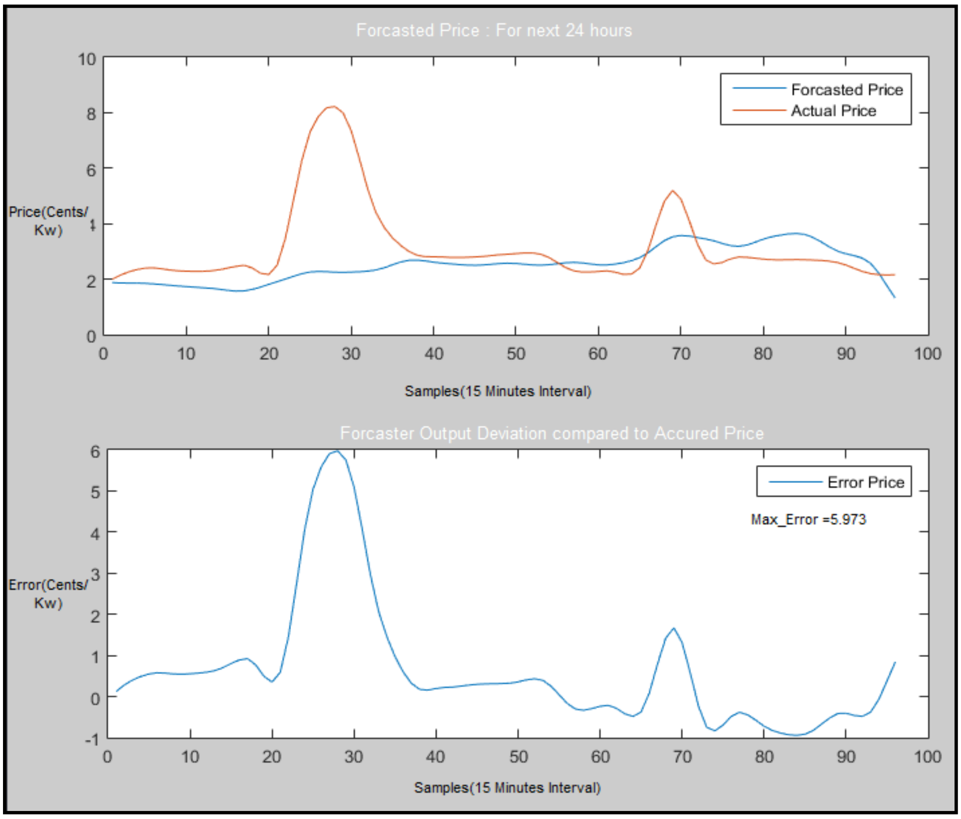


Figure 11*. Price predictions curve (Device: Any device, City: Anchorage, Predictor model: ARIMA) and its prediction error curve for the next 24 h*

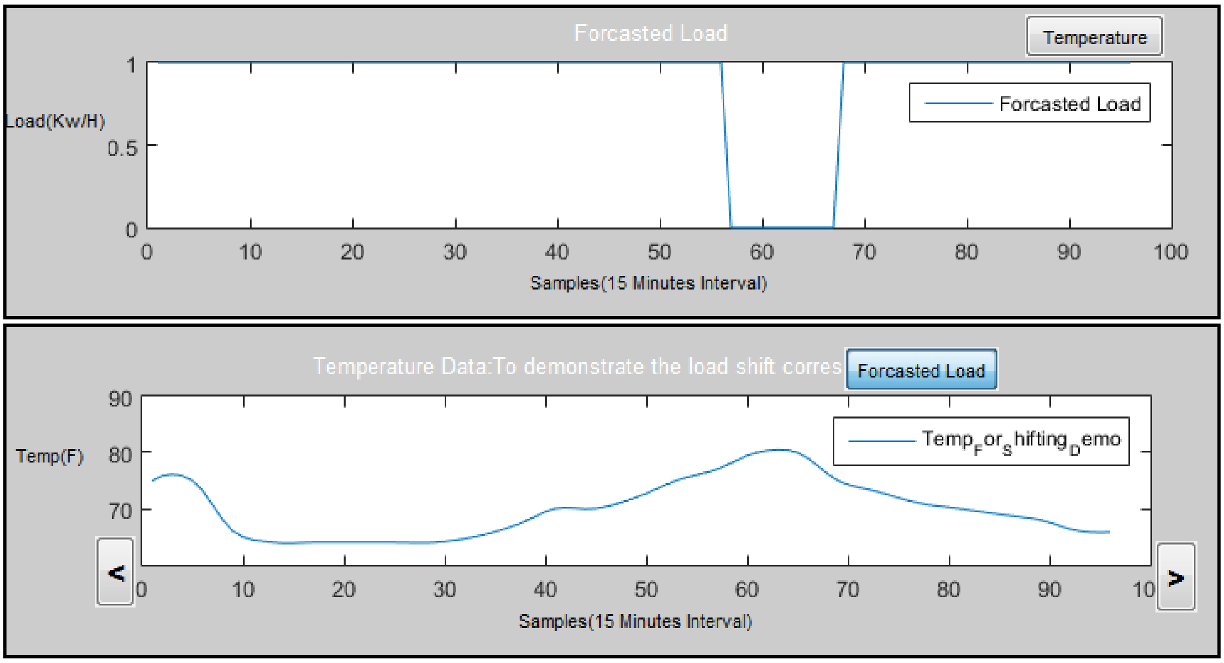


Figure 12. *Heater's switching pattern and temperature curve for the next 24 h*

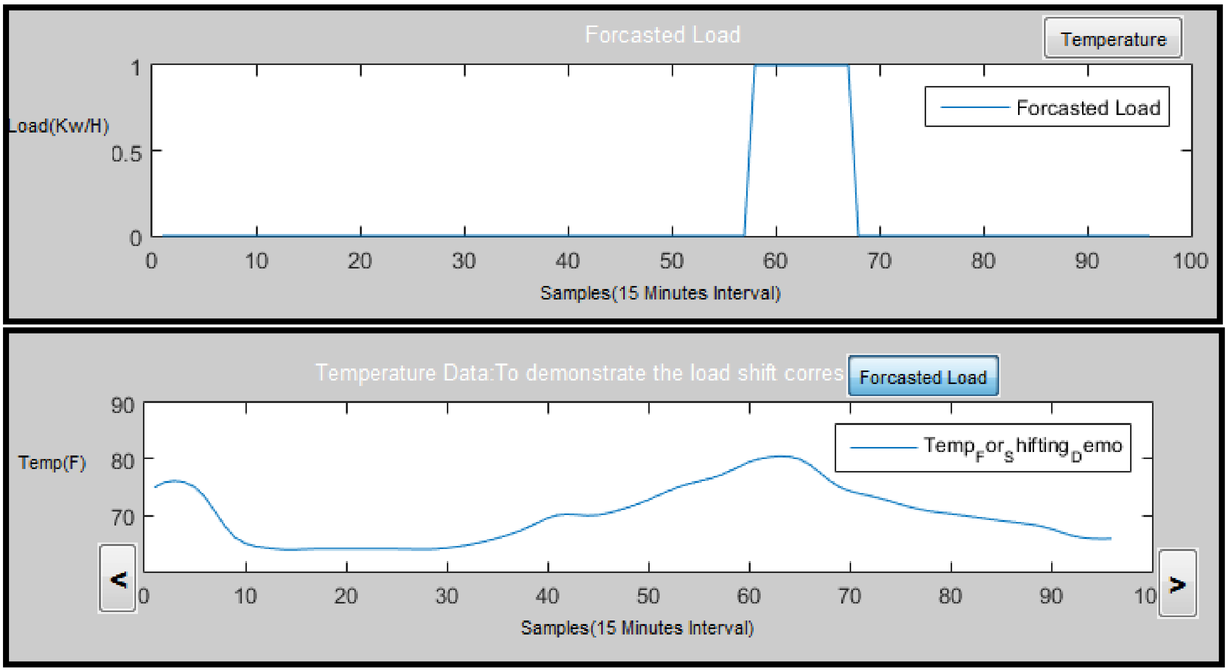


***Figure 13**. Cooler's switching pattern and temperature curve for the next 24 h*

3.8 Conclusion of Phase 1

In order to achieve the primary objective of this thesis, i.e. to formulate a smart scheduling algorithm for DEMS, the individual device switching patterns ω_{Load} were considered as a part of the phase 1 output rather than the aggregated power consumption $\omega_{Agg\_Power\_Consumption}$. In both the cases the prediction results were visualized and analyzed to identify a predictor model that predicts the future trend with relatively high prediction accuracy A. The predictor models have been trained with the historical data H to predict the future for finite-horizon x_{t+p}, where p is the horizon in hours, i.e. 24 h. The $H = [H_{Load}, H_{Price}, \upsilon_p \mid \upsilon_i]$ data is assumed to be stochastic. MAPE is a statistical tool that has been used to measure the error in prediction. i.e. $MAPE = OC - \omega$.

Case 1: Prediction of the aggregated power consumption and electricity price pattern:

The $H = [H_{Agg\_Power\_Consumption} + H_{Price} + \upsilon_p \mid \upsilon_i]$ data with an horizon of two years is applied as an input to the predictor models and the next 24 h power consumption and the electricity price predictions are obtained. The aggregated power consumption is stated as P^t_{Total}, i.e. $P^t_{Total} = P^t_{WD} + P^t_{Heater} + P^t_{Cooler} + \cdots + P^t_{TV}$. Each predictor model manages to predict the pattern for the next 24 h, i.e. $\omega_{Agg\_Power\_Consumption|ARIMA},\ldots,\omega_{Agg\_Power\_Consumption\ |NN}$, however, with different prediction accuracies A. The accuracy of the prediction algorithm A is evaluated w.r.t. the occurred data OC, i.e. the predicted 24 h power consumption and electricity price data ($\omega_{Agg\_\ Power\_Consumption}$ and ω_{Price} of Jan 02nd, 2017) for each predictor model is compared against the occurred data ($OC_{Agg\_Power\_Consumption}$ and OC_{Price} of Jan 02nd, 2017) to compute A. Accuracy is given by $A = (1\text{-}MAPE) \bullet 100$. The MAPE values for different predictor models are shown in *Table 2*. In case of the aggregated power consumption prediction, *Neural Network* predicts relatively better compared to the other models, i.e. $MAPE_{Load|NN}^{Working_{Day}} \leq MAPE_{Load|ANFIS}^{Working\_Day} \leq MAPE_{Load|ARIMA}^{Working\_Day} \leq MAPE_{Load|Lin\_Reg}^{Working\_Day} \leq MAPE_{Load|Non-Lin\_Reg}^{Working\_Day} \leq MAPE_{Load|State\_Space}^{Working\_Day}$. In case of the electricity price prediction, the *ANFIS* is predicting relatively better. $MAPE_{Price|ANFIS}^{Working\_Day} \leq MAPE_{Price|ARIMA}^{Working\_Day} \leq MAPE_{Price|Lin\_Reg}^{Working\_Day} \leq MAPE_{Price|Non-lin\_Reg}^{Working\_Day} \leq MAPE_{Price|NN}^{Working\_Day}$. Another notable fact is that the state space model in both cases, the electricity price and the aggregated power consumption prediction, is consuming a lot more time than acceptable: nearly in hours, in addition, the prediction accuracy is relatively bad, i.e. $MAPE_{Load|State\_Space}^{Working\_Day} \longrightarrow 25\%$. Therefore, the state space model is discarded for further activities and is not considered in this work.

Case 2: Prediction of the device switching pattern and electricity price pattern:

The $H = [H_{Load} + H_{Price} + \upsilon_p \mid \upsilon_i]$ data with an horizon of one week is applied as an input for each predictor model and the next 24 *h* switching and electricity price predictions are obtained. In this case, the term *load* refers to the switching pattern of a device. Each predictor model manages to predict the switching pattern for the next 24 *h,* i.e. $\omega_{Load|ARIMA},\ldots,\omega_{Load|NN}$, however, with different prediction accuracies *A*. The accuracy of the prediction algorithm *A* is evaluated w.r.t. the occurred switching pattern, i.e. the predicted 24 *h* switching pattern and electricity price data (ω_{Load} and ω_{Price} of April 29th, 2017) for each predictor model is compared against the occurred data (OC_{Load} and OC_{Price} of April 29th, 2017) to compute its prediction accuracy. In case of the device switching prediction, the prediction accuracy A_{Load} of each predictor model is better, i.e. *Figure 10* show that *A* for the considered date is approximately 100%, and MAPE is of the order 10^{-15}, which is negligible. However, on the other hand, the electricity price predictions had varied prediction accuracies *A*. The MAPE value of the electricity price prediction for different algorithms is shown in *Table 2*, i.e. irrespective of the device count (either for the aggregated power consumption or for the individual device) the same electricity price data is used. A good predictor model for device's switching pattern and electricity price predictions is decided by carrying out rigorous evaluation tests as stated in SECTION V.

4. Phase 2: Stochastic scheduling/Optimization

The output of phase 1 is the un-optimized device's switching pattern and electricity price data, i.e. ω_{Load} and ω_{Price}. In phase 2, an MDP problem is employed to achieve the desired output, ω^*_{Load}. The desired output is: optimized ON-OFF scheduling pattern for next 24 h, considering several aspects such as temperature (if device is temperature dependent), the seasonality SEN, the maximum allowed device waiting time W_D^{Max}, the maximum device interruptions possible in a day $INTR_D^{Max}$, total power requirement in a day $P_D^{Total\_Day}$, RTP ζ_t, the next 24 h CoEC $C_t^{Total\_Day}$ and the total device execution time left T_{Exe}. This section states the following

1. The theory of Markov chains and MDP.
2. The methods for obtaining the optimum policies with precise mathematical models.
3. The outline of the MDP implementation.
4. The simulations of phase 2 in detail and the considered transition probability P and reward function R.
5. Conclusion.

4.1 Introduction to Markov Decision theory

Markov decision theory has a wide range of applications, such as inventory control, computer science, resource allocation etc. A Markov chains is a stochastic process, in which the probabilistic occurrence of next state is assumed to be dependent on current state alone, instead of the past states and this is called *Markov process*. The two fundamentals of Markov processes are the states and state transitions. The system states are assumed to be a random variable indicating the property of the system at an instant and the state transition is the change in system state at a given instant. Consider a set of states $S=\{s_1,s_2,...,s_n\}$, at time t, the process starts from one of the states s_i and jumps to another s_j in the next time step $t+1$. This transition probability is captured by p_{ij} *or* $P(S)$. The Markov process is broadly classified into two categories, i.e. continuous-time Markov process and discrete-time Markov process. A case in which the states are random variables and time between state transitions is fixed, is called *discrete-time Markov process*. Similarly a case in which both states and time between state transitions are random variables is called *continuous-time Markov process* [12].

Discrete-time Markov process:

Any sequence of system states, in which the probabilistic occurrence of next state is assumed to be dependent on current state alone, instead of the past states and the time between state transitions is assumed to be fixed is called *Discrete-time Markov process*. This technique is considered a way to include stochasticity or randomness in decision making. Discrete-time Markov process is a stochastic process, which is discrete in both time and space.

$$P(S_{t+1} = l_{t+1} \mid S_t = l_t, S_{t-1} = l_{t-1},...,S_0 = l_0) = P(S_{t+1} = l_{t+1} \mid S_t = l_t), \quad t = 0,1, ... \qquad [38]$$

Where,

$S_t, t = \{0,1,2,...\}$: *Sequence of system states in a discrete time stochastic process with a discrete state space S.*

$l_0,.....l_{t+1}$: *All possible values.*

t : *Discrete time description.*

Continuous-time Markov process:

Continuous-time Markov process is seen to be a generic model, since time is allowed to be continuous $t \geq 0$. The continuous-time Markov process is simply a discrete-time Markov process, but in which the state transitions can happen at any time, i.e. the time between state transitions is assumed to be a random variable. Transition rate forms the central concept in continuous-time Markov process.

` $$P(S(t + \Delta t = l \mid S_t = l)) = 1 - v_l \Delta t + o(\Delta t), \; t \geq 0 \qquad [39]$$

Where,

$S_t, t \geq 0$: *Sequence of system states in a discrete time stochastic process with a discrete state space S.*

$o(\Delta t)$: *is negligibly small compared to Δt when $\Delta t \rightarrow 0$. Therefore $v_l \Delta t + o(\Delta t)$ is the probability that the process moves to another state in Δt time units.*

$l_0,...,l_{t+1}$: *All possible values.*

v_l : Time interval between the state transitions. *$l \in Z$ are bounded.*

Since the next state is a function of transition probability T. $v_1 T$ is the transition rates of continuous Markov process.

$$P(S(t + \Delta t = k \mid S_t = l) = v_1 T \Delta t + o(\Delta t), \quad k \neq l \qquad [40]$$

Markov decision process (MDP)

The theory of MDP is a theory of controlled Markov chains [24]. In Markovian process, it is assumed that the next state could be predicted by present state alone, i.e. the next state's prediction using present state is assumed to be same as prediction done using complete available historic data. The theory of MDP could be used to solve many real world problems, out of which DEMS is an example. In the real world scenarios or the environment, there could be several possible next steps. The MDP helps to choose the optimal action or the optimal control decision that can lead to an optimal step/state. MDP is employed when the state transition is partly controlled by the system user. MDP contains Markov chains in addition to the action space and reward functions. Action space captures possible next action and, the reward function will influence the decision by adding weighting factors to output an optimal policy. At any given instant of time the process is in state *S.* Each state in the process has several available actions *a.* If the action is chosen randomly the resulting next state is *S'* and this state *S'* has a corresponding reward function $R_a(S, S')$. Every state has a reward function corresponding to the action from which it resulted. Thus the next state *S'* is dependent on the present state *S* and chosen action *a*, an action which is allowed in state *S*. This state transition is denoted by $P_a(S, S')$. Similar to Markov process the MDP could be broadly categorized as discrete-time Markov decision process and continuous-time Markov decision process.

MDP contains

a) State space: Captures the possible states of the stochastic process. It is denoted by *'S'*. In MDP each state has state value called $V_i(s)$. The state value could be calculated by Bellman's equation, i.e. *Equation 46*. Accordingly from each state, there could be many possible actions which leads to a different state.

b) Action space: The allowed actions at each state are captured. It is denoted by *'A'*. Different actions lead to different states.

c) Transition probability: The transition between states could be captured by transition probabilities. It is denoted by *T(S,a,s')* or *P(S,a,s')*. Which represents the conditional probability of next state *S'* occurring, given the present state *S* and action *a*.

$$P(S_{t+1} = s' \mid S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \ldots, S_0 = s_0) = P(S_{t+1} = s' \mid S_t = s_t, A_t = a_t)$$

The above representation signifies the fact that the next state is dependent on the current state, i.e. the predicted next state having all the previous states and actions into consideration is no different from the next step predicted with the present state and corresponding action.

d) Reward function: These are the influencing factors to decide on an optimized action or step. It is denoted by *R(S,a,s')* or just *R(S)* or *R(s').* Each chosen action results in corresponding reward *R(S)*, which is assigned to the next state once the decision is executed.

e) Discount factor: The reward function might lose their reward value over a time, this is captured in discount factor. It is denoted by *ϒ*. It is the parameter that decides amount of priority assigned to immediate rewards compared to later rewards.

