Battery optimization in microgrids using Markov decision process integrated with load and solar forecasting

Prateek Jain

BATTERY OPTIMIZATION IN MICROGRIDS USING MARKOV DECISION PROCESS INTEGRATED WITH LOAD AND SOLAR FORECASTING

by

PRATEEK JAIN

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

2018

Approved by

Jonathan W. Kimball, Advisor
Mehdi Ferdowsi
Robert Landers

ABSTRACT

Rising climatic concerns call for unconventional/renewable energy sources which reduce the carbon footprint. Microgrids that integrate a variety of renewable energy resources play a key role in utilizing these energy resources in a more efficient and environmentally friendly manner. Battery systems effectively help to utilize these energy resources more efficiently. This research work presents a framework based on Markov Decision Process (MDP) integrated with load and solar forecasting to derive an optimal charging/discharging action of Battery with rolling horizon implementation. The load forecasting regression models are discussed and developed. Also, various solar forecasting models like clear sky, multi-regression and Non-Linear Autoregressive Neural Network model with Exogenous time-series are discussed and compared. The control algorithm is developed to reduce the monthly billing cost by reducing the peak load demand while also maintaining the state of charge of the battery. The presented work simulates the control algorithm for one month based on historic load and solar data. The results indicate substantial cost savings are possible with the proposed algorithm.

ACKNOWLEDGMENTS

First and foremost, I would like to express my immense gratitude to my advisor Dr. Jonathan Kimball for his valuable insights, enthusiasm and engagement in my research work. His guidance has helped me completing this thesis.

I would also like to thank my committee members Dr. Robert Landers and Dr. Mehdi Ferdowsi for their constant support and technical feedback in my research work. My sincere thanks to Dr. Jonghyun Park for taking time and giving valuable points for improving my research quality. Thanks to Jacob Mueller for his technical support for modeling the mathematical framework of this thesis. This work was supported in part by the National Science Foundation, award 1610396.

I would thank my parents and family for their constant moral and emotional throughout my life and making it possible to follow my dreams. This thesis wouldn‘t have been possible without them. Lastly, my friends who were always there by my side and extending emotional support during this journey.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

The rising popularity of microgrids has encouraged integration of more renewable sources like Solar, wind and bio-diesel energies [6]. Integration of renewable energy in a microgrid helps to reduce the monthly bill as well as the maximum peak load demand. The load peak and renewable sources peaks must be matched to utilize the full potential of the renewable resources. The microgrid has no control over the renewable sources output and it is not a viable solution to expect the customer to plan their load demand according to the renewable source's output. Battery systems play a key role in managing such system by synchronizing load demand and renewable output [7]. Solar energy is widely popular in the consumer & industrial markets as a source of renewable energy. The system taken into consideration has a solar energy system as its renewable source. Solar energy and consumer load are stochastic in nature. Therefore, there is a need to design a smart grid controller which would predict load demand, solar energy output and eventually schedule the battery charging/discharging rate.

1.1. MOTIVATION

Energy storage devices are key components in utilizing the capabilities of microgrids while eliminating the need of fossil fuels/power system in some cases like island mode. However, energy storage has a variety of functionalities beyond providing energy while islanded, such as smoothing out an intermittent alternative generation source, load peak shaving, frequency regulation, demand response, etc. There are numerous applications of energy storage. Therefore, the main benefits of energy storage that this research work addresses are as follows:

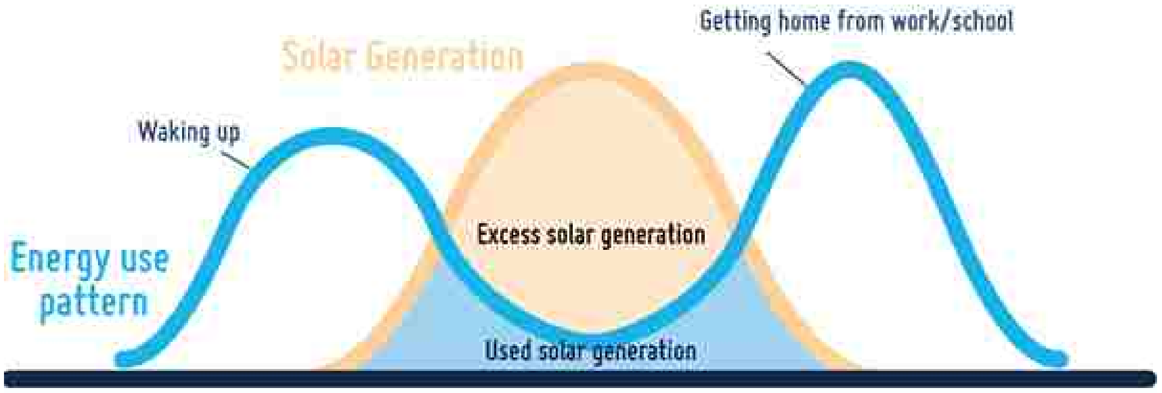


Figure 1.1. Load and solar profiles during the day. [1]

1. Synchronizing load and solar power profiles
2. Curbing the Duck curve
3. Economic factors

1.1.1. Synchronizing Load and Solar Power Profiles. Typically, the consumer load has a common trend throughout the day. The load increases in the morning as everyone in the house wakes up and get ready for work. In the afternoon load demand decreases and again rises up to a second peak in the evening depending on the consumer's lifestyle. On the contrary, the solar power output gradually increases in the morning, attains a maximum value at solar noon and reduces to zero in the evening (Figure 1.1). Without energy storage, surplus energy generated in the microgrid is fed back to the grid for which the utilities gives incentives to the customer. However, the costs of buying & selling electricity are different. The buying tariff is seven to ten times the selling tariff. From the economics point of view, the customer is buying the same electricity which was produced in the afternoon.

Due to the non-synchronized peaks of solar and load profiles, the renewable energy's capabilities are not fully utilized. Therefore, energy systems can help overcoming such problems by storing the surplus energy and utilizing it when needed (Figure 1.2) using sophisticated control algorithms which will be discussed in the later chapters.