Discrete-time Markov decision model

The stochastic system $\{Z_t, t = 0, 1, 2...\}$ represent a discrete-time Markov decision model. The sequence of system states belongs to a discrete state space. The time *t* corresponds to a state transition interval or system review instances and are identical. At each state of the system, the allowed set of actions A_l, $l \in Z$ is assumed to be finite. For each executed action the resulting state obtains a corresponding reward *R(S)*. For the action executed, the state transition probability function is given by $p_{lk}(a)$. The state transition probability at any state considering all the correspondingly allowed actions should satisfy

$$\sum_{k \epsilon Z}^{Z} p_{lk} = 1, \; l \epsilon Z \quad [41]$$

Such a system is called *Discrete-time Markov decision model*. The main aim is to obtain the stationary policy π defining optimal actions at every state. This policy π can be found by employing techniques like value-iteration algorithms or policy-iteration algorithms. The resulting policy π^* has the *average reward optimal* for each policy π.

$$R(\pi^*) \geq R(\pi) \quad [42]$$

Where,

$$R(\pi) = \sum_{k \epsilon Z}^{Z} r_k(\pi) Q_k(\pi), \; k \epsilon Z \quad [43]$$

Q_k defines the steady state probabilities, i.e. a steady state distributions indicates that a limiting distribution is reached where at a state s_i, the transition probability to an accessible state s_j, at that specific instant t and in the future $t+x$ is same. With an assumption that for a chosen policy π a state s exists, which can be reached from any state, a normalized steady state probabilistic equation could be defined for any policy.

$$Q_k(\pi) = \sum_{k \epsilon Z}^{Z} p_{lk}(\pi) Q_k(\pi), \; k \epsilon Z \quad [44]$$

$$\sum_{k \epsilon Z}^{Z} Q_k(\pi) = 1 \quad [45]$$

Continuous-time Markov decision model

The stochastic system $\{Z_t, \; t \geq 0\}$ represents a continuous-time Markov decision model. Continuous-time Markov decision model is also referred as semi-Markov decision process. The time t corresponds to the system review instances and are not identical i.e. the states and the state transition interval are stochastic. At each state of the system, their exists, allowed set of actions A_l, $l \epsilon Z$ assumed to be finite. For each executed action the resulting state obtains a corresponding reward $R(S)$. The main aim is to obtain the stationary policy π defining optimal actions at every state. This policy π could be found by employing an algorithm such as value-iteration algorithm and policy-iteration algorithm. Continuous time value iteration algorithm is derived from discrete time value iteration algorithm by assuming identical average transition time τ'.

Methods for obtaining optimum MDP policies are:

1) Policy iteration algorithm:

In policy iteration the value of the policy is evaluated, i.e. the $V_\pi(s)$ is evaluated. For any given new policy the policy value is

$$\sum_{S'}^{n} T(s, \pi(s), s')\big(R(s, \pi(s), s') + \Upsilon V_\pi(S)\big) \quad [46]$$

Where,

$T(s, \pi(s), s') = p_{lk}(\pi')$: *Transition probability for a selected action.*

$\pi' \qquad An\ improved\ policy\ with\ R(\pi *) \geq R(\pi).$

1) An initial arbitrary policy is chosen.

2) For the chosen policy the policy value is determined by equation.

$$V_\pi\ (s) = \sum_{S'}^{n} T(s, \pi(s), s')\big(R(s, \pi(s), s') + \Upsilon V_\pi(S)\big) \qquad [47]$$

3) Find an action a at each state such that the value of the policy is maximized.

$$\max_{a\epsilon A}\{\sum_{S'}^{n} T(s, \pi(s), s')\big(R(s, \pi(s), s') + \Upsilon V_\pi(S)\big)\} \qquad [48]$$

If $V_{\pi'}(S) \neq V_\pi(S)$ then go to step 2 and repeat the computation with π' as π. This process is iterated until the policy converges to π^*. The $V_\pi(S)$ is the optimal policy if $V_{\pi'}(S) = V_\pi(S)$ and correspondingly the π' is termed as π^* and $R(\pi')$ is termed as $R(\pi^*)$.

2) Value iteration algorithm:

Instead of solving linear equations, this algorithm adopts dynamic programming to recursively evaluate the value $V_i(S)$, where i is iteration. The search is for the optimal policy π^*, which gives optimal actions at each state. The solution to find the optimal policy or action sequence is provided by Bellman equation stated in *Equation 49*. This equation is used recursively to arrive at the optimal value. $V(s')$ corresponds to the value of the current state and $V_{i+1}(s)$ is value of the next state, according to Bellman equation. This is calculated for all the possible actions at each state. As a result the optimum state value: $V_{i+1}(s)$, is opted.

$$V_{i+1}(s) \leftarrow \max_a \sum_{S'}^{n} T_a(s, s')\left(R_a(s, s') + \Upsilon V^i(s')\right) \qquad [49]$$

1) To start the algorithm, initialize $V_i(S) = 0$, where $i=0$.

2) Find a policy π which maximizes the R.H.S[7] of the *Equation 49* for all states S.

3) Compute the convergence or the required relative accuracy limits. For an optimal policy the average reward $\bar{R}(\pi) = V_i(S) - V_{i+1}(S)$ is assumed to be within the bounds.

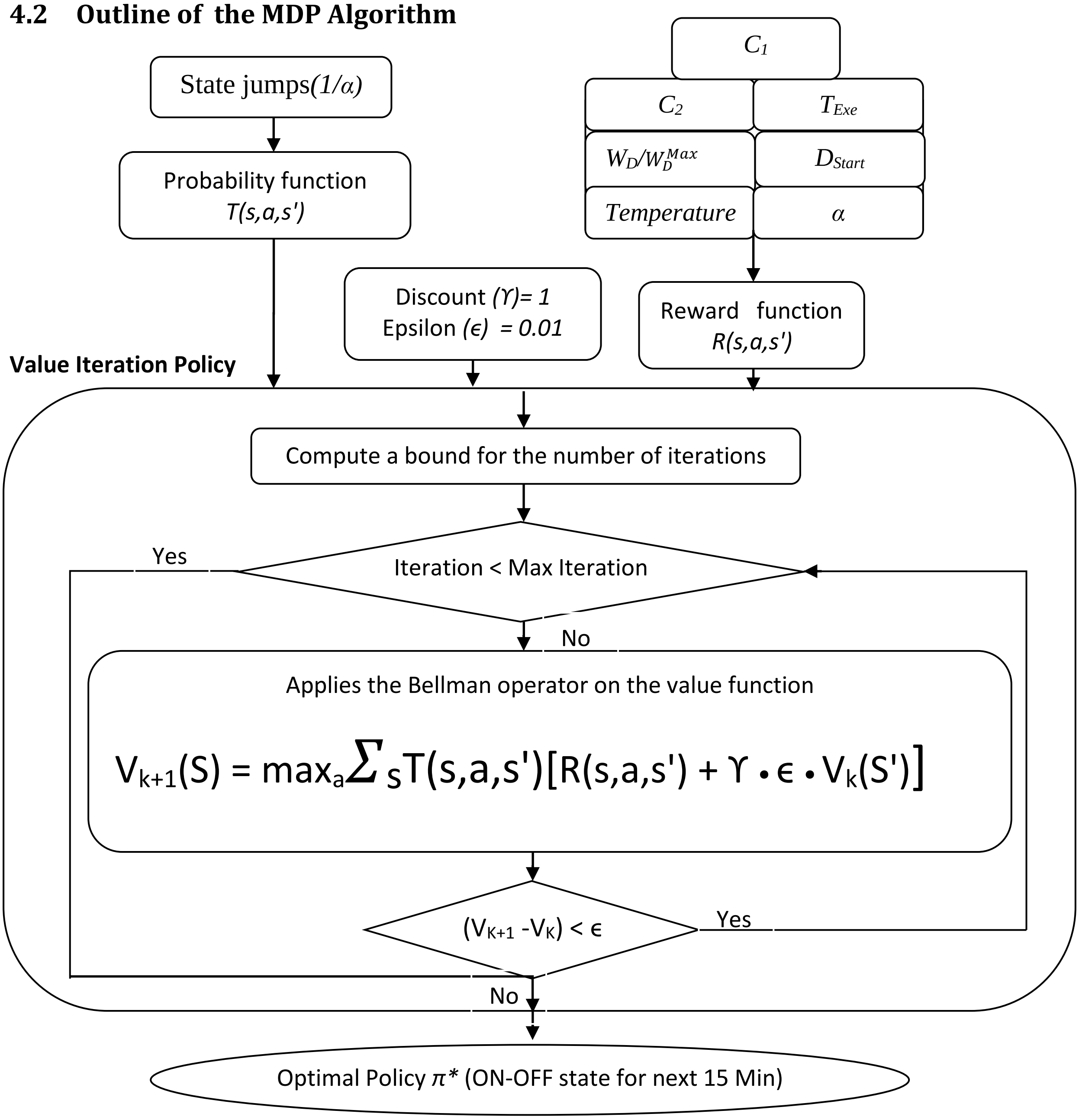
$$m_n = \min\{V_i(S) - V_{i+1}(S)\} \qquad [50]$$

$$M_n = \max\{V_i(S) - V_{i+1}(S)\} \qquad [51]$$

If $\epsilon m_n \geq M_n - m_n \geq 0$, where ϵ determines the required relative accuracy, stop the algorithm with policy π' when the difference approaches the maximum reward from the system. If not go to the step 2.

[7]R.H.S = Right hand side

4.2 Outline of the MDP Algorithm



Suggested approach:

In this thesis, a Value iteration algorithm is adopted. In case of a Water pump the next 15 minute's forecasted device switching ω_{Load} and in case of heater or cooler the 15 minute's determined device switching and 15 minute's forecasted electricity price ω_{Price} (irrespective of device type) is applied as input to MDP. MDP outputs or determines, the optimal switching state of the device for next 15 Minutes π^*. The MDP is reformulated and value iteration policy is run for every sampling instant (15 Minute interval). This process is repeated for all 96 samples to obtain ω^*_{Load} = $[\pi^*_1, \pi^*_2, \pi^*_3,...,\pi^*_{96}]$. The optimization is carried out based on 6 primitive parameters and an additional optional parameter, i.e. temperature (If temperature dependent device: for e.g.: Cooler/Heater). These parameters along with transition probability *T(s,a,s')* is applied as input to the value iteration policy to obtain ω^*_{Load}, i.e. the optimum device switching ω^*_{Load} is obtained by recursively running the Bellman equation. The reward components are

a) Price for next 1 hour *(C_1)*
b) Electricity price ω^{t}_{Price} at a specific instant of time *t* *(C_2)*
c) Total device execution time left *(T_{Exe})*
d) Present waited time *(W_D/W_D^{Max})*
e) Alpha: state jumps *(α)*
f) Checks, if device needs to be turned ON that day *(D_{Start})*

Inputs Applied:

States (S): The cost of energy consumption is dynamically divided equally into 5 states.

Step1:

Obtain the cost of energy consumption for the next 24 *h*: 96 samples-having 15 minutes interval.

$$C_t^{Total\_Day} = \text{Real Time Price} \bullet \text{Load} = \zeta_t \bullet \omega_{Load}$$

$$C_t^{Total\_Day} = \{set\ of\ CoEC\ of\ next\ 24\ h\} = \omega_{Price} \quad [52]$$

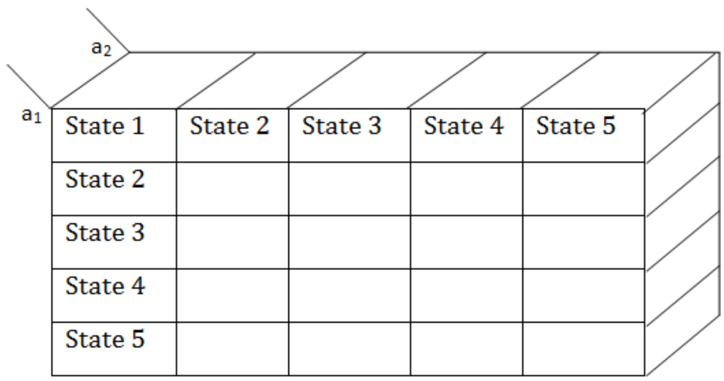
Step2:

The maximum in $C_t^{Total\_Day}$ is divided by 5 to derive the step size. This step size defines the building block of the *T(s,a,s')*, i.e. the stochasticity of the random variable $C_t^{Total\_Day}$ is recorded within a state probability transition function. State space captures every day's price dynamics effectively, since *Max* *(* $C_t^{Total\_Day}$ *)* varies every day with ω_{Price}.

$$Step_{size} = Max\ (\ C_t^{Total\_Day})\ /\ 5 \quad [53]$$

Transition/Probability function (T(s,a,s')): The states space is dynamically formed for each day by dividing the $C_t^{Total\_Day}$ range into 5 equal steps. The difference between the successive states is equal to the *$Step_{size}$*. The transition probability is a *p* x *p* x *q* matrix, where *p* is the number of states and *q* is the number of possible actions *a*.

$$T(s,a,s') = \frac{Number\ of\ transisions\ from\ state\ S\ to\ S'}{Total\ number\ of\ transistions\ from\ the\ state\ S} \quad [54]$$



Typical example: In this case, the transition probability is a 5 x 5 x 2 matrix, i.e. 5 states [S1, S2, S3, S4, S5] and 2 actions [ON, OFF].

OFF

```
val(:,:,1) =

     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
```

ON

```
val(:,:,2) =

    0.9737         0         0         0    0.0263
    0.2000    0.6000    0.2000         0         0
         0    0.1429    0.7143    0.1429         0
         0         0    0.1250    0.7500    0.1250
         0    0.0270         0    0.0270    0.9459
```

Reward function (R(s,a,s')):

Reward function helps to choose an optimal action. Reward function depends on the current state S and the actions A_l, $l \in Z$, chosen at that sate. MDP provides an optimal switching schedule for the next 15 Minutes, based on 7 parameters or 7 tuples that constitute the reward function, as proposed in *Equation 59*.

1. Reward component 1 (α)- Alpha or State Jump factor [6]

By this reward component a state transition or jump from a higher CoEC state s_i to a lower CoEC state s_j is assigned a better reward than a jump from lower to higher, where $i > j$. Where state s_i is the initial state and s_j is the next state. The index value ranges from 1 to 5, as the length of state space is 5. As a consequence, an action which causes the state transition from higher to lower is preferred and chosen.

$$\alpha = \frac{Index\ of\ S_i}{Index\ of\ S_j}.$$

2. Reward component 2 (C_1)- Cost for next 1 hour

This reward component observers the next 1hour electricity price pattern and help to decide an optimal action. When this optimal actions is taken it leads to reduced $C_t^{Total\_Day}$. The standard deviation $H_{std(X)}$ is computed for the predicted price, ω_{Price}. If any sample in next 1 hour's predicted electricity price $\omega_{\mathrm{Price}}^{1\,\mathrm{h}} = [\omega_{\mathrm{Price}}^{t=0}, \omega_{\mathrm{Price}}^{t+15}, \omega_{\mathrm{Price}}^{t+30}, \omega_{\mathrm{Price}}^{t+45}]$ is higher than the $H_{std(X)}$, this reward component affects the reward function such that the device chooses to remain turned OFF for the next one hour, and chooses to remain ON when all the samples are below $H_{std(X)}$.

$$H_{std(X)} = \mu + std(X) \quad [55]$$

$$C_1 = \Sigma_4 \{True([\omega_{Price}^{t=0}, \omega_{Price}^{t+15}, \omega_{Price}^{t+30}, \omega_{Price}^{t+45}] < H_{std(X)}\} \quad [56]$$

Where,

μ: Mean of the Forecasted price data: set of 96 samples, each sample at15 min interval.

$std(X)$: Standard Deviation of the Forecasted price data: set of 96 samples, each sample at 15 min interval.

3. Reward component 3 (C_2) - Electricity price ω^t_{Price} at a specific instant of time t

This reward component observers the electricity price at that specific instant of time t and help to decide an optimal action. When this optimal actions is taken it leads to reduced $C_t^{Total\_Day}$. The predicted electricity price at successive time instant t, ω^t_{Price} is normalized as per *Equation 57*. Lower the ω^t_{Price} better the reward, as a consequence, the device chooses to remain turned ON. On the other hand the device chooses to remain turned OFF as the ω^t_{Price} approaches the day's high RTP or 1 (since the electricity price is normalized).

$$C_2 = \frac{\omega^t_{Price} - P_{Min}}{P_{Max} - P_{Min}} \quad [57]$$

Where,

ω^t_{Price} : *Electricity price at the current sampling time.*

P_{Min} : *Minimum electricity price of the day: Min{Set of prices: 96 samples} = min{ω_{Price} }*

P_{Max} : *Maximum electricity price of the day: Max{Set of prices: 96 samples} = max{ω_{Price} }*

4. Reward component 4 (T_{Exe}) - Device execution time left

This reward component assigns lower reward during the devices initial execution time, thereby less curtailment on its operation. On the other hand, this reward component forces the device to remain turned ON, when the device approaches its required execution time, thereby allowing the device to execute till the completion of specific task.

T_{Exe} = Required execution time - Elapsed execution time.

5. Reward component 5 (W_D/W_D^{Max}) - Present waited time [6]

A reward component for waiting time of the device. The reward increases as the device's present waiting time approaches the device's maximum waiting time. The reward function is the ratio of the W_D/W_D^{Max}, where W_D^{Max} is the maximum device waiting time and W_D is the device present waited time. The purpose of this reward component is to prioritize the device to turn ON when the device is delayed more.

6. Reward component 6 (D_{Start})- Checks, if device needs to be turned ON that day.

   This reward component scales the reward function with '*1*' if at all the device is required to be scheduled/ turned ON that particular day or else scales with *'0'*. As a consequence the scenario in which the devices being turned OFF at weekends are covered.

7. Reward component 7 (*Temperature*) - Temperature (If temperature dependent device).

   Depending upon the chosen device, the temperature factor is employed to scale the reward function. In case of a washing machine, the temperature factor is not used. In case of a heater, the operation is inversely proportional to temperature, i.e. the reward component scales the reward function with higher value to turn the device ON as the temperature reduces and forces the device to turn OFF when temperature crosses 25°C. In case of cooler it is directly proportional to temperature. The scaling factor is normalized to '1' as shown in the *Equation 58*.

$$Temperature = \frac{\omega^{t}_{Temperature} - Temperature_{Min}}{Temperature_{Max} - Temperature_{Min}} \quad [58]$$

Where,

$\omega^{t}_{Temperature}$: *Temperature at the current sampling time.*

$Temperature_{Min}$: *Minimum of entire day's temperature*

Min{Set of temperature: 96 samples} = min$\{\omega_{Temperature}\}$

$Temperature_{Max}$: *Maximum of entire day's temperature*

Max{Set of temperature: 96 samples} = max$\{\omega_{Temperature}\}$

Reward function (R(s,a,s')):

$$\left(R(s,a,s')\right) = \alpha \bullet C_1 \bullet C_2 \bullet T_{Exe} \bullet \frac{W_D}{W_D^{Max}} \bullet D_{Start} \bullet Temperature \quad [59]$$

Where,

α : *State jump factor.*
C_1 : *Cost for next 1 hour.*
C_2 : *How Low is the present cost (This very Moment).*
T_{Exe} : *Device execution time left.*
W_D/W_D^{Max} : *Present waited time.*
D_{Start} : *Checks, if device needs to be turned ON that day.*
Temperature : *Temperature (If Temp Dependent Device).*

4.3 Conclusion of Phase 2

The required device scheduling is achieved by considering entire next day's peak electricity price, temperature and the device specific parameters. A simulator is designed for the ease of analysis and performance evaluation. *Figure 14* shows the simulator, where the selections of training dates, locations, device types and several other features are provided e.g.: performance evaluation, display of predictor input and output plots etc.

***Figure 14**. 'Smart device Scheduler'- The Graphical User Interface (GUI)*

Optimization of the Water pump switching pattern:

Figure 15 shows a snapshot of the input and output plots of the proposed MDP. All data shown are normalized to 1. The upper axes contain the un-optimized ON-OFF patterns (Forecaster output) of a Water pump $\omega_{Load}|_{Norm}$, normalized electricity price of city Anchorage $\omega_{Price}|_{Norm}$, the daily mean value of normalized electricity price $\mu(\omega_{Price}|_{Norm})$ and the standard deviation for normalized electricity price $H_{std(X)}$ where X= $(\omega_{Price}|_{Norm})$. 2nd axes shows the optimized switching pattern for next 24 h, ω^*_{Load}, i.e. the device switching pattern if followed results in a lower CoEC, $C_t^{Total\_Day}$. In the second plot, the black markers in the graph indicate that, the device remains turned OFF when the prices overshoot above the standard deviation (as shown in the first plot).

Optimization of the heater switching pattern:

Figure 16 shows a snapshot of the input and output plots of MDP. All data shown are normalized to 1. Upper axes shows the optimized ON-OFF patterns for next 24 h, if followed results in a lower CoEC, $C_t^{Total\_Day}$. The 2nd axes contains the un-optimized ON-OFF patterns of a heater, normalized electricity price of city Anchorage $\omega_{Price}|_{Norm}$, mean value of normalized electricity price $\mu(\omega_{Price}|_{Norm})$ and standard deviation of normalized electricity price $H_{std(X)}$ where X= $(\omega_{Price}|_{Norm})$. In the first plot, the black markers in the graph indicate that, the device remains turned OFF when the prices overshoot above the standard deviation and/or when an ambient temperature is higher than 25° C too (as shown in the second and third plot).

Optimization of the cooler switching pattern:

Figure 17 shows a snapshot of the input and output plots of MDP. All data shown are normalized to 1. Upper axes shows the optimized ON-OFF patterns for next 24 h, that which if followed result in a lower CoEC $C_t^{Total\_Day}$. The 2nd axes contains the un-optimized ON-OFF patterns of cooler, normalized electricity price of city Anchorage $\omega_{Price}|_{Norm}$, mean value of normalized electricity price $\mu(\omega_{Price}|_{Norm})$, standard deviation of normalized electricity price $H_{std(X)}$ where X= $(\omega_{Price}|_{Norm})$. In the first plot, the black markers in the graph indicate that, the device remains turned OFF when the prices overshoot above the standard deviation and/or when an ambient temperature is lower than 25° C too (as shown in the second and third plot). The scheduled output in axes 1 is shifted slightly towards left than the forecasted switching because it's capable of tracking high ambient temperatures and accordingly schedule the device[8].

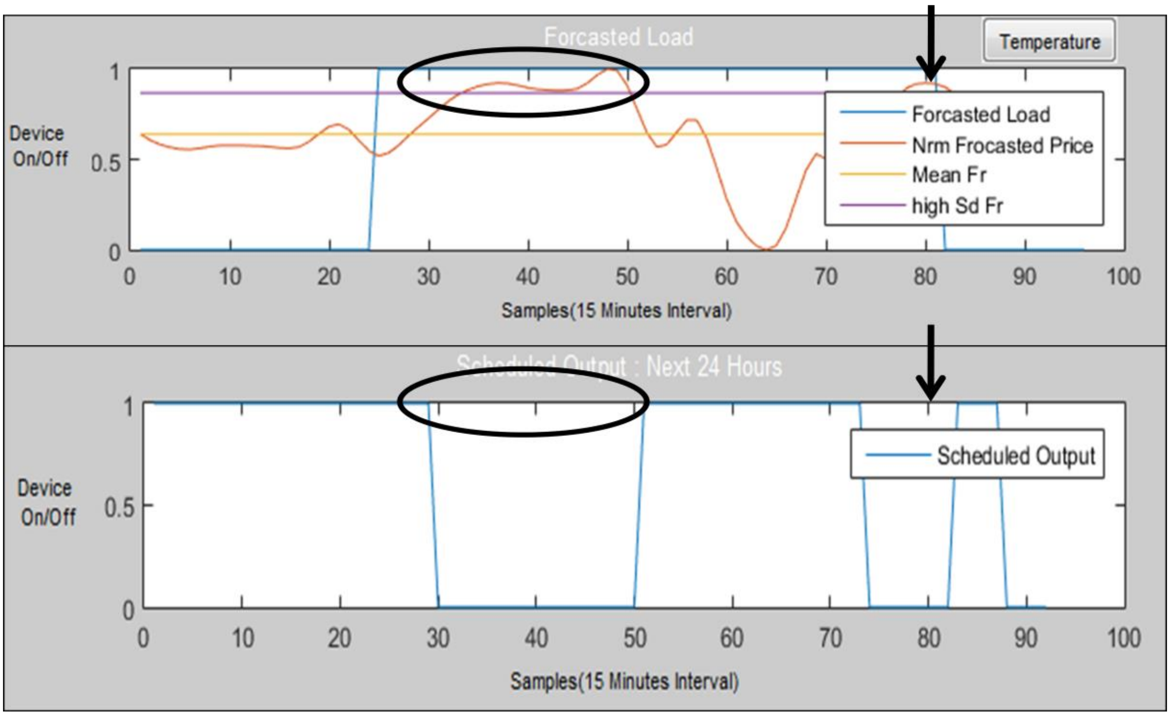


***Figure 15**. Optimized switching for Water pump: Forecast Date (April 29th of 2015)*

[8] The axes 1 and 3 should be observed keenly. Irrespective of the cooler's/heater's switching schedule, the MDP takes a valid decision and reschedules by tracking the high values of temperature in a day.

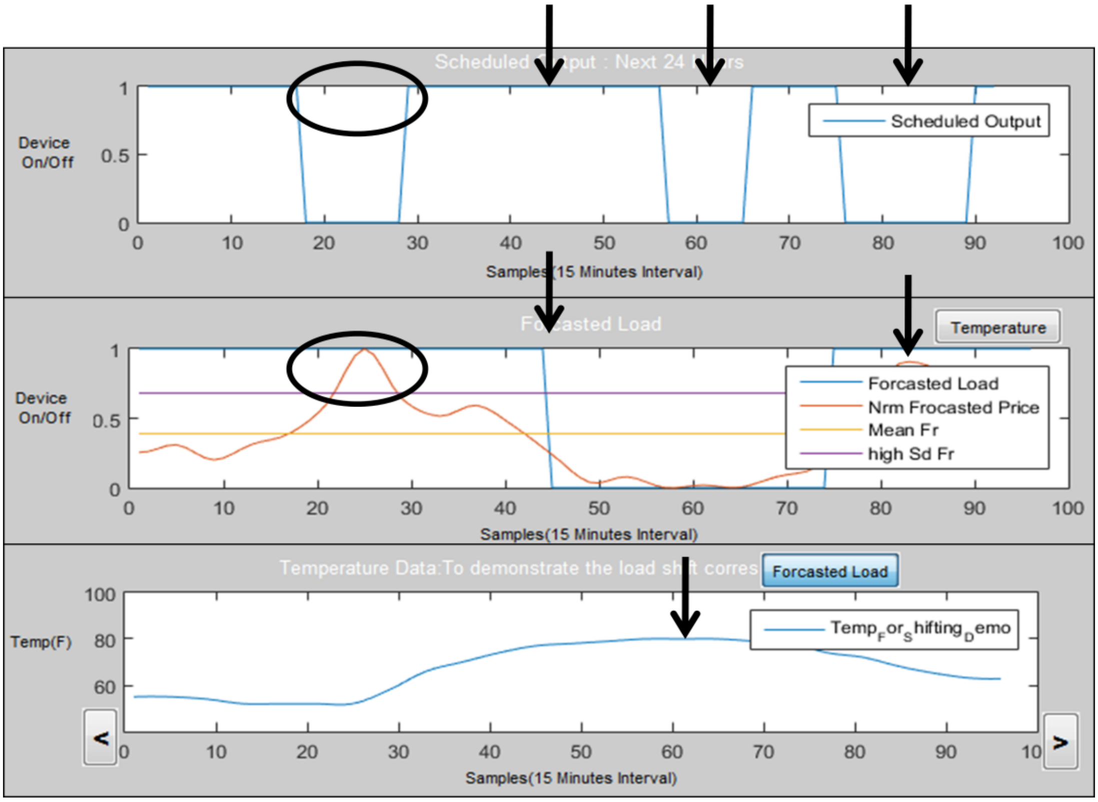


Figure 16. *Optimized switching for Heater: Forecast Date (April 29th of 2015)*

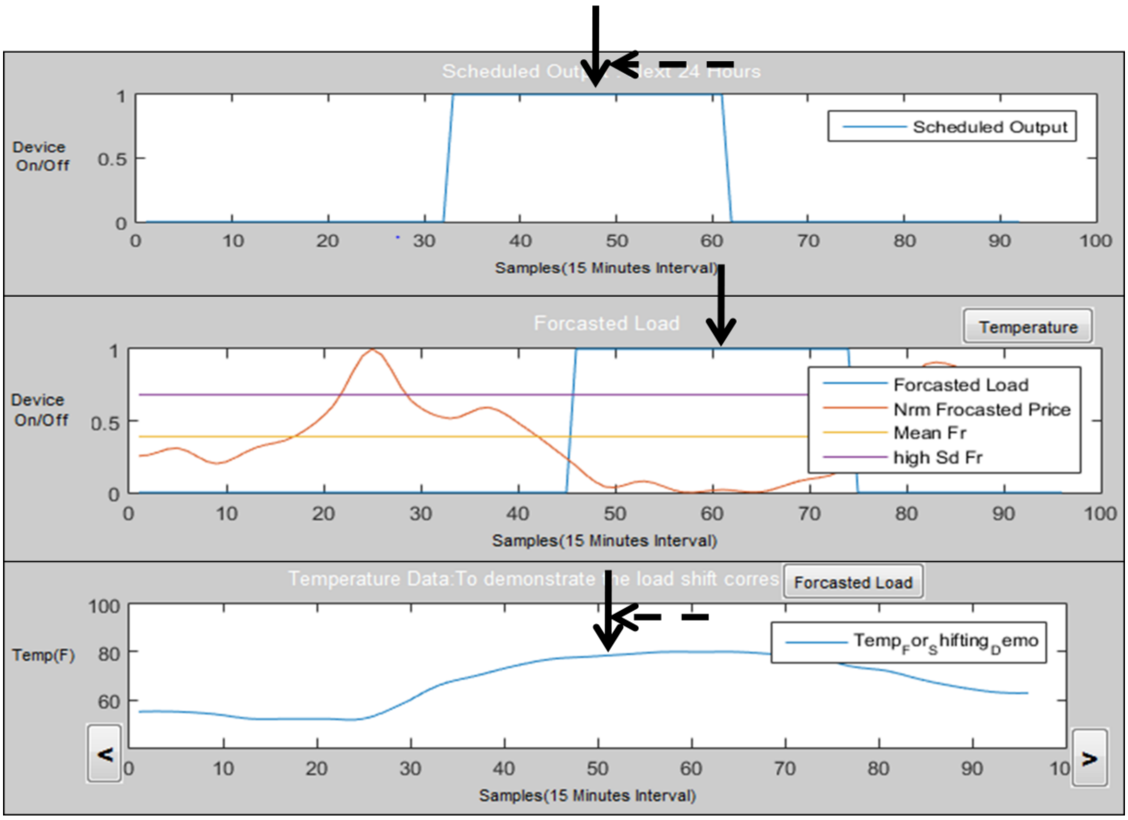


Figure 17. *Optimized switching for Cooler: Forecast Date (April 29th of 2015)*

5. Performance Evaluation

To show the good performance of the proposed algorithm, it is essential to evaluate the quality of algorithm through a solid performance evaluation. A simulator is designed, that helps to select different predictor algorithms, different training dataset (various start-end dates and respective forecasting date), different cities (warm city, colder city, moderate city) and different device type (Water pump, Heater and Cooler). Along with these selection facilities, the simulator also provides several other features that help to carry out effective performance evaluation. The performance evaluation investigates the following aspects.

1. The prediction accuracy A of each predictor model.
2. The best-fit predictor model for the electricity price prediction.
3. The best-fit predictor model for the device switching prediction.
4. The optimization of device switching schedule using MDP, i.e. the load shift achieved by considering peak price in a day and many other aspects, as compared to a schedule without MDP.

5.1 Prediction Accuracy

The prediction accuracy is evaluated by verifying the next 24 h forecasted data ω with the next 24 h occurred data OC. The training period considered is one week for each predictor model, irrespective of the chosen city and device. The forecasted data for an entire year, i.e. for 365 days $\omega^{D1}, \omega^{D2}, \omega^{D3}, \dots, \omega^{D365}$, is obtained by sliding training window/period (one week) H=[H_{Load}^{D-1},...,H_{Load}^{D-7},H_{Price}^{D-1},...,H_{Price}^{D-7},υ_p^{D-1},...,υ_p^{D-7}| $\upsilon_i^{D-1}, \dots, \upsilon_i^{D-1}$] over a year and it is compared with the occurred data of the year 2015, $OC^{D1}, OC^{D2}, \dots, OC^{D365}$, where D refers to a specific forecast date $1 \le D \le 365$.

The prediction accuracy is evaluated for various combinations of test conditions such as.

- Different City (Anchorage, Little Rock and Palmadel).
- Price or device switching pattern prediction.
- Different time within a year.
- Different predictor models.

Time considered: The year 2015 (Each day of the year 2015)

For example: The prediction accuracy of the predictor model *ARIMA,* is captured for an entire year, for the chosen city *Palmadel* and for the chosen device *Water pump*.

Device : Water pump

Table 3. *Prediction accuracy evaluation in case of Water pump*

Water pump	ANKORAGE		LITTLE_ROCK		PALMADEL	
	PRICE	LOAD	PRICE	LOAD	PRICE	LOAD
ARIMA	Mean:42.48 SD :72.72	Mean:97.94 SD : 115.76	Mean:42.84 SD :72.90	Mean:.97.94 SD : 115.76	Mean:42.73 SD :72.13	Mean:.97.94 SD : 115.76
Non Linear Regression	Mean:78.91 SD :109.79	Mean:97.92 SD : 115.83	Mean:78.91 SD :109.79	Mean:97.92 SD : 115.83	Mean:78.91 SD :109.79	Mean:97.94 SD : 115.86
Linear Regression	Mean:45.05 SD :75.07	Mean:97.80 SD : 115.90	Mean:44.81 SD :74.85	Mean:97.82 SD : 115.92	Mean:45.27 SD :75.27	Mean:97.80 SD : 115.90
ANFIS	Mean:42.41 SD :70.51	Mean:106.70 SD : 143.63	Mean:42.41 SD :70.51	Mean:106.70 SD : 143.62	Mean:42.41 SD :70.51	Mean:106.40 SD : 131.70
Neural Networking	Mean:54.30 SD :84.95	Mean:97.36 SD : 116.31	Mean:52.61 SD :83.39	Mean:97.28 SD : 115.63	Mean:52.36 SD :82.86	Mean:98.51 SD : 116.65

Device: Heater

Note: In case of Heater as the chosen device, only the electricity price is predicted.

Table 4. *Prediction accuracy evaluation in case of Heater*

Heater	ANKORAGE		LITTLE_ROCK		PALMADEL	
	PRICE	LOAD	PRICE	LOAD	PRICE	LOAD
ARIMA	Mean:42.48 SD :72.72	-	Mean:42.84 SD :72.90	-	Mean:42.73 SD :72.13	-
Non Linear Regression	Mean:78.91 SD :109.79	-	Mean:78.91 SD :109.79	-	Mean:78.91 SD :109.79	-
Linear Regression	Mean:45.05 SD :75.07	-	Mean:44.81 SD :74.85	-	Mean:45.27 SD :75.27	-
ANFIS	Mean:42.41 SD :70.51	-	Mean:42.41 SD :70.51	-	Mean:42.41 SD :70.51	-
Neural Networking	Mean:54.30 SD :84.95	-	Mean:52.61 SD :83.39	-	Mean:52.36 SD :82.86	-

Device: Cooler

Note: In case of Cooler as the chosen device, only the electricity price is predicted.

***Table 5.** Prediction accuracy evaluation in case of Cooler*

Cooler	ANKORAGE		LITTLE_ROCK		PALMADEL	
	PRICE	LOAD	PRICE	LOAD	PRICE	LOAD
ARIMA	Mean:42.48 SD :72.72	-	Mean:42.84 SD :72.90	-	Mean:42.73 SD :72.13	-
Non Linear Regression	Mean:78.91 SD :109.79	-	Mean:78.91 SD :109.79	-	Mean:78.91 SD :109.79	-
Linear Regression	Mean:45.05 SD :75.07	-	Mean:44.81 SD :74.85	-	Mean:45.27 SD :75.27	-
ANFIS	Mean:42.41 SD :70.51	-	Mean:42.41 SD :70.51	-	Mean:42.41 SD :70.51	-
Neural Networking	Mean:54.30 SD :84.95	-	Mean:52.61 SD :83.39	-	Mean:52.36 SD :82.86	-

Table 3 shows the device switching prediction accuracy evaluation results of Water pump, for each city and for each predictor model under consideration. The predictors trains with one week of *H,* w.r.t specific forecast date. For obtaining the performance evaluation of each day of an year, this training period is slid over a year, so as to predict all 365 days of year $H = [H_{Load}^{D-1},...,H_{Load}^{D-7}, H_{Price}^{D-1},...,H_{Price}^{D-7}, \upsilon_p^{D-1},...,\upsilon_p^{D-7} \mid \upsilon_i^{D-1}, \dots, \upsilon_i^{D-7}]$, where *D* refers to specific forecast date $1 \leq D \leq 365$. Each day ω_{Load} and ω_{Price} results are compared with the occurred data OC_{Load} and OC_{Price}. As per *Equation 37,* the MAPE value is calculated for each day. Accuracy in percentage is calculated by $A = (1\text{-}MAPE) \bullet 100$. Accordingly, 365 accuracy samples are computed, each corresponding to each day of the year *D,* where $1 \leq D \leq 365$. A Mean value in the table represents the mean of accuracies over an entire year in percentage $\mu_{accuracy}^{year}$. A standard deviation value $H_{std(accuracy)}^{year} = \mu_{accuracy}^{year} + std(A^{D1}, A^{D2}, A^{D3}, \dots, A^{D365})$ in the table represents the deviation (over a year) from the mean accuracy value and lower standard deviation value infers a better prediction. For ω_{Load} the mean of A over an year , $\mu_{accuracy}^{year}$ value is greater than 97% and $H_{std(accuracy)}^{year}$ is of range 15%-20%, irrespective of chosen city and chosen predictor model, except ANFIS, i.e. except ANFIS all other predictor models are competitively predicting well, i.e. e.g.: $A_{Load\,|ARIMA} \approx \dots \approx A_{Load\,|Non\_Lin\_Reg}$. For ω_{Price}, due to its stochastic behavior, it is difficult for any chosen predictor model to predict with 100% accuracy. Correspondingly, the $\mu_{accuracy}^{year}$ and $H_{std(accuracy)}^{year}$ varies drastically. *Table 4* and *Table 5* shows the prediction accuracy evaluation results for Heater and Cooler respectively. For the device type Heater or Cooler, the device switching patterns are not predicted but are determined from the daily temperature

curves $\omega_{Temperature}$. Only the ω_{Price} evaluation is tabulated in *Table 4* and *Table 5*. The ambient temperature to be maintained by heater or cooler is at 25° C as briefed in SECTION III.