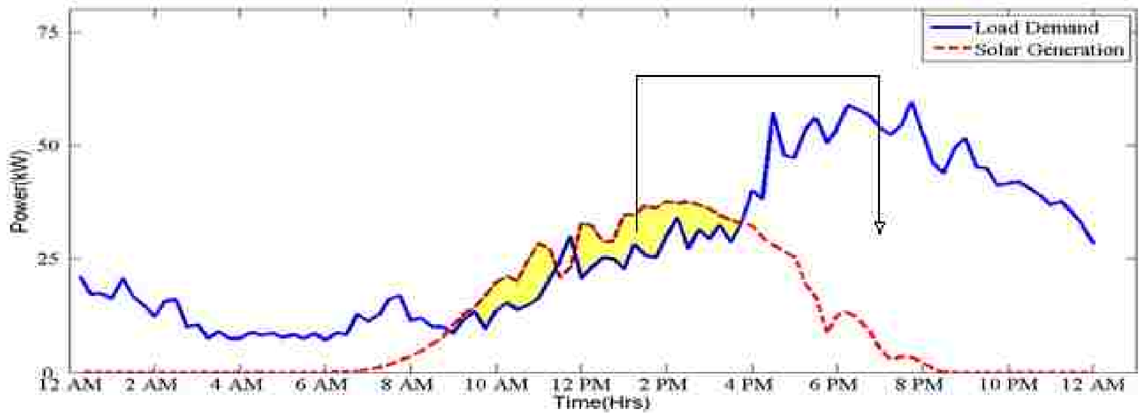


Figure 1.2. Store and utilize the surplus energy during peak load

1.1.2. Curbing the Duck Curve. In 2013, California ISO (Independent System Operator) published a series of graphs projecting the future timing imbalance between peak load demand & renewable generation (Figure 1.3). The large scale installation of solar energy in the state of California has led to an imbalance of load demand during the solar peak and evening. The duck curve refers to the transition from solar peak to the sunset where the electricity generators needs to quickly ramp up the production during this time period. This leads to instability in the power system & higher operational cost. There are two solutions to this problem. One is at the macro level where different states/utilities can share load so that the load curve is consistent in power system. This solution is more complex & time consuming due to various economic and political hurdles. Another solution is at the micro level where the customer can locally install energy storage. The energy storage can store the surplus energy generated during solar peak hours and use it during load peak hours. This not only minimizes the imbalance between load demands during different times but also helps in peak shaving. This approach helps both the customers to lower down there monthly billing cost and also reduce operational cost for utilities which indirectly saves customer's money.

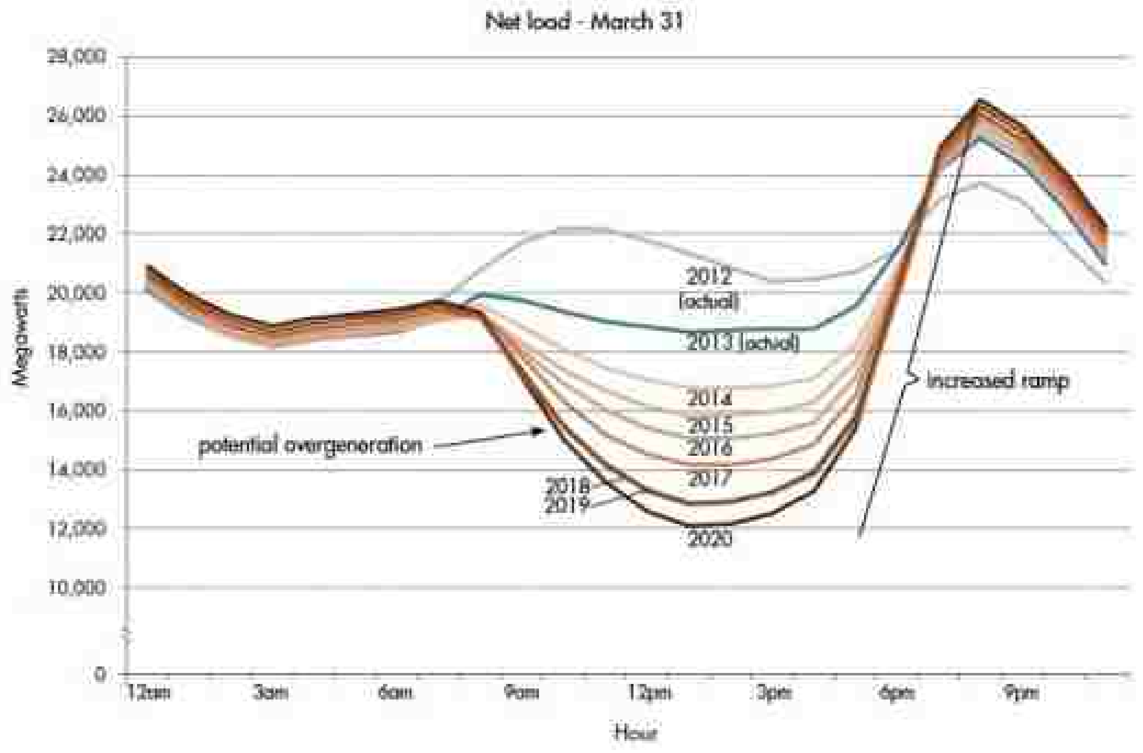
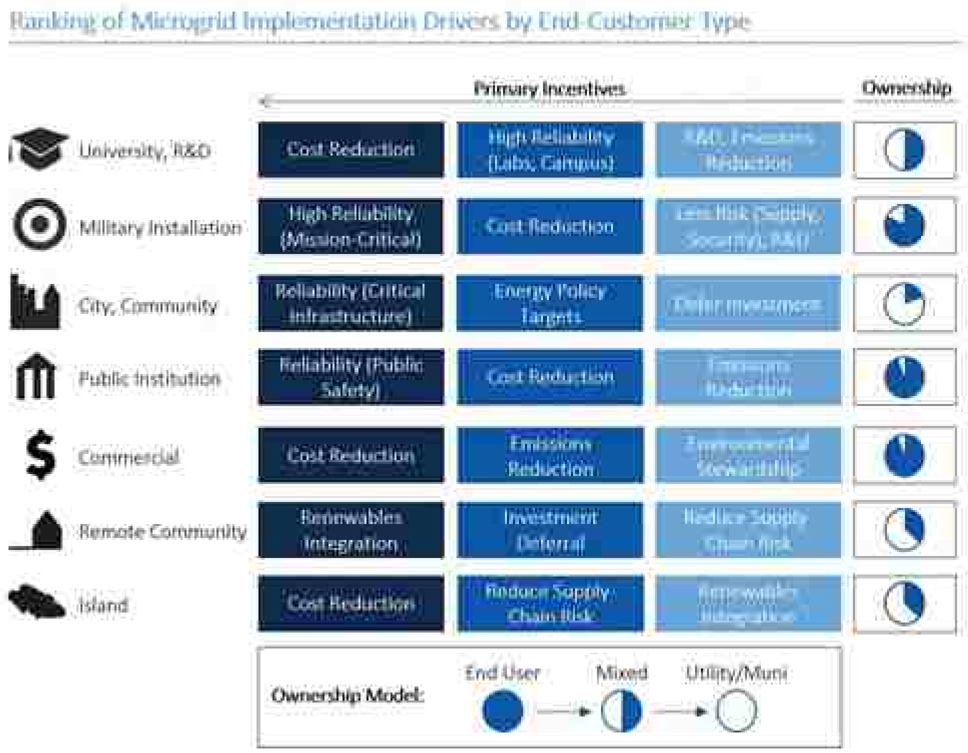