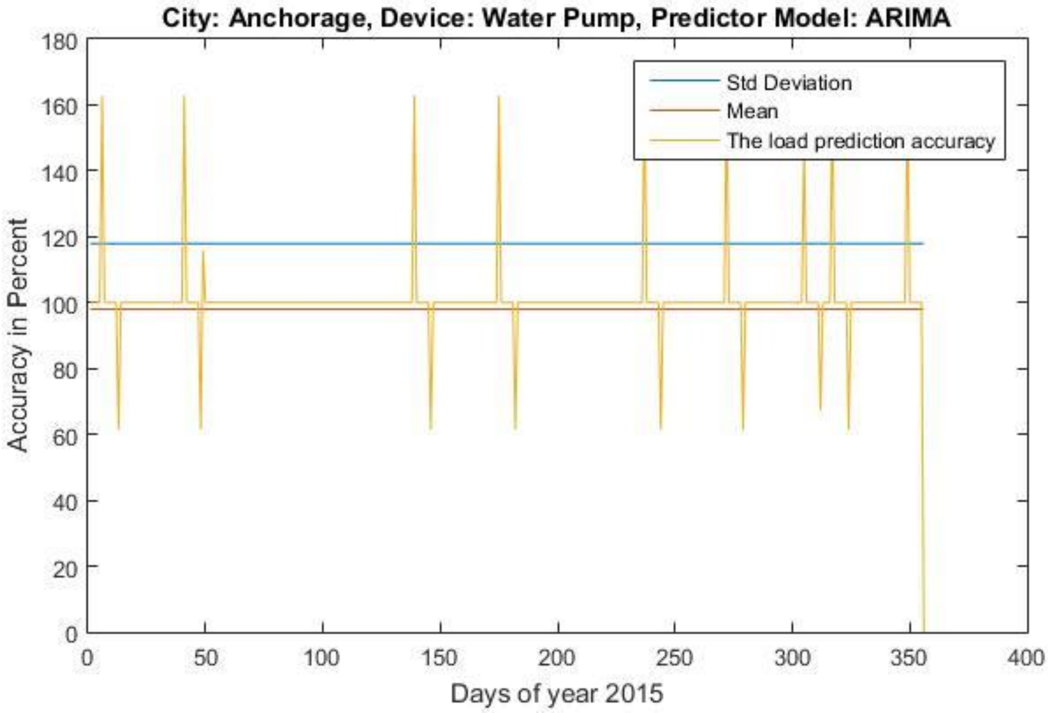


***Figure 18**. Device switching prediction accuracy for the considered case (Device: Water pump, City: Anchorage, Predictor model: ARIMA)*

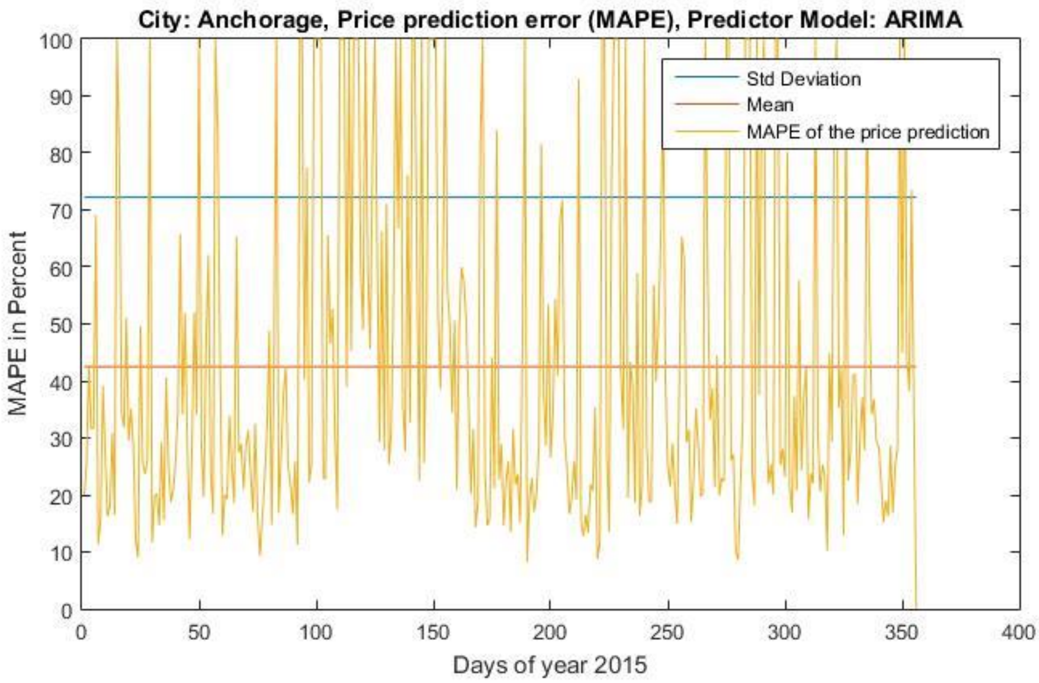


***Figure 19**. Error in Electricity price prediction for the considered case (Device: Water pump, City: Anchorage, Predictor model: ARIMA)*

Figure 18 and *Figure 19* shows the plot of prediction accuracies $A_{Load|ARIMA}$ and $A_{Price|ARIMA}$ respectively. The considered test case is for the device type Water pump, the chosen predictor model is ARIMA and the chosen city is Anchorage. The plots of prediction accuracies, corresponding to various test cases is provided in Appendix I. Since the year 2015 alone is considered, there exist no sufficient training data to predict each day of the 1st week, hence in all these plots, we observe the highest sampling index in X-axis as 357 instead of 365. The 1st week of the year serves as an initial training period for predicting device switching and electricity price on Jan 8th of 2015. The plots contain accuracies in percentage i.e. the plot contains A_{Load}, A_{Price}, the mean accuracy over the year $\mu_{accuracy}^{year}$ and the standard deviation over the year $H_{std(accuracy)}^{year}$.

Table 3, Table 4 and *Table 5* shows that, it is difficult to select the best predictor model that fairs best in all test cases, as certain predictor model predict better than its peer in few test conditions but do not performs well in the rest of test conditions. These selections are based on the $\mu_{accuracy}^{year}$, MAPE, and respective $H_{std(accuracy)}^{year}$ values, i.e. based on these values the predictor models are assigned with various priorities. The closer the yearly mean of device switching prediction accuracy to 100%, $\mu_{accuracy}^{year} \longrightarrow$ 100% and lesser the standard deviation value $H_{std(accuracy)}^{year} \longrightarrow \mu_{accuracy}^{year}$, higher the priority assigned to that predictor model for the corresponding test case. An example is shown in *Table 7* for the load/ device's switching prediction evaluation of each predictor model for the device type Water pump and the chosen city, Anchorage. In case of price prediction, the prediction error is considered as an evaluation parameter, i.e. the lower the Mean of MAPEs $\mu_{MAPE}^{year} \longrightarrow 0\%$ and the lower the standard deviation value $H_{std(MAPE)}^{year} \longrightarrow \mu_{MAPE}^{year}$ the higher the priority. An example is shown in *Table 6* for price prediction evaluation of each predictor model for the device type Water pump and the chosen city, Anchorage. Similarly, the task of priority assignment is carried out for all other test cases and *Table 8* shows the summary of evaluation, considering different cities and different device types. From *Figure 20*, it is evident that ARIMA is a clear winner of the prediction accuracy evaluation.

Table 6.*Electricity Price prediction evaluation*

Washing Machine	ANKORAGE	
	PRICE	Priority Assignment
ARIMA	Mean:42.48 SD :72.72	4
Non Linear Regression	Mean:78.91 SD :109.79	1
Linear Regression	Mean:45.05 SD :75.07	3
ANFIS	Mean:42.41 SD :70.51	5
Neural Networking	Mean:54.30 SD :84.95	2

Table 7. *Load prediction evaluation*

Washing Machine	ANKORAGE	
	Load	Priority Assignment
ARIMA	Mean:97.94 SD : 115.76	5
Non Linear Regression	Mean:97.92 SD : 115.83	4
Linear Regression	Mean:97.80 SD : 115.90	3
ANFIS	Mean:106.70 SD : 143.63	1
Neural Networking	Mean:97.36 SD : 116.31	2

Table 8. *Overall Best fit evaluation*

Predictor	Priority Assignment value												SUM
DEVICE	Water pump						Heater			Cooler			
CITY	Anch		Ltl Rok		Pal		Anch	Ltl Rok	Pal	Anch	Ltl Rok	Pal	
Price/Load	Price	Load	Price	Load	Price	Load	Price	Price	Price	Price	Price	Price	
ARIMA	4	5	4	5	4	5	3	4	4	3	4	4	49
Non Linear Regression	1	4	1	4	1	4	1	1	1	1	1	1	21
Linear Regression	3	3	3	3	3	3	5	3	3	5	3	3	40
ANFIS	5	1	5	1	5	1	4	5	5	4	5	5	46
Neural Networking	2	2	2	2	2	2	2	2	2	2	2	2	24

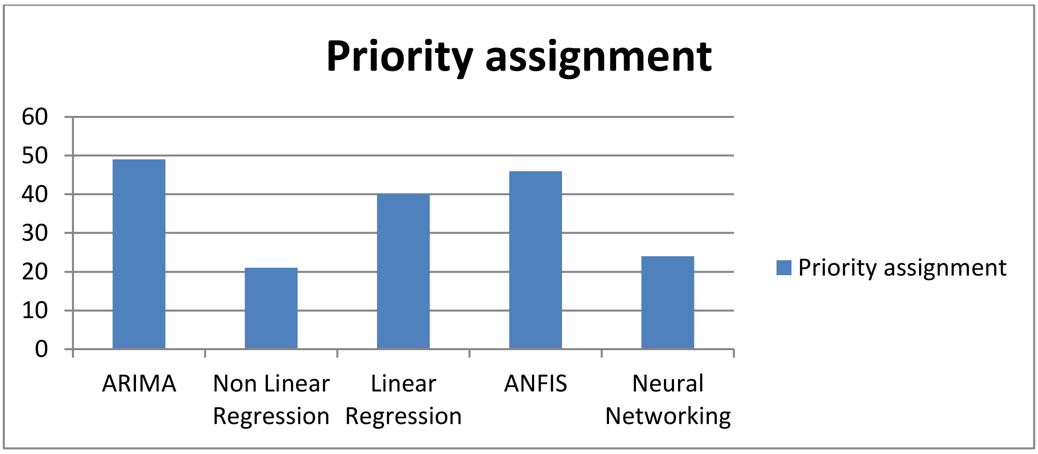


Figure 20. *Overall best fit evaluation*

1. The Prediction Accuracy of all the predictor models.

   Table 3, Table 4 and *Table 5*, shows the detailed accuracy evaluation. Ideally, the expected prediction accuracy A of any predictor models are 100%, i.e. 0% error or negligible error. However, due to the stochastic nature of the data under consideration, the predictor models will yield different prediction accuracies. In this thesis, the device switching patterns and electricity price patterns are assumed to be random variables. The considered device type is a flexible device with no human interventions or minimal human interventions. From the *Table 3, Table 4* and *Table 5* it is evident, that for the device type washing machine irrespective of the chosen predictor model and the chosen environmental test conditions, the prediction accuracy of ω_{Load} is greater than 97%, A_{Load} *> 97%*. However, for price data which is relatively more random than switching data under consideration, the prediction accuracy does vary significantly. From *Table 3, Table 4* and *Table 5* it is evident that the prediction accuracy of ω_{Price} vary from 0% - 60%, which is derived from the fact that the corresponding MAPE varies from 100% - 40% respectively. With all these results, the challenge lies in selecting the best predictor model for the DEMS, that fairs in each test condition.

2. The best predictor model for the electricity price prediction.

   A relatively good predictor model for electricity price prediction is selected by considering each predictor model's mean MAPE μ_{MAPE}^{year} and standard deviation over the year $H_{std(MAPE)}^{year}$ in varied test conditions. The price data is very random in nature due to this stochastic behavior, there exists no single predictor model which can provide the best fit irrespective of the test conditions (e.g. predictor model: linear regression, predicts with a relatively lower error than the peer for the city Anchorage and in contrary, it is not the best predictor for the city Palmadel or Little rock). Thus the intuitive approach of assigning a priority number to each predictor model according to the fit is adopted, i.e. in a considered test condition, the predictor model with the lowest MAPE $\mu_{MAPE}^{year} \longrightarrow 0$ and lowest standard deviation value $H_{std(MAPE)}^{year} \longrightarrow \mu_{MAPE}^{year}$, is set with a highest priority number 5 and assigned with a lower priority number, if the opposite is the case. Since there exist 5 predictor models under consideration, the priority value varies from 5 to 1, where 5 indicates a relatively better fit and 1 indicates a relatively worse fit. This process of priority assignment is carried out in all the possible test conditions as shown in *Table 8*. The calculation denotes that, the predictor model *ANFIS* tops the peer with a priority value of 42, ARIMA stands 2nd with priority value 33, Linear regression takes 3rd position by value 31, NN with 18 and Non-linear regression with 9 at the end. Thus *ANFIS,* relatively fits good for price prediction. $\mu_{MAPE|ANFIS}^{year} \leq \mu_{MAPE|ARIMA}^{year} \leq \mu_{MAPE|Lin\_Reg}^{year} \leq \mu_{MAPE|NN}^{year} \leq \mu_{MAPE|Non-Lin\_Reg}^{year}$.

3. The best fit predictor model for predicting the device switching patterns (Only Water pump).

   A relatively good predictor model for the device switching prediction can be inferred from *Table 3, Table 4* and *Table 5.* A relatively good predictor model is selected by considering each predictor model's mean accuracy and standard deviation over the year in various test conditions. The considered flexible device type is of zero or minimal human intervention. These device type is being operated for a fixed duration of time or until the completion of its specific task. Thus, it is relatively less stochastic and more deterministic in nature. Due to this reason, the device switching pattern is competitively predicted well by each predictor model considered, i.e. with the mean prediction accuracy greater than 97% over a year, i.e. $\mu_{accuracy|ARIMA}^{year} \approx \mu_{accuracy|NN}^{year} \approx \cdots \approx \mu_{accuracy|Non-Lin\_Reg}^{year} \approx 98\%$. However, the intuitive approach of assigning a priority number to each predictor model based on its fit, is applied for switching prediction as well, i.e. in a considered test condition, the predictor model with the mean accuracy closer to 100%, $\mu_{accuracy}^{year} \longrightarrow 100\%$ and least standard deviation value $H_{std(MAPE)}^{year} \longrightarrow \mu_{accuracy}^{year}$, is set with highest priority number 5 and assigned with a lower priority number, if the opposite is the case. Since there exist 5 predictor models under consideration, the priority value varies from 5 to 1, where 5 indicates a relatively better fit and 1 indicates a relatively worse fit. This process of priority assignment is carried out in all the possible test conditions as shown in *Table 8*. The calculation denotes that the predictor model *ARIMA* tops the peer with a priority value of 15, the nonlinear and linear regression models stands 2nd with the priority value 10, NN with 6 and ANFIS with 3 stands at the end. In case of device switching prediction, all the predictor models competitively perform well in all the test condition. The predictor models are reordered based on varied priority values. The predictor models are not assigned with equal priority, even if there exists a minimal difference in the prediction accuracy, i.e. even if it is in the order of <10^{-3} percent. Thus *ARIMA* is relatively the best predictor model for device switching prediction. $\mu^{year}_{accuracy|ARIMA} \geq \mu^{year}_{accuracy|Non-Lin\_Reg} \geq \mu^{year}_{accuracy|Lin\_Reg} \geq \mu^{year}_{accuracy|NN} \geq \mu^{year}_{accuracy|ANFIS}$.

5.2 MDP Evaluation

The MDP solving algorithm receives the predicted outputs (the ω_{Load} and the ω_{Price} in case of a Water pump, determined switching pattern and the ω_{Price} in case of Cooler and Heater). The main expectation from the MDP evaluation is identifying one predictor algorithm that helps to obtains an optimal ON-OFF scheduling policy for the next 24 h ω^*_{Load} (96 ON-OFF states) which reduces $C_t^{Total\_Day}$; considering temperature, Peak Price and device specific parameters irrespective of device, place and time of year.

One optimal policy ω^*_{Load} is obtained by solving MDP for the next 24 h with different waiting time, interruptions and at the same time verifying, whether the device switching prediction accuracy is greater than 97%, $A_{Load} \geq 97\%$ (in case of a Water pump). *Table 9* depicts the evaluation of each predictor model under consideration for an arbitrarily chosen date i.e. April 29th of 2015. The MDP evaluation investigates whether the optimized switching schedule satisfies the following concerns

1. Maximum allowed waiting for the device W_D^{Max}: This parameter is one of the operation and requirement critical specifications. The W_D^{Max} is arbitrarily chosen as *five* hours. The waiting time is presented in *Table 9* as the number of 15 minutes interval count, i.e. 20 sampling interval in five hours. However, the Maximum allowed waiting for the device can be chosen based on the device's specification and manufactures recommendations. Accordingly, the devices should be turned ON, before the device's present waited time W_D reaches the W_D^{Max}.

2. Maximum allowed interruptions of the device in a day is *two* times $INTR_D^{Max}$: This parameter is one of the safety critical specifications. The $INTR_D^{Max}$ is arbitrarily chosen as *two* times. However, the Maximum allowed interruptions of the device in a day can be chosen based on the device's specification and manufactures recommendations. Accordingly, the device can be curtailed when its within $INTR_D^{Max}$ count and when the present interruptions $INTR_D$ count reaches $INTR_D^{Max}$, the device should remain turned OFF.

3. The total execution time (duration for which the device is in ON state in a day) of the optimized schedule obtained from phase 2 must be same as that of the device's total execution time without the proposed algorithm, i.e. *MDP scheduled ON time = Occurred ON time*.

4. The total cost of energy consumption $C_t^{Total\_Day}$ of the optimized schedule obtained from phase 2 must be less than or equal to that of the device's $C_t^{Total\_Day}$ without the proposed algorithm, i.e. *CoEC from MDP scheduled switching ≤ Occurred CoEC*.

Once we obtain the optimal policy from one predictor algorithm (Ex: ARIMA: $\omega^*_{Load|ARIMA}$), a similar exercise is carried out for each predictor algorithm (Ex: Non Linear regression, Linear regression, NN, ARIMA and ANFIS: $\omega^*_{Load|Non-Lin\_Reg}$, $\omega^*_{Load|Lin\_Reg}$...) for each day of the year, choosing a different location and device type. From *Table 10* we see the wins of each predictor algorithm-MDP pair ($\omega*$ data resulting in lowest CoEC with good accuracy A) for the entire year, for a single device (Water pump) and for the chosen location, i.e. similar to *Table 9*, MDP evaluation exercise is carried out for 357 days and wins are captured for different device and chosen city for a year. The incapability indicates, that the predictor-MDP pair fails to provide device's switching prediction accuracy A_{Load} ≥ 97% and/or reduced forecasted CoEC than occurred CoEC, $C_t^{Total\_Day}|_{Occured} > C_t^{Total\_Day}|_{Predicted}$. In *Table 9* these incapabilities are denoted by *Inf*. The win count indicates, the number of times in a year corresponding predictor model with MDP has optimized the schedule relatively better. *Table 11* shows the winning pairs of different device type and different city. The Non-linear regression and ANFIS indicate their incapabilities to support MDP optimization in all the scenarios. Since the predictor model and MDP pair has to provide an optimal fit in all the scenarios, different Predictor model-MDP pairs are considered and evaluated. From *Table 11,* it is conclusive that the ARIMA-MDP pair wins, although the difference of wins would be marginal with immediate next alternative predictor model and MDP pair.

Table 9*. MDP evaluation for chosen date 29-4-2015*

	ANKORAGE					
	Water pump					
29-4-15	**Max Waiting Time (Samples 15 Min Interval)**	**Max Interruptions**	**MDP Scheduled ON Time (Samples 15 Min Interval)**	**Occurred ON Time (Without MDP) (Samples 15 Min Interval)**	**CoEC from MDP scheduled switching in cents**	**Occurred CoEC in cents**
ARIMA	20	2	57	57	1.2768	1.7046
Non Linear Regression	*Inf*	*Inf*	*Inf*	*Inf*	*Inf*	
Linear Regression	20	2	57	57	1.2979	
ANFIS	*Inf*	*Inf*	*Inf*	*Inf*	*Inf*	
Neural Networking	10	2	57	57	1.2934	

***Table 10**. Wins and incapability of all predictors for Water pump as load, in Anchorage in the year2015*

ANKORAGE		
Water pump		
Year 2015	**Wins**	**Incapability**
ARIMA - MDP	6	
Non Linear Regression - MDP	2	-8
Linear Regression - MDP	4	
ANFIS - MDP	2	-3
Neural Networking - MDP	6	

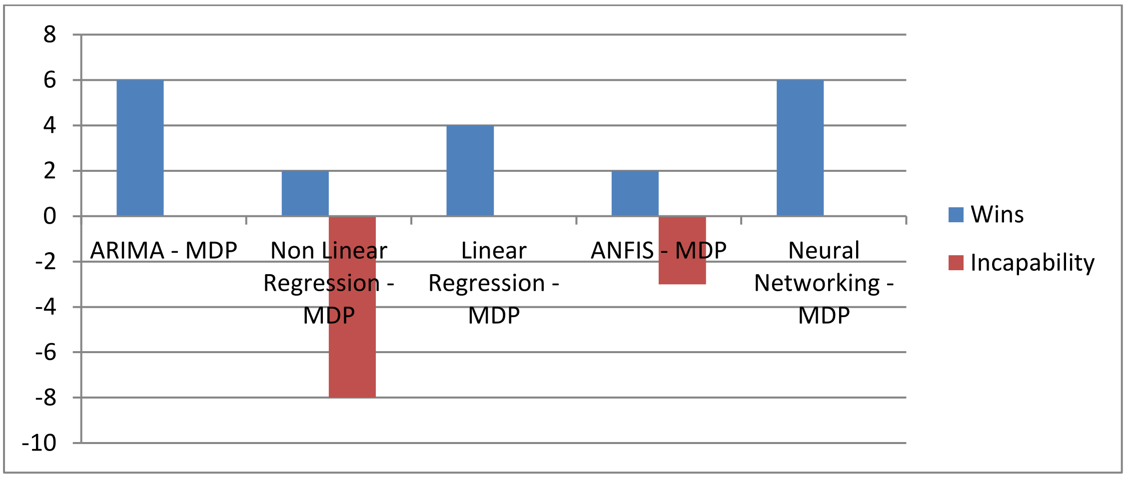


***Figure 21.** Wins and incapability of all predictor-MDP pair for Water pump, in Anchorage in the year2015*

***Table 11.** The resulting Predictor-MDP pair for different scenarios over the year*

Winners	**ANKORAGE**	**LITTLE_ROCK**	**PALMADEL**
Water pump	ARIMA-MDP	ARIMA-MDP	ARIMA-MDP
Heater	ARIMA-MDP	ARIMA-MDP	ARIMA-MDP
Cooler	ARIMA-MDP	ARIMA-MDP	ARIMA-MDP

4. The optimization of device switching schedule using MDP, i.e. the load shift achieved by considering peak price in a day and many other aspects, as compared to a schedule without MDP.