Figure 1.3. Load demand projections for different years[2]

1.1.3. Economic Factors. We live in an economically driven society where every product is bought with a motivation to have economic benefit from it. When a customer installs a microgrid, they expect some kind of service or a return of investment. Figure 1.4 depicts the ownership of microgrid by different sectors and the incentives they expect to receive. The top two motivations/expectations for installing a microgrid are system reliability and Cost reduction. As discussed in previous sections, solely using renewable energy cannot ensure cost reduction. Installing Energy storage not only ensures reliability but cost reduction too so that the consumer get return of investment in minimum possible time frame.



Ranking of Microgrid Implementation Drivers by End-Customer Type

	Primary Incentives			Ownership
University, R&D	Cost Reduction	High Reliability (Labs, Campus)	R&D, Emissions Reduction	◑
Military Installation	High Reliability (Mission-Critical)	Cost Reduction	Less Risk (Supply, Security), R&D	●
City, Community	Reliability (Critical Infrastructure)	Energy Policy Targets	Defer investment	◔
Public Institution	Reliability (Public Safety)	Cost Reduction	Emissions Reduction	●
Commercial	Cost Reduction	Emissions Reduction	Environmental Stewardship	●
Remote Community	Renewables Integration	Investment Deferral	Reduce Supply Chain Risk	◑
Island	Cost Reduction	Reduce Supply Chain Risk	Renewables Integration	◑

Ownership Model: End User ● → Mixed ◑ → Utility/Muni ○

Figure 1.4. Microgrid ownership in different sectors and their motivation[3]