An introduction to MDP, its role in this thesis and its implementation details are provided in SECTION IV. Accordingly, MDP has been employed to optimize the phase 1 output, i.e. to optimize the prediction result of any predictor model ω^{*}_{Load}. The effect of optimization is a reduced CoEC per day (next 24 h is under consideration) than a CoEC resulting from any regular switching pattern, as predicted at phase 1, $C_t^{Total\_Day}|_{Predicted} < C_t^{Total\_Day}|_{Occured}$. This optimization, i.e the load shift also takes several influencing parameters into consideration as narrated in SECTION IV. There exists a significant price reduction using MDP, i.e. as shown in *Table 9,* there exists a 39.35% of $C_t^{Total\_Day}|_{Predicted}$ reduction w.r.t occurred CoEC. However, this magnitude of reduction is not the same for each day of the year. This scale of reduction is influenced by the accuracy of the chosen predictor model. Thus the scale of CoEC reduction is inconsistent, however, there exists a reduction in CoEC of optimized schedule ω_{Load}^{*}.

5. The final outline of the proposed DEMS algorithm.

Table 9 indicates a fact that MDP is reducing the CoEC by optimizing the ON-OFF schedule. From *Table 3, Table 4* and *Table 5,* the derived inference is that the preferred predictor model for price prediction ω_{Price} is ANFIS and the preferred predictor model for device's switching prediction ω_{Load} is ARIMA. However, w.r.t the price prediction accuracy, ARIMA stands as the 2nd best and on the other hand, the device's switching prediction accuracy of ANFIS is relatively the worst among its peers. For that reason, ARIMA is referred in this thesis as a best-fit predictor model. *Table 10* and *Table 11* indicates that ARIMA in conjunction with MDP is providing a cost-effective and optimized switching schedule i.e

$$C_t^{Total\_Day}|_{ARIMA-MDP} \leq C_t^{Total\_Day}|_{NN-MDP} \leq C_t^{Total\_Day}|_{Lin\_Reg-MDP} \leq C_t^{Total\_Day}|_{ANFIS-MDP} \leq C_t^{Total\_Day}|_{Non-Lin\_Reg-MDP}.$$

Accordingly, the outline of the algorithm is *"ARIMA predictor model at phase 1 and MDP at phase 2 and continued with phase 3, having microcontroller I/O pins to control the device action: ON or OFF "*.

6. Conclusion

In this thesis, a new modeling approach for DEMS is proposed. The proposed work is one of the possible approaches to reduce the CoEC $C_t^{Total\_Day}$, to improve the efficiency and the effectiveness of power usage η_{Power} at the device level. An essential understanding of the problem statement is achieved, i.e. to eliminate the modern day critical electrical aspects such blackout, brownouts and other similar concerns, with an approach which needs less infrastructure and implementation costs. For that purpose, in this thesis, a 3 layered algorithm is proposed to reduce the $C_t^{Total\_Day}$. The proposed algorithm could be readily embedded into the existing or new flexible devices at a negligible cost since the algorithm will be provided as a software update and is implemented in the existing device microcontroller.

The 3 layers of the algorithm are phase 1: RTM, phase 2: STS and phase 3: RTC, respectively. The primitive intentions of phase 1 are to predict the device's switching pattern and electricity price pattern for the next $24h$. In phase 2 the predicted device switching pattern is optimized by considering the device type, the day's temperature (if temperature dependent), peak electricity price, the device specifications such as its maximum allowable waiting time, maximum allowable interruptions in a day, the total power requirement in a day etc., as described in SECTION IV. After a quality evaluation of the proposed algorithm on several flexible devices in various cities, the magnitude of $C_t^{Total_Day}$reduction, inferred that the MDP provides a cost-effective and an optimized switching schedule.

The 6 predictors employed are shown in *Table 1. List of Predictor models* and its accuracy is tested in various scenarios as explained in SECTION V. Irrespective of the various test conditions(e.g.: different device type or different city etc.), the evaluation is performed over the entire year, i.e. for the year 2015. A relative best-fit predictor model is decided by considering the mean prediction accuracy and the standard deviation of the prediction accuracy. The predictor models with relatively high yearly mean accuracy value and the low standard deviation is inferred to be the best fit. For electricity price prediction and device switching prediction, ARIMA is relatively predicting better in each test case compared to its peer. The MDP reschedules the device's switching pattern considering several influencing parameters (e.g.: lower electricity price, daily required Power etc.).The optimized switching schedule obtained from the MDP is depending on the accuracy of the predictor model as well. The switching schedule obtained from different MDP and predictor model pairs, are evaluated and compared. From SECTION V it is evident that the ARIMA model followed with the MDP provides a cost-effective schedule, irrespective of the chosen city, season of the year and the device type. Apart from ARIMA-MDP pair, the other predictor model and MDP pairs either provide a relatively higher CoEC schedule or fail to provide a schedule for a given device and city throughout the year. To conclude, the ARIMA predictor model with MDP provides a relatively optimum schedule in various scenarios.

The future work will focus on methods to optimize the modeling and prediction of the chaotic and nonlinear time series. The work will investigate the ways to understand the electricity price dynamics and improve its prediction accuracy. The efforts include further optimization of the algorithm for robust performance and faster execution time. The work will address the challenges to reformulate the MDP to further reduce the cost of energy consumption by tracking the lowest electricity price at first.

7. Reference

[1] *Geir Warland, Birger Mo :"Stochastic optimization model for detailed long-term hydro thermal scheduling using scenario-tree simulation". SINTEF Energy Research, Sem Sælands vei 11, Trondheim 7034,Norway.*

[2] *M. Erol-Kantarci and H. T. Mouftah, "Wireless sensor networks for cost-efficient residential energy management in the smart grid," Smart Grid, IEEE Transactions on, vol. 2, no. 2, pp. 314–325, 2011*

[3] *Tùng T. Kim, Member, IEEE, and H. Vincent Poor, Fellow, IEEE : "Scheduling Power Consumption With Price Uncertainty", IEEE TRANSACTIONS ON SMART GRID, VOL. 2, NO. 3, SEPTEMBER 2011 519*

[4] *K. M. Tsui and S. C. Chan, Member, IEEE :"Demand Response Optimization for Smart HomeScheduling Under Real-Time Pricing", IEEE TRANSACTIONS ON SMART GRID, VOL. 3, NO. 4, DECEMBER 2012.*

[5] *Murat Kuzlu , Manisa Pipattanasomporn, Saifur Rahman:"Communication network requirements for major smart grid applications in HAN, NAN and WAN". Virginia Tech, Advanced Research Institute, Arlington, VA 22203, USA*

[6] *Cynthujah Vivekananthan, Student Member, IEEE, Yateendra Mishra, Member, IEEE, and Fangxing Li, Senior Member, IEEE "Real-Time Price Based Home Energy Management Scheduler." IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 30, NO. 4, JULY 2015 2149*

[7] *Understanding Kalman filter: Accessed on 2.4.2017 :: under the address : https://de.mathworks.com/videos/series/understanding-kalman-filters.html*

[8] *SAS Institute Inc. 2004. SAS/ETS 9.1 User's Guide.Cary, NC: SAS Institute Inc. SAS/ETS 9.1 User's Guide Copyright © 2004, SAS Institute Inc., Cary, NC, USA*

[9] *JOHN WILEY & SONS, INC.:"Nonlinear Regression Analysis and Its Applications" Second edition, Douglas M. Bates and Donald G. WattsA Wiley-Interscience Publication.*

[10] *J. Doyne Farmer and John J. Sidorowich : "Predicting chaotic time series", Accessed on: 24-8-1987, physical review letters, las alamos, newmexico 87545*

[11] *Neural Networking : Accessed on : 5.4.2017 :: under the address : http://www.obitko.com/tutorials/neural-network-prediction/prediction-using-neural-networks.html*

[12] *Z. Dziong.: "ATM Network Resource Management". McGraw-Hill publications, 1997.*

[13] *Electricity generation, transmission and distribution, Accessed on : 5:5:2017 :: under the address https://www.eia.gov/energyexplained/index.cfm?page=electricity_delivery, Last updated: August 31, 201*

[14] *Linear regression, Accessed on : 5:5:2017 :: under the address https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/linear_regression.*

[15] *ARIMA and ARIMA box jenkins : Accessed on : 5:5:2017 :: under the address http://www.statsref.com/HTML/index.html?arima.html*

[16] *Maysam Behmanesh, Majid Mohammadi, Vahid Sattari Naeini "Chaotic Time Series Prediction using Improved ANFIS with Imperialist Competitive Learning Algorithm". Blue Eyes Intelligence Engineering & Sciences Publication Pvt. Ltd. September 25th :2014*

[17] *Frank Z. Xing, Xiaomei Zou and Erik Cambria: "Predicting Evolving Chaotic Time Serieswith Fuzzy Neural Networks"*

[18] *Jesus Soto, Oscar Castillo and José Soria : "Chaotic Time Series Prediction Using Ensembles of ANFIS", springerlink.com © Springer-Verlag Berlin Heidelberg 2010*

[19] *Colin Fyfe, "Artificial Neural Networks and Information Theory", Department of Computing and Information Systems, The University of Paisley. Edition 1.2, 2000.*

[20] *Jouni Helske, "Prediction and interpolation of time series by state space models" Department of Mathematics and Statistics,University of Jyväskylä,2015.*

[21] *A. Soofi and L. Cao, "Nonlinear Deterministic Modelling and Forecastingof Economic and Financial Time Series", 15 September 2000.*

[22] *D. Kugiumtzis, "State Space Local Linear Prediction", Department of Statistics, University of Glasgow, Glasgow G12 8QW, UK.*

[23] *Anders Gjelsvik, Birger Mo and Arne Haugstad :"Long- and Medium-term Operations Planningand Stochastic Modelling in Hydro-dominated Power Systems Based on Stochastic Dual Dynamic Programming"*

[24] *Nicole B, Auerle and Ulrich Rieder : " MARKOV DECISION PROCESSES", on July 06, 2010.*

[25] *Stationarity and differencing: Accessed on : 5:5:2017 :: under the address https://people.duke.edu/~rnau/411diff.htm*

[26] *Dr. Roman V Belavkin," Lecture 11:Feed-Forward Neural Networks", Accessed on October 10, 2017.*

[27] *JIN Xue-bo, WANG Jiang-feng, ZHANG Hui-yan, CAO Li-hong: "ANFIS Model for Time Series Prediction", College of Computer and Information Engineering, Beijing Technology and Business University, Beijing, China, Article · August 2013*

[28] *Shumway and stoffer: " Time series analysis and its application: with R example", Edition 4, New York : Springer, 2016*

[29] *MATLAB: https://de.mathworks.com/*

8. Appendix

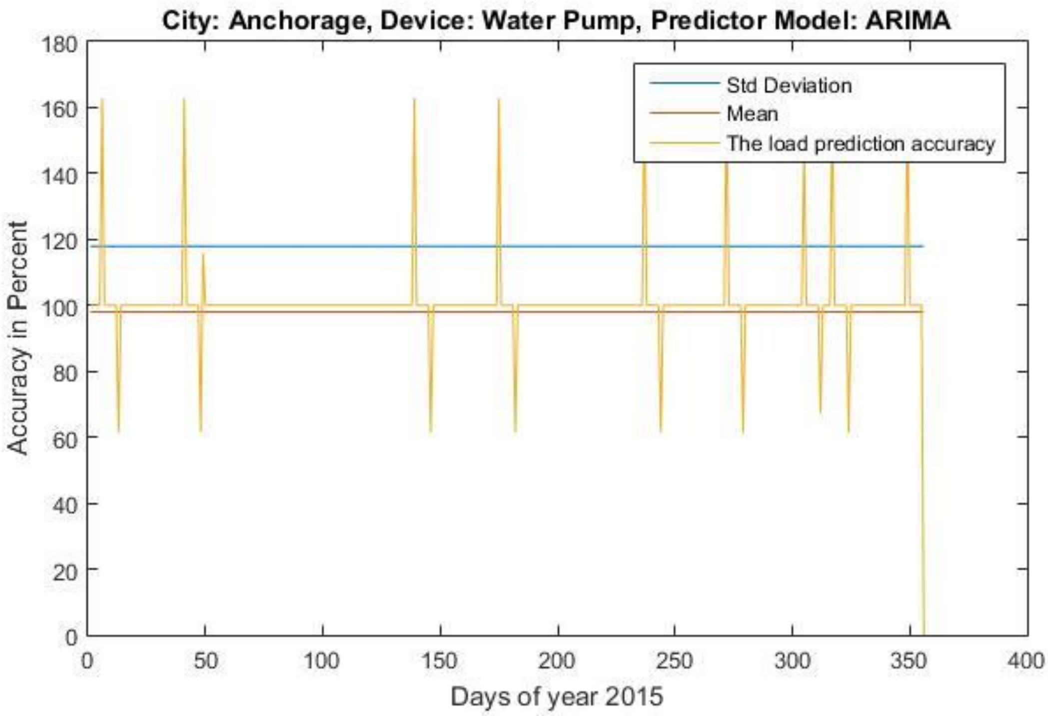


Figure 22 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Anchorage, Predictor model: ARIMA)*

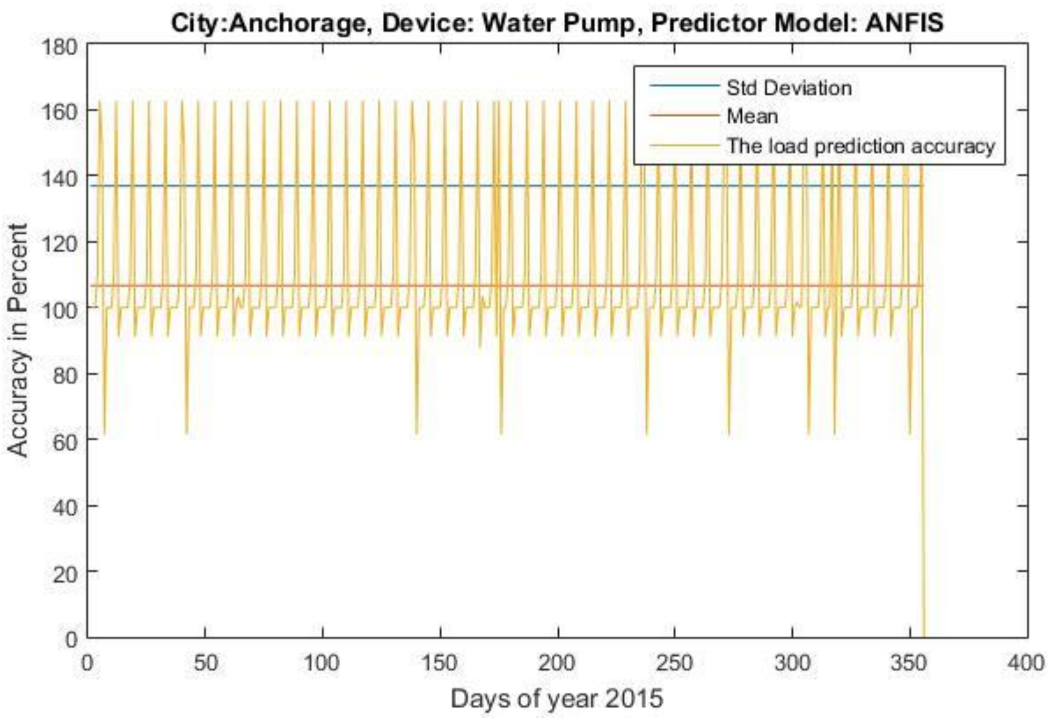


Figure 23 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Anchorage, Predictor model: ANFIS)*

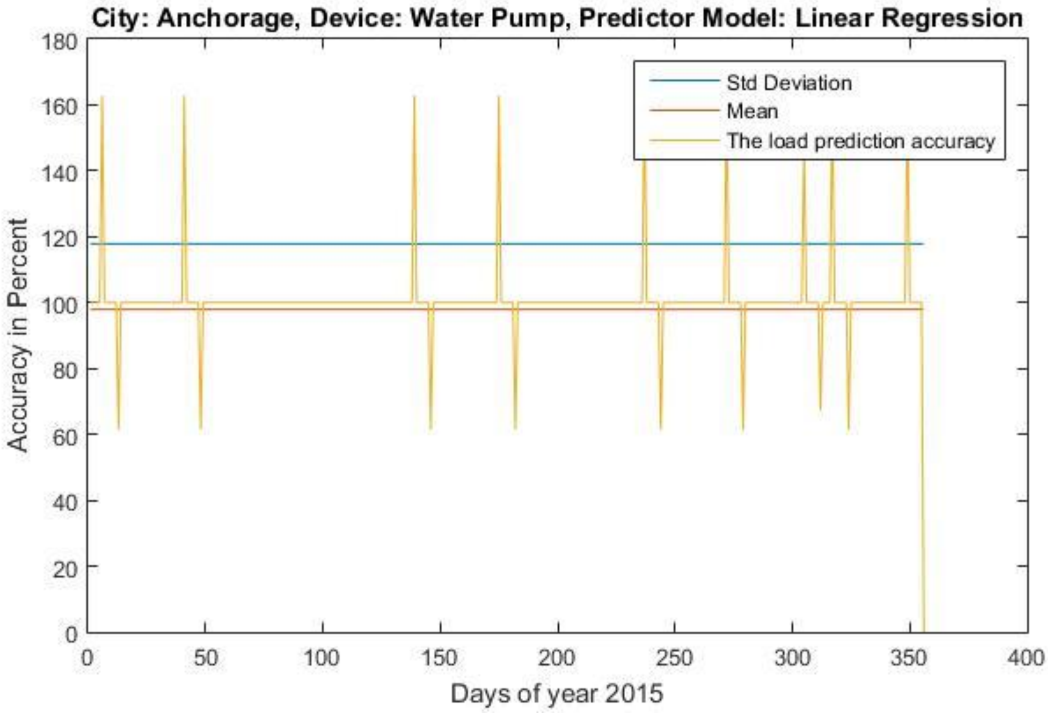


Figure 24 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Anchorage, Predictor model: Linear regression)*

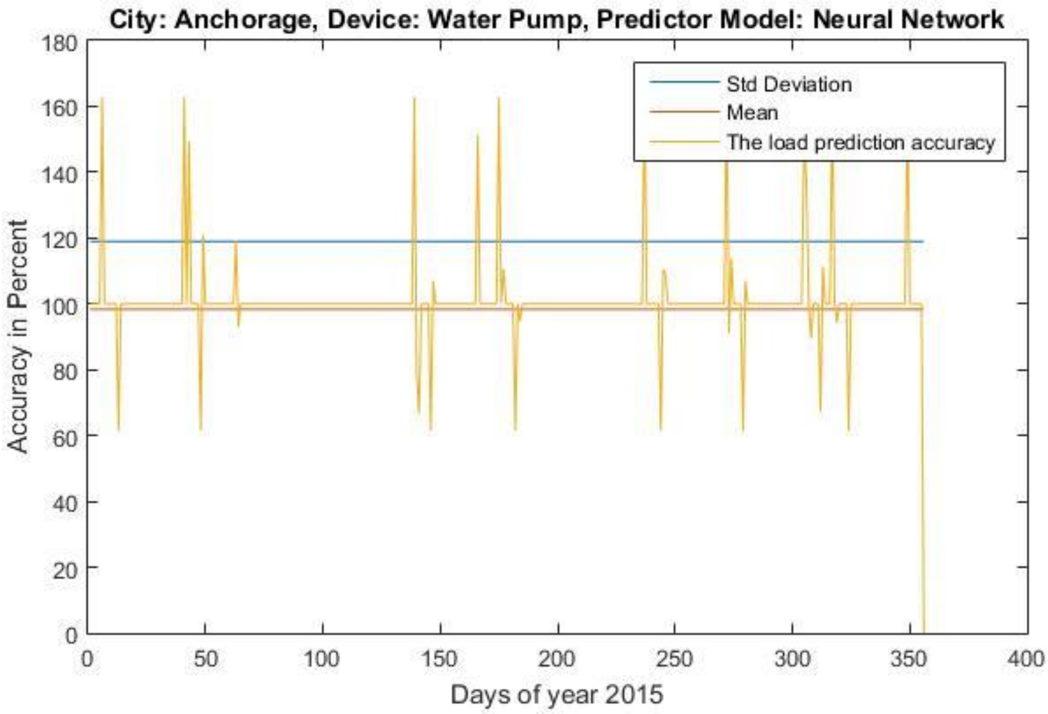


Figure 25 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Anchorage, Predictor model: Neural Network)*

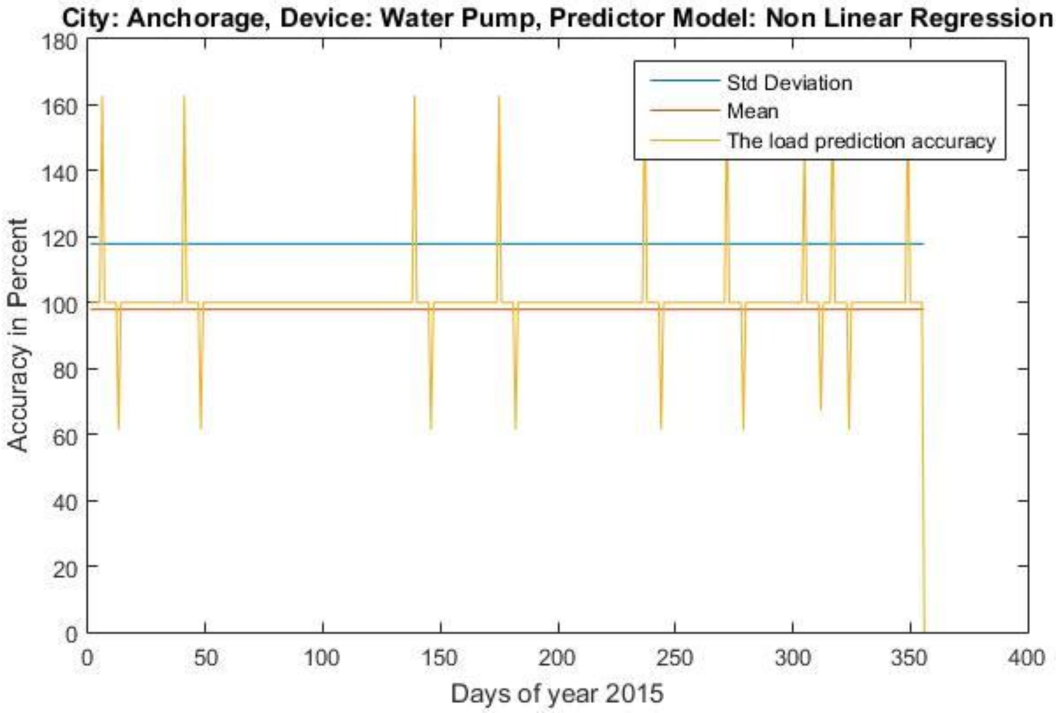


Figure 26 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Anchorage, Predictor model: Nonlinear regression)*

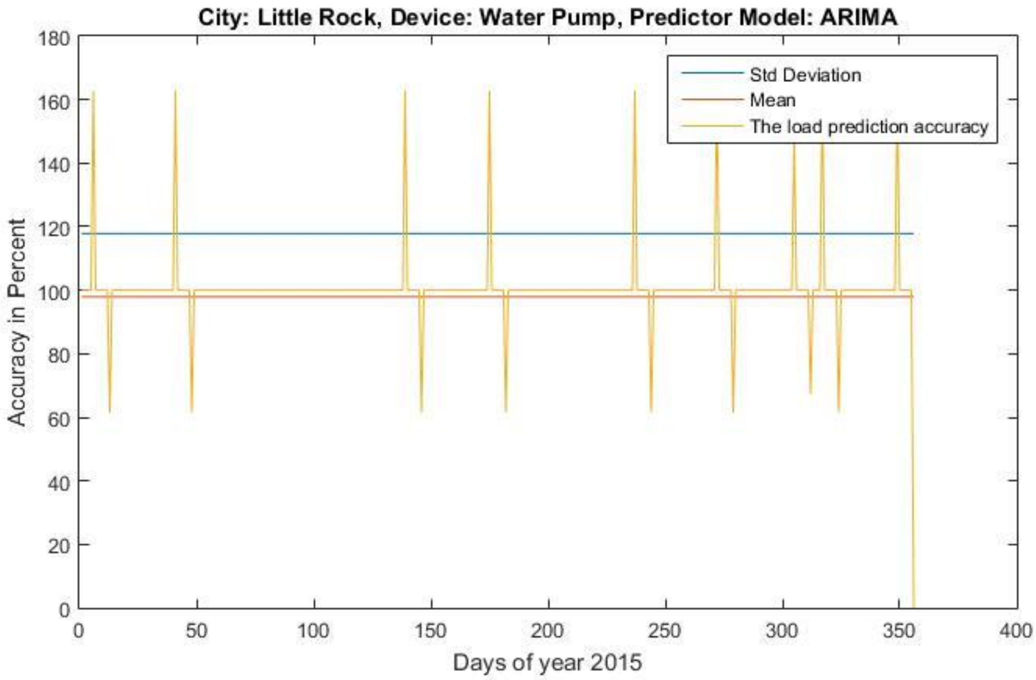


Figure 27 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Little Rock, Predictor model: ARIMA)*

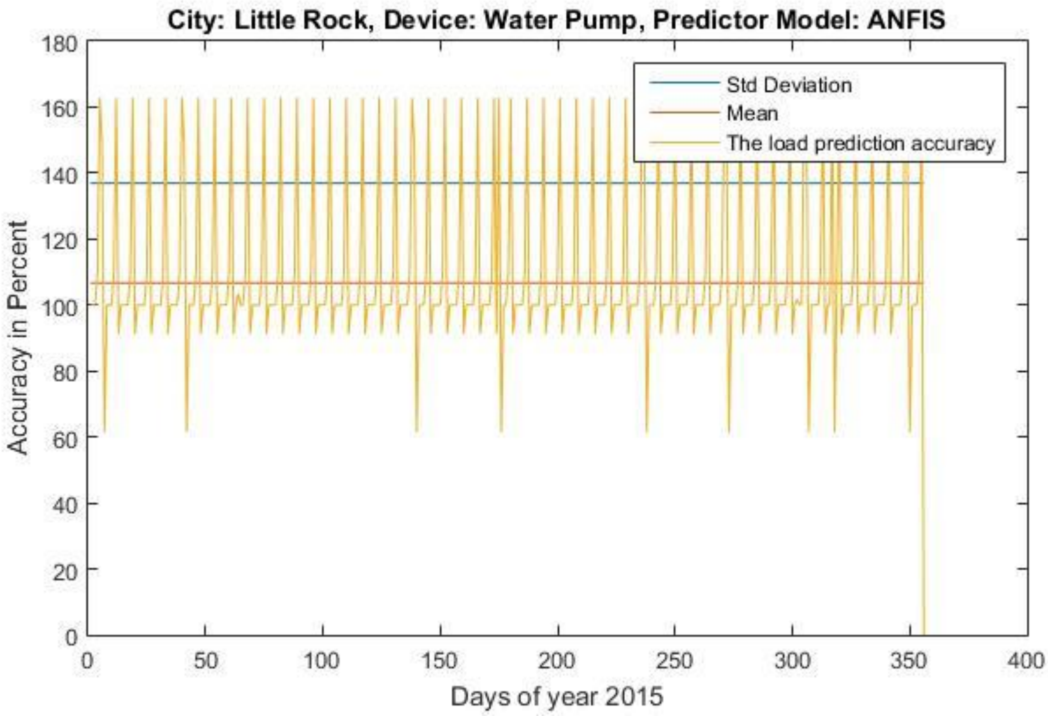


Figure 28 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Little Rock, Predictor model: ANFIS)*

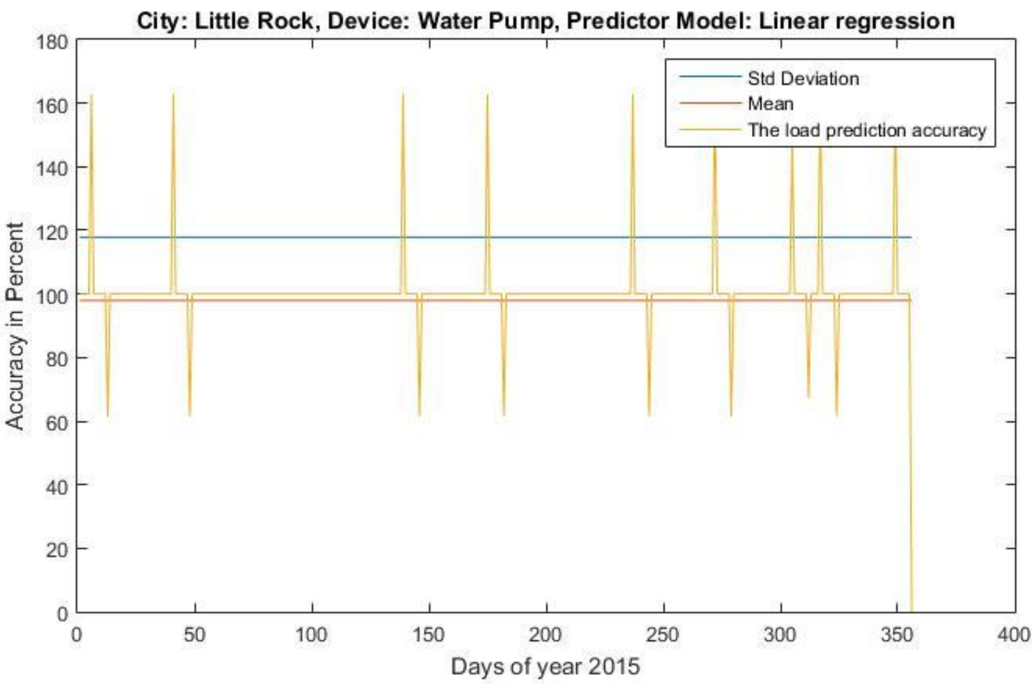


Figure 29 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Little Rock, Predictor model: Linear regression)*

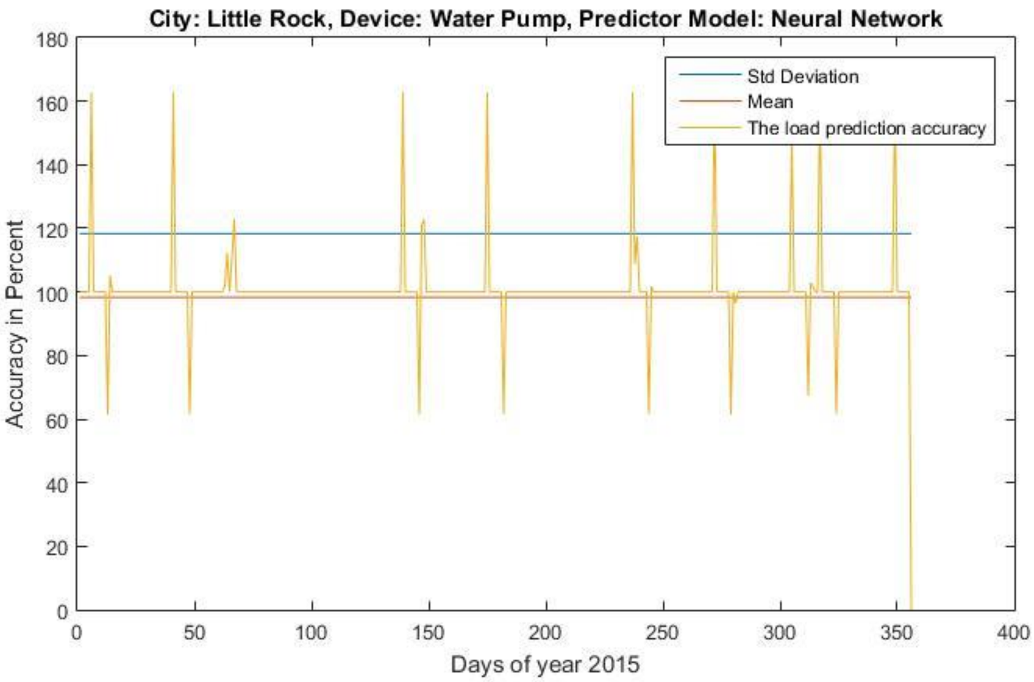


Figure 30 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Little Rock, Predictor model: Neural Network)*

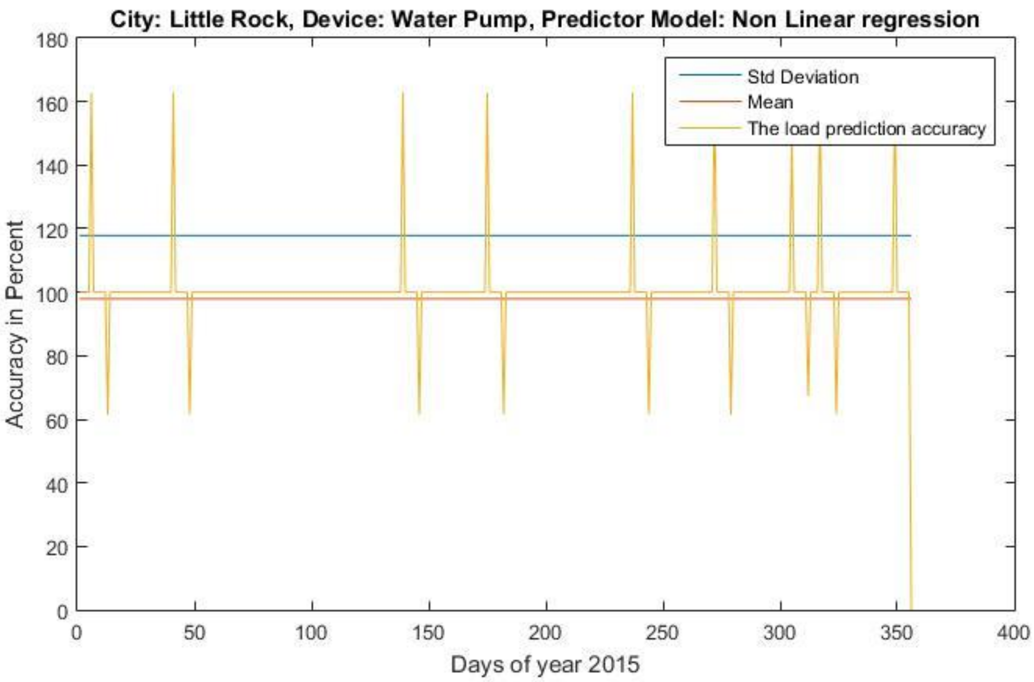


Figure 31 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Little Rock, Predictor model: Nonlinear regression)*

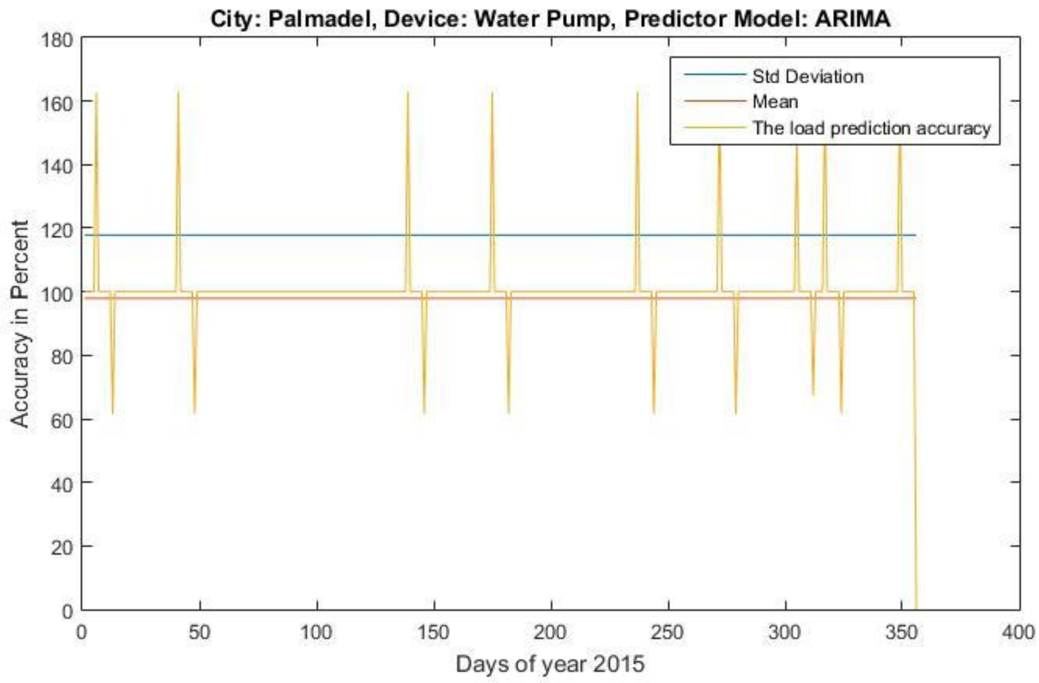


Figure 32 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Palmadel, Predictor model: ARIMA)*

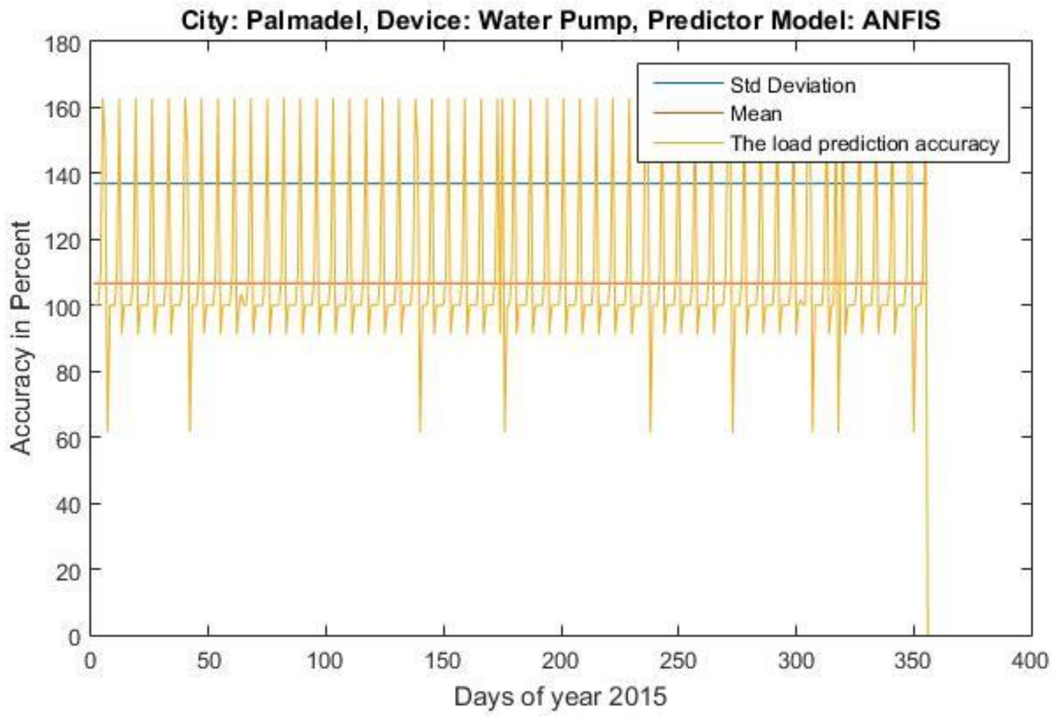


Figure 33 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Palmadel, Predictor model: ANFIS)*

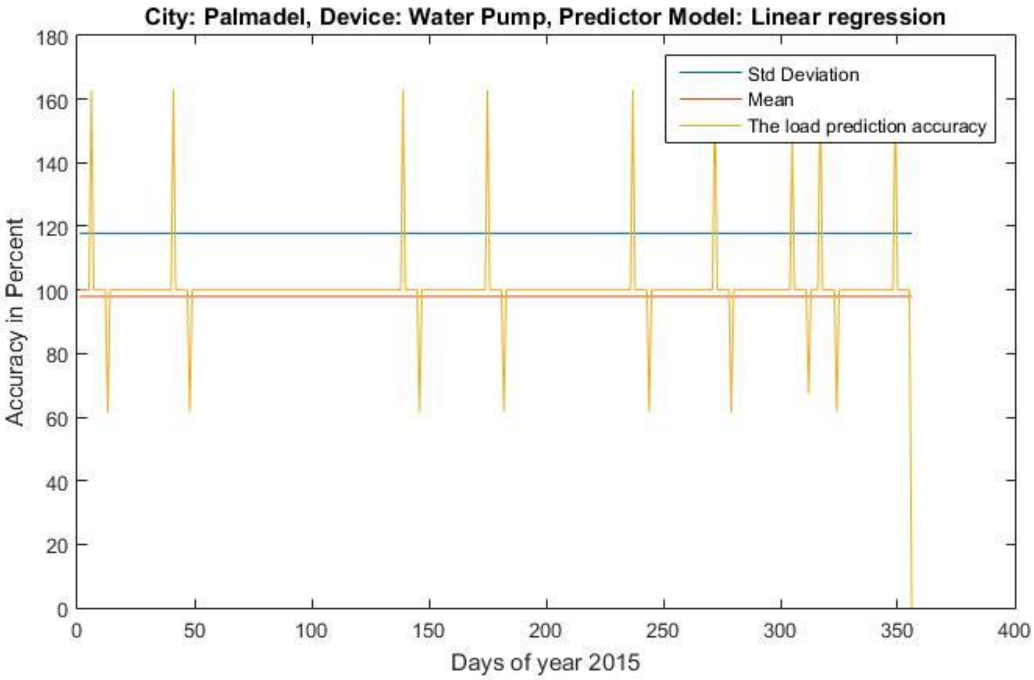


Figure 34 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Palmadel, Predictor model: Linear regression)*

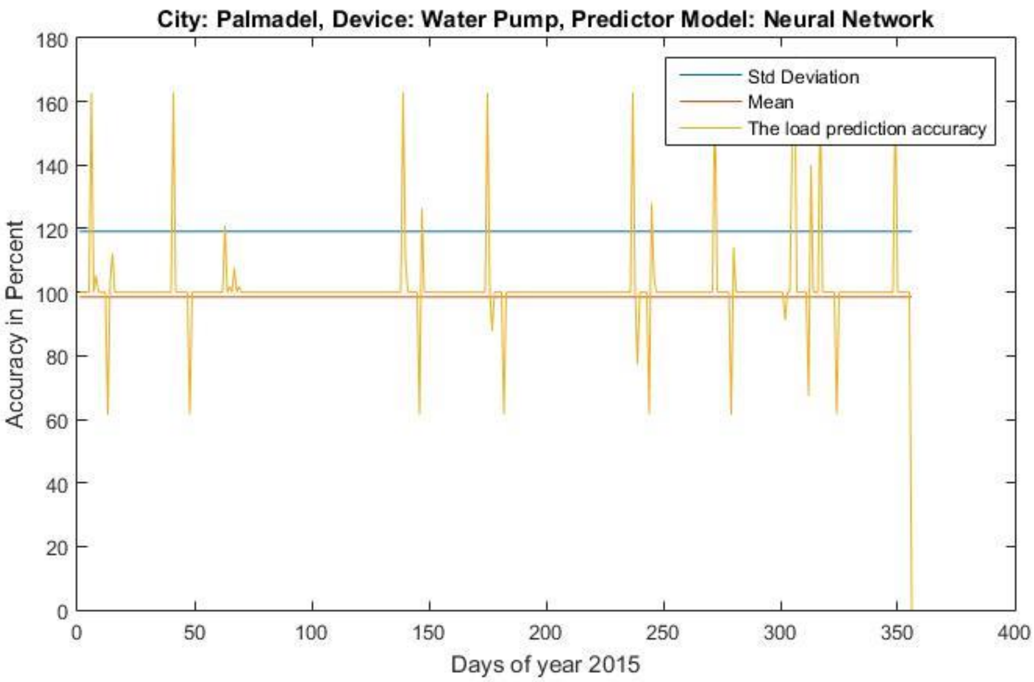


Figure 35 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Palmadel, Predictor model: Neural Network)*

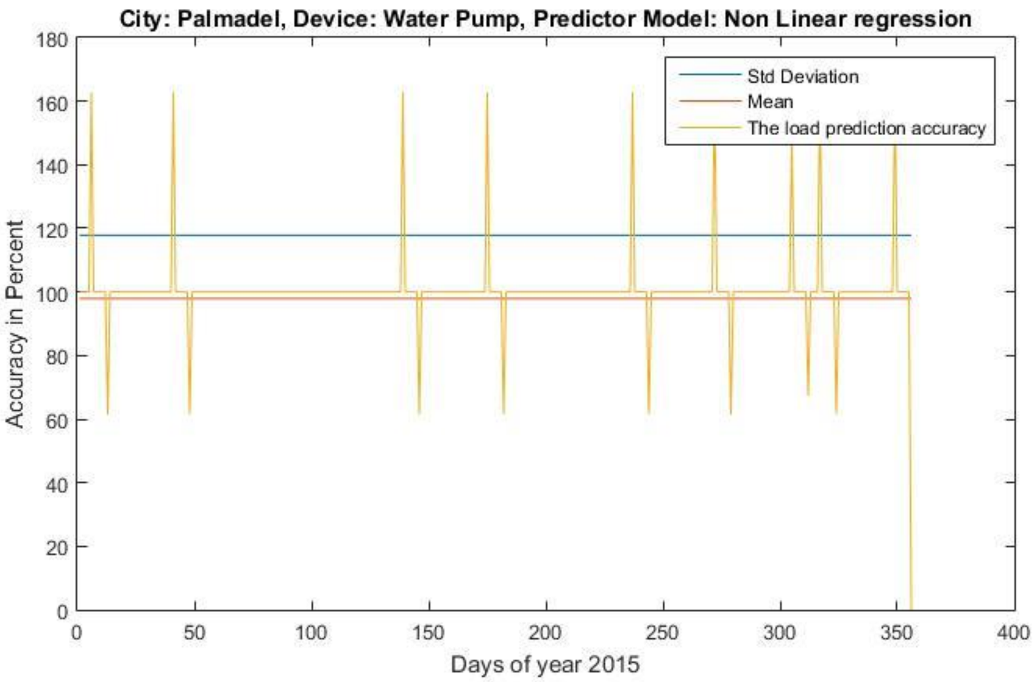


Figure 36 *Device Switching prediction accuracy for the considered case (Device: Water pump, City: Palmadel, Predictor model: Nonlinear regression)*

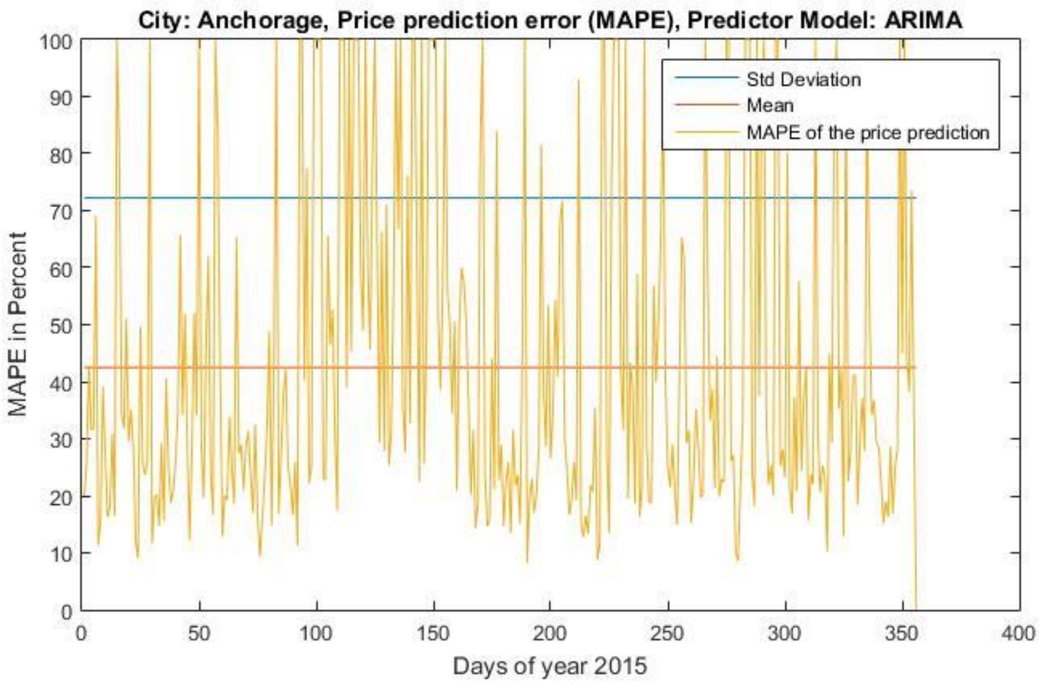


Figure 37 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Anchorage, Predictor model: ARIMA)*

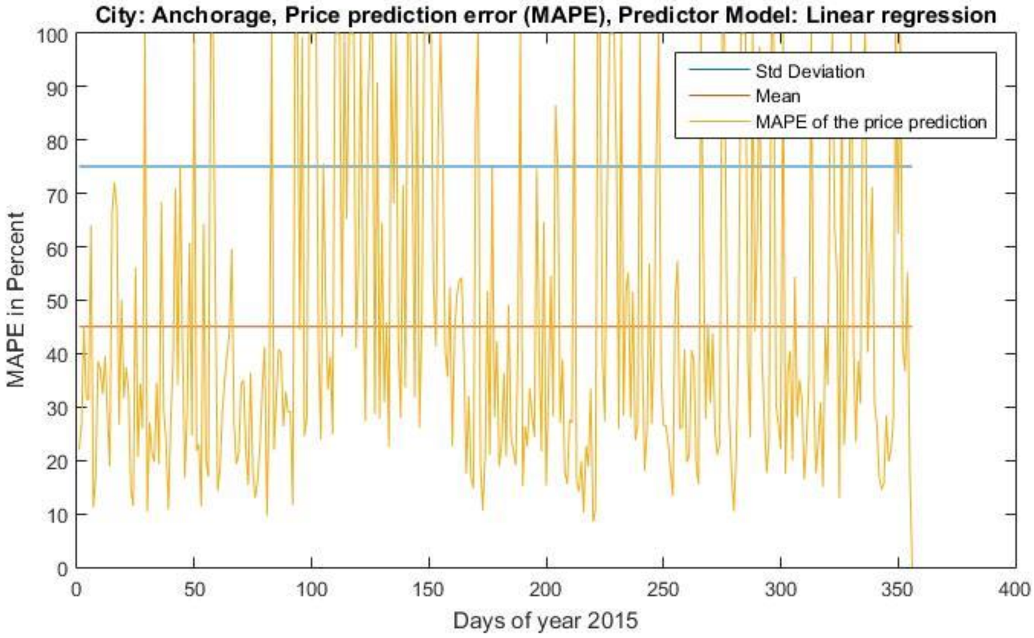


Figure 38 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Anchorage, Predictor model: Linear regression)*

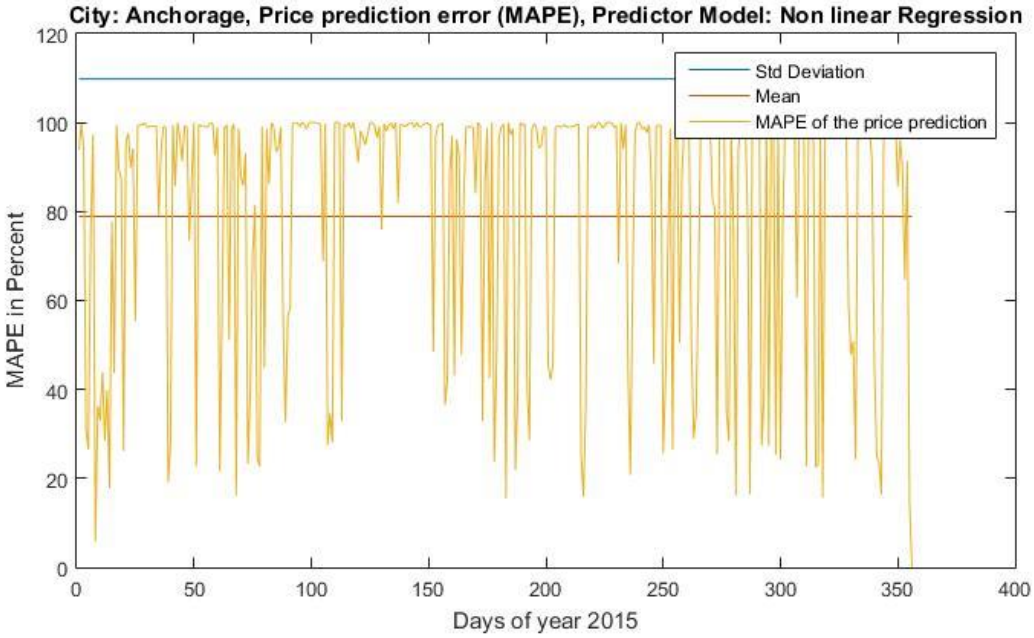


Figure 39 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Anchorage, Predictor model: Nonlinear regression)*

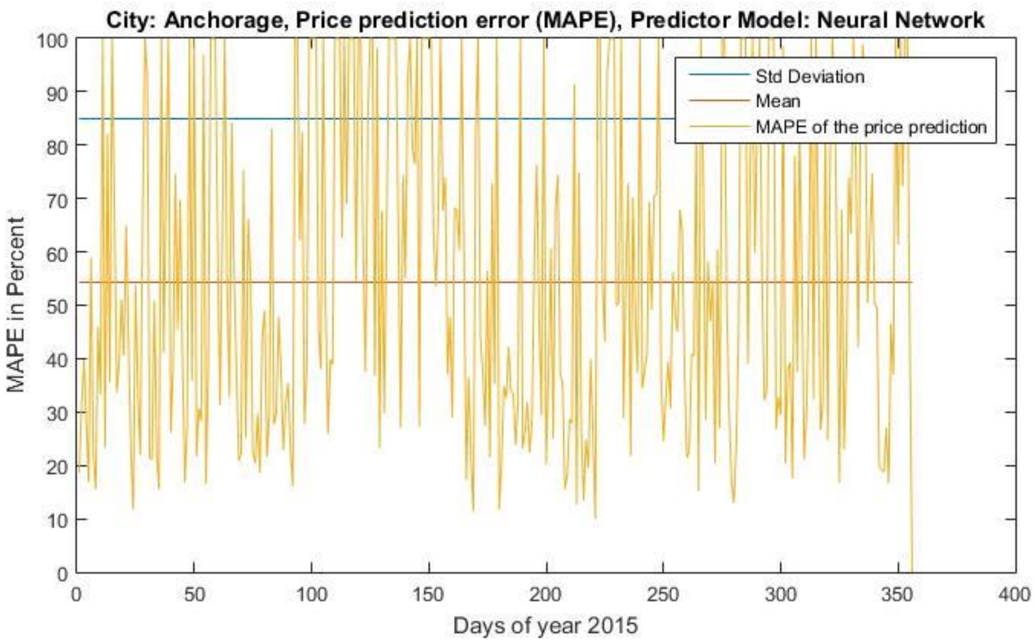


Figure 40 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Anchorage, Predictor model: Neural Network)*

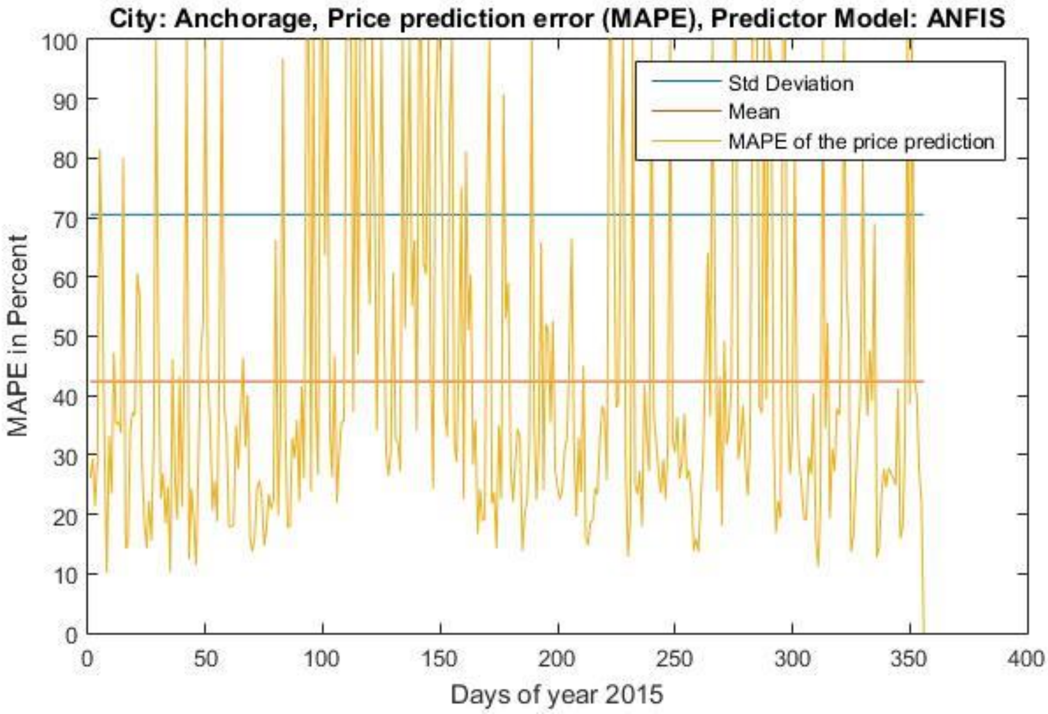


Figure 41 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Anchorage, Predictor model: ANFIS)*

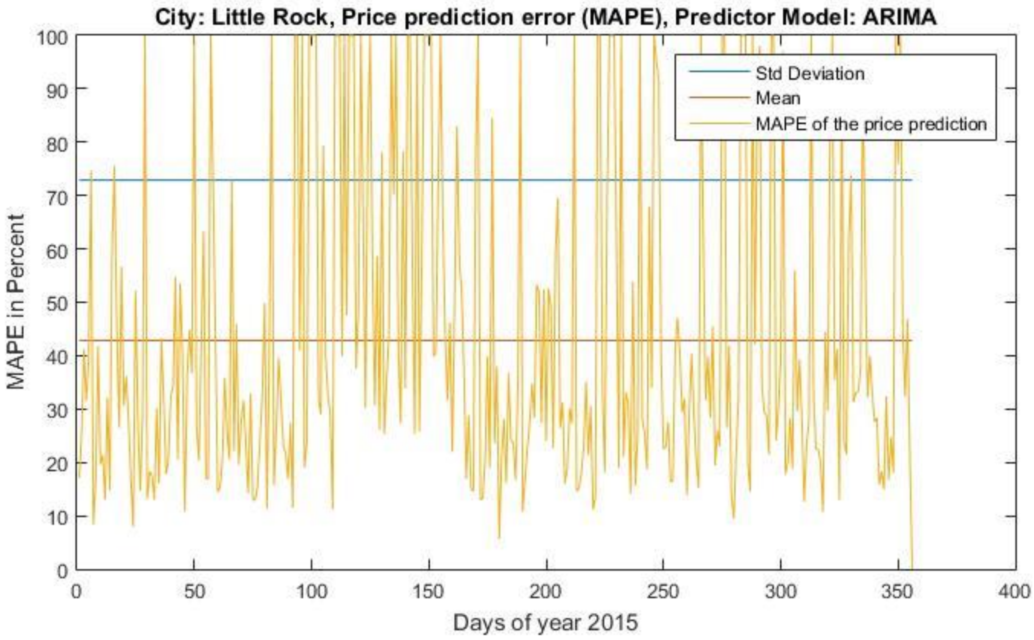


Figure 42 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Little Rock, Predictor model: ARIMA)*

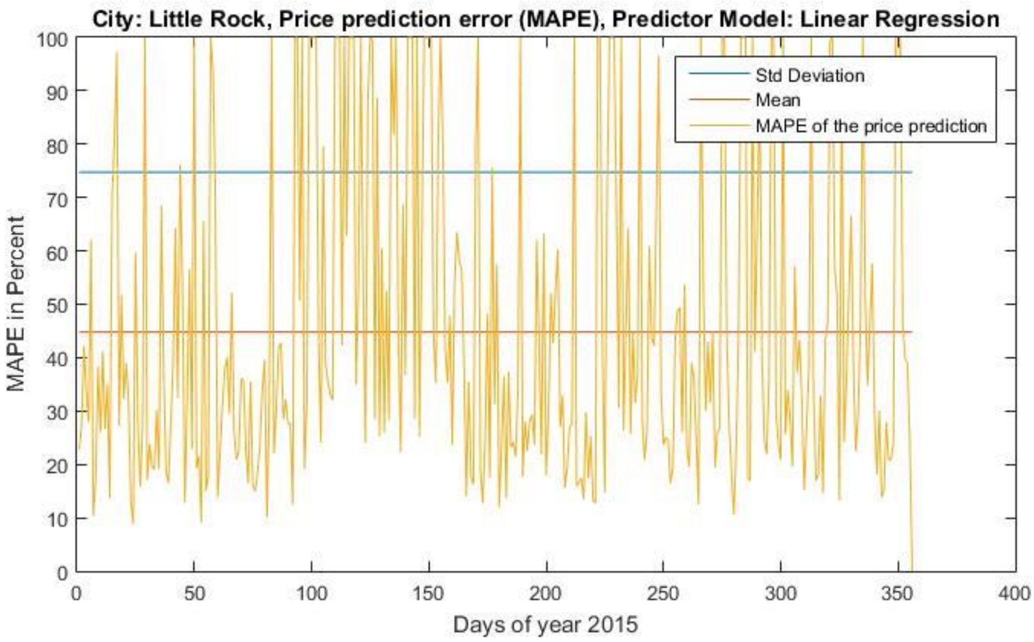


Figure 43 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Little Rock, Predictor model: Linear regression)*

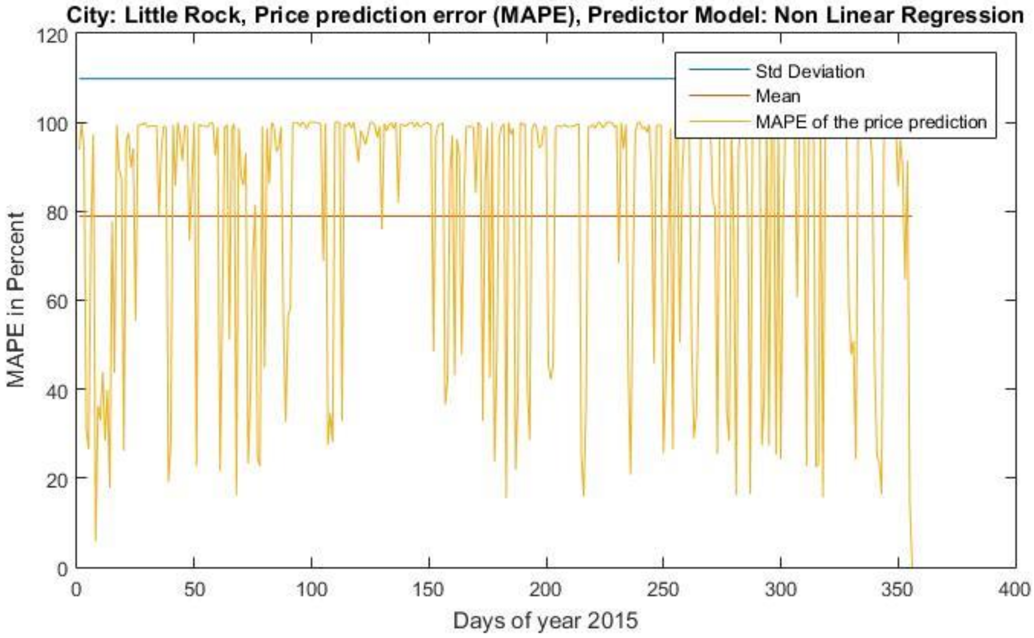


Figure 44 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Little Rock, Predictor model: Nonlinear regression)*

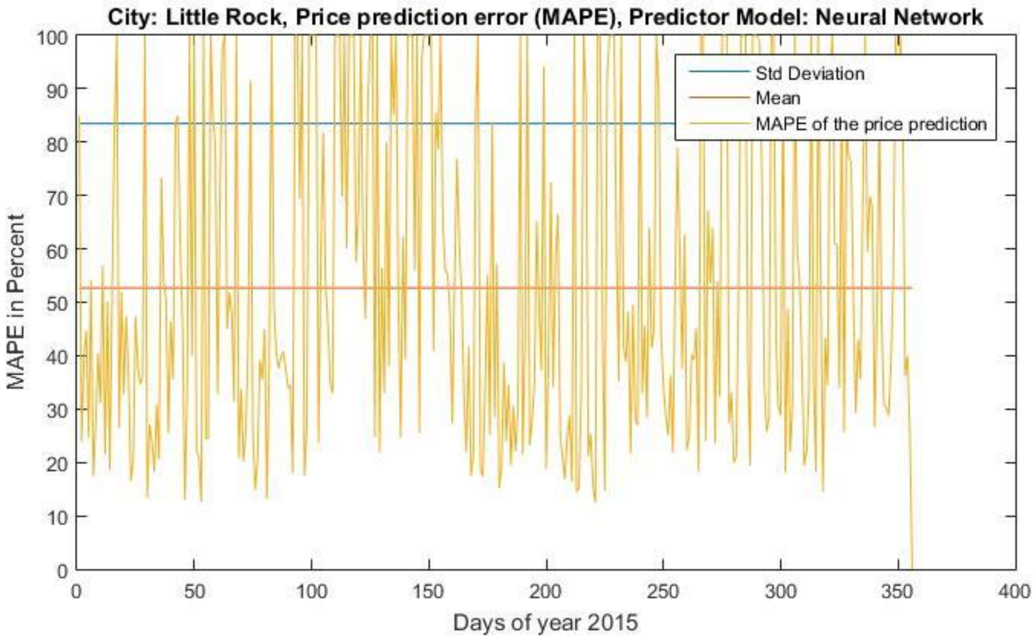


Figure 45 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Little Rock, Predictor model: Neural Network)*

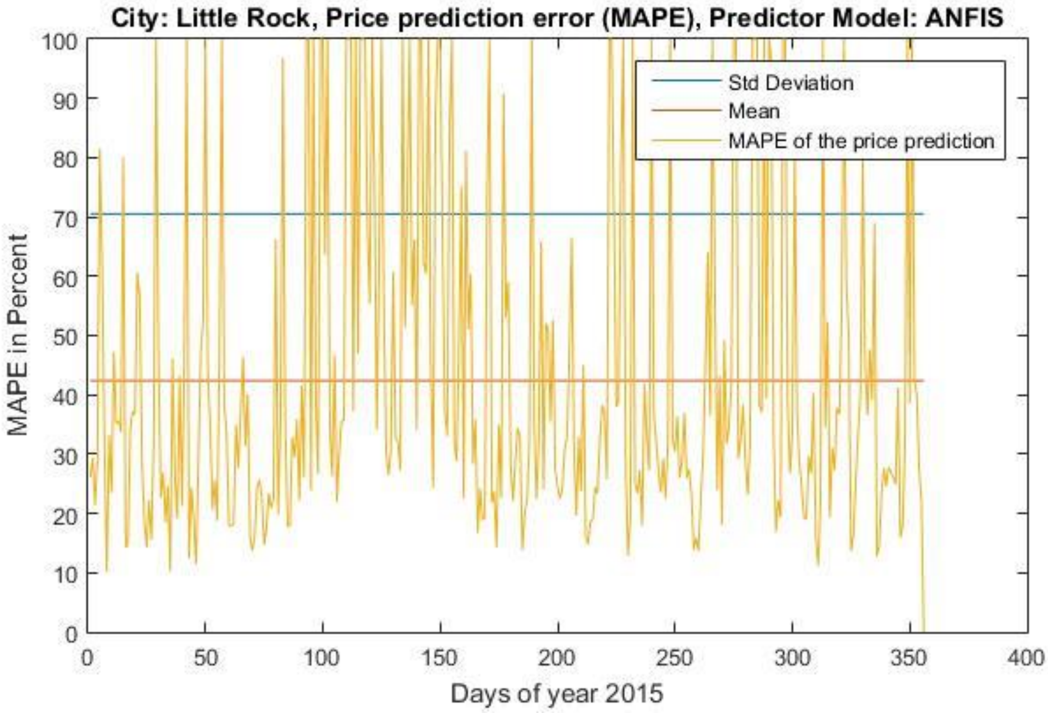


Figure 46 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Little Rock, Predictor model: ANFIS)*

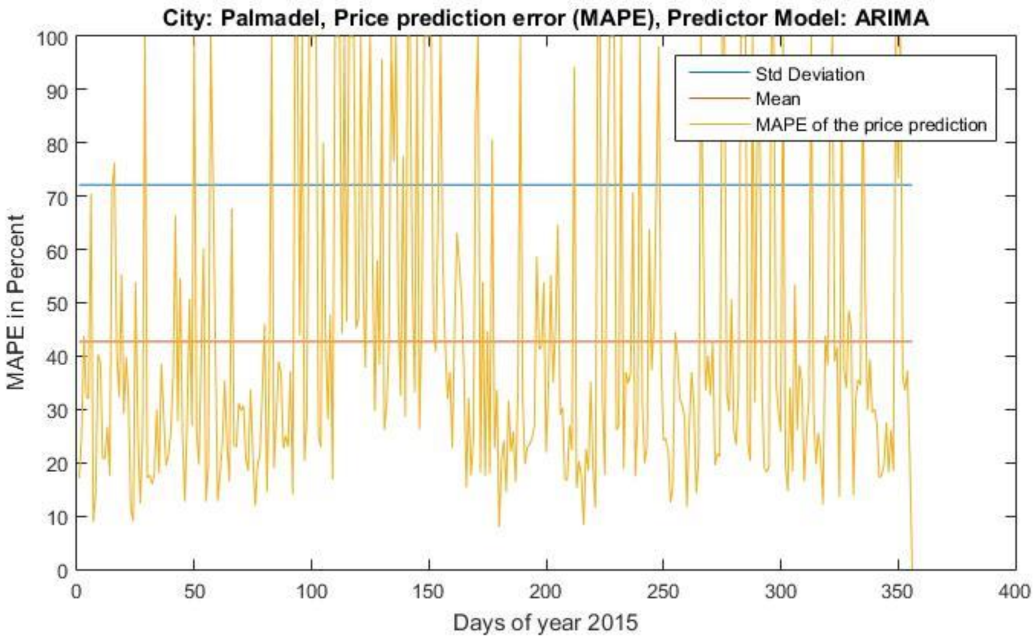


Figure 47 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Palmadel, Predictor model: ARIMA)*

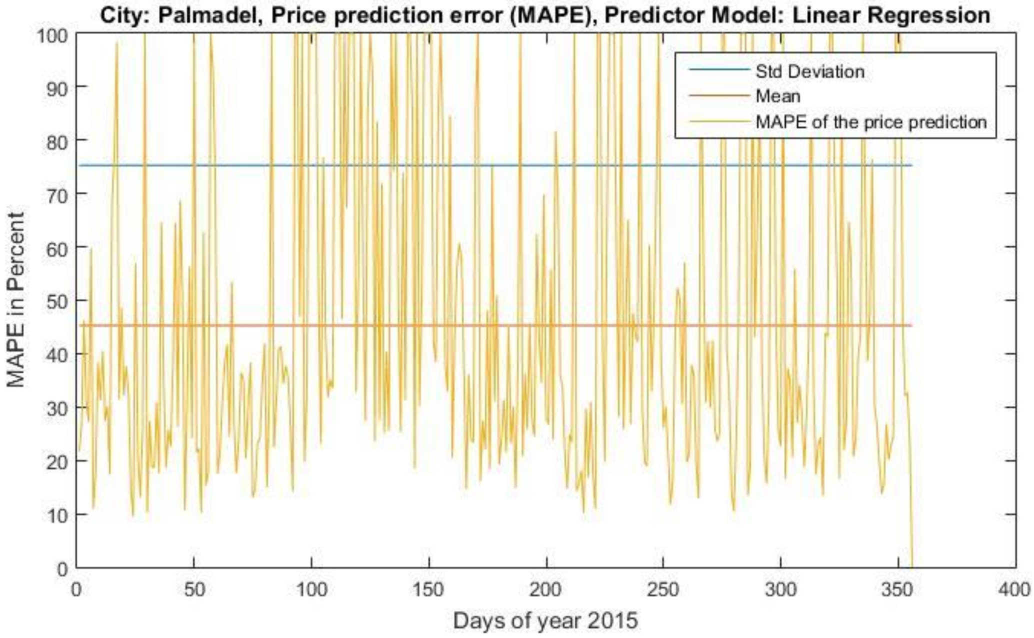


Figure 48 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Palmadel, Predictor model: Linear regression)*

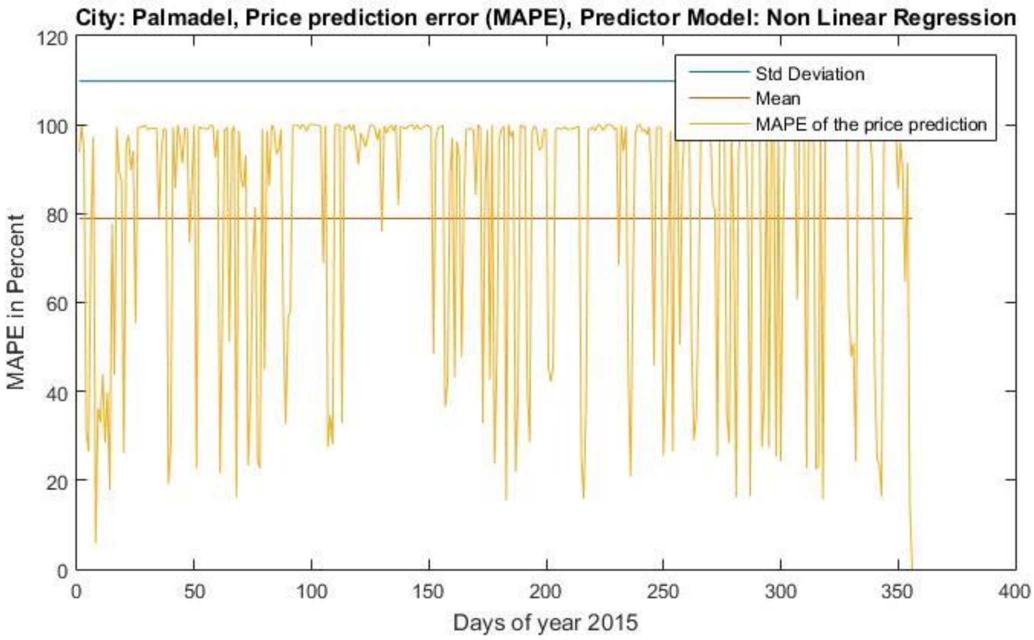


Figure 49 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Palmadel, Predictor model: Nonlinear regression)*

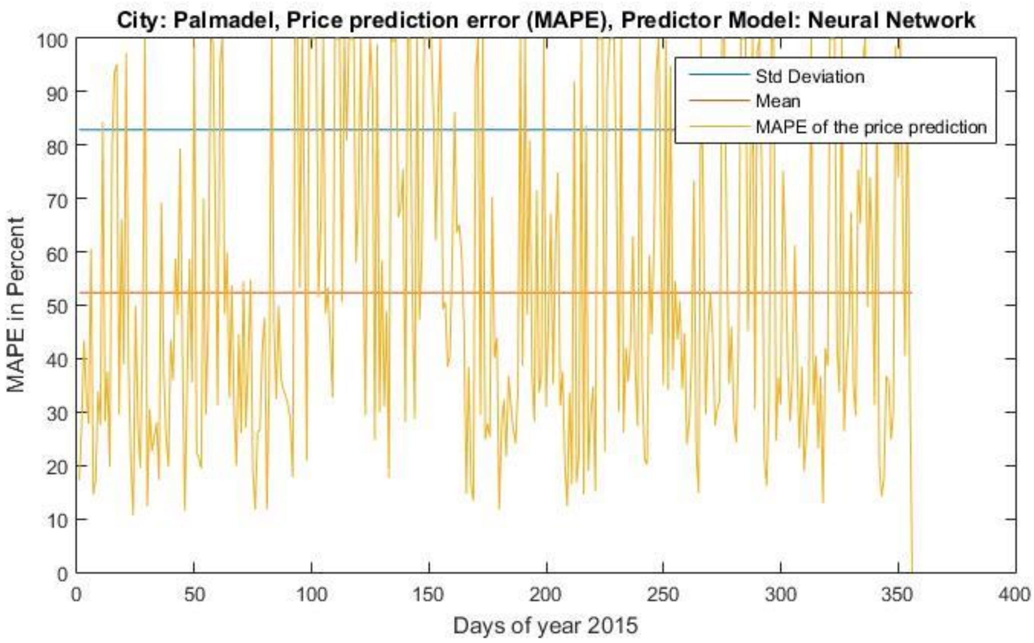


Figure 50 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Palmadel, Predictor model: Neural Network)*

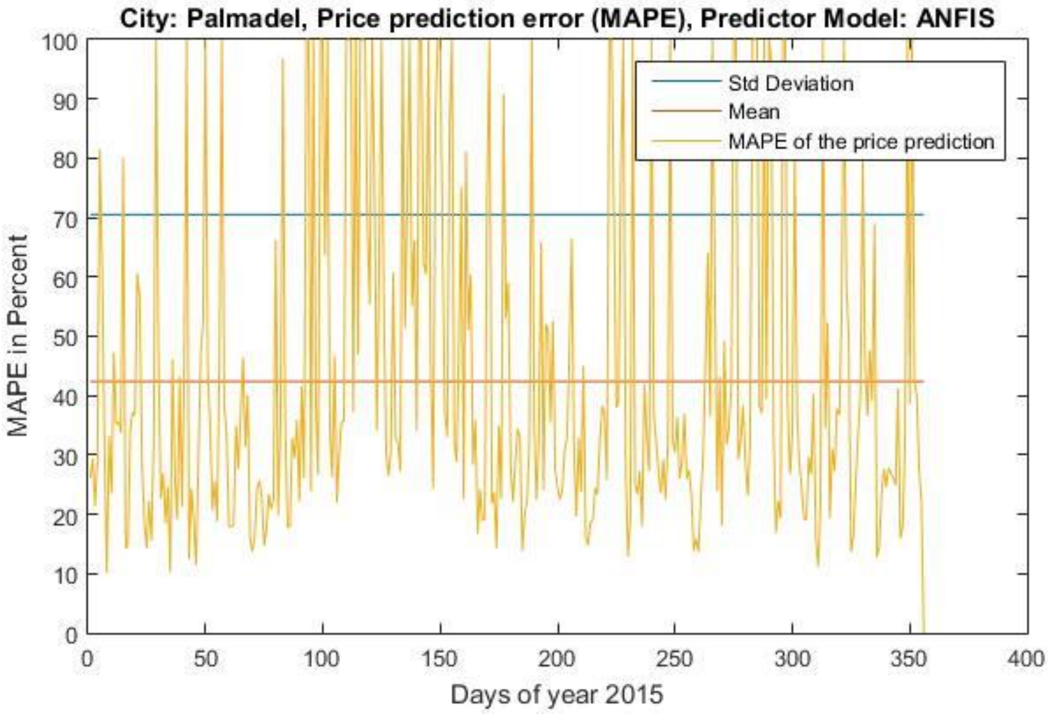


Figure 51 *Error in Electricity price prediction for the considered case (Device: Water pump, City: Palmadel, Predictor model: ANFIS)*

Statement Of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text. This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources. This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published

Date : December 20th, 2017

Signature : Sunil Hebbur Srikantamurthy