1.2. LITERATURE REVIEW

A Markov Decision Process (MDP) is a mathematical tool to take decision in systems where the outcome is partially stochastic & partially deterministic. MDP is widely popular in different areas of specialization from banking to biomedical engineering. The emergence of smart grids has called for implementation of MDP algorithms in microgrids too. Different problem statements might demand for different MDP models for the microgrids. For instance, in [6, 7], the authors discuss the problem of reducing the difference between the demand & the supply. At every time step, system takes a decision of meeting the power requirement either by renewable energy or the main grid. The MDP model is solved using Multi-Agent Q-Learning technique which doesn't require any prior information of the system. [7] further discusses the difference between Q-learning and Coordinated Q-

Learning where the objective function remains reducing the power consumption from the grid. Both of these research works do not include any forecasting models for load and solar. Rather, they depend on the immediate changes on the environment.

Research work presented in this thesis is more closely related to works in [8, 9]. In [9], the MDP optimally schedules the energy storage in power distribution including renewable resources. The output of the algorithm is an optimal policy for scheduling the energy system while minimizing the objective function, which includes total cost and the energy losses in the power system. Besides scheduling the battery using dynamic programming, this paper assesses & compares the battery system size optimal for a network operation.

Paper [10] formulates an optimal management & sizing of energy storage with dynamic pricing keeping dynamic pricing stochastic in nature. The algorithm solves the problem with minimum cost incurred to the customer keeping conversion losses, transmission losses and investment costs into consideration. This paper also analyzes the size of the energy system vs its gains.

The MDP algorithm in microgrids runs for indefinite period of time. Therefore, a finite horizon problem i.e. an algorithm which optimizes only a finite time frame cannot work for microgrid applications. Therefore, infinite horizon or rolling horizon MDP is a feasible solutions for the given scenario. In [11], the Rolling horizon MDP algorithm in microgrids is presented with combined heating and power (CHP) generation to satisfy the electric & heat loads in the system.The research solves the problem with variable number of CHPs using greedy algorithm. The research work lacks the prediction of wind turbine output & load demand which is an integral part in implementing a practical energy management system.

Research work [12] presents an energy management system based on rolling horizon strategy with solar and wind as renewable resources. The energy system proposed also considers the SOH (State of Health) of the batteries which accounts for the investment and the life of ESS. The paper also implements the load forecasting using Neural Network and solar forecasting using clear sky model integrated with MDP. The load and solar energy are stochastic in nature therefore the forecasting models cannot predict them accurately every time. Therefore simulating only one day of battery scheduling cannot accurately capture the performance of Energy Management System. Research work presented in this thesis attempts to develop the energy management system with MDP, Load forecasting using Auto-regressive Moving Average with Exogenous Input (ARIMAX) and solar forecasting using Non-Linear Auto-Regressive Exogenous Input (NARX) simulating for 30 days.

Load forecasting is a central area of interest for electric utilities and microgrids [13, 14, 15]. Time series models, which include a linear combination of past values and Gaussian errors, have been widely popular in the research community for short-term load forecasting. [16] implements a short-term load forecasting using ARIMA model & transfer function model by considering the weather forecast. It formulates models for different sectors like residential, commercial and industrial loads. The transfer function relationship between load and weather are different for different sectors therefore, different models are required. The paper concludes that the transfer function ARIMA models perform better than the ARIMA models.

Paper [17], analyzes nine different methods for short-term load forecasting like regression, adaptive load forecasting, stochastic time-series, fuzzy logic, etc. The research concludes that the load forecasting models require more sophisticated forecasting models with an inclination towards stochastic & dynamic forecasting techniques. There is also a trend towards developing hybrid models which combine two or more techniques to extract the best features of these techniques.

Similar forecasting techniques are applied to Solar forecasting too. Either the solar radiation is predicted or the PV output is predicted directly using various forecasting models. Time series models have been well used in solar forecasting [18, 19, 20]. The solar energy is partially predictive due the Earth-Sun geometry constraints and partially stochastic due its dependence on surrounding environment like clouds & nearby materials. Therefore, traditional models tend to fail. Techniques like Neural Networks and Support Vector Machines are used for solar forecasting due to their abilities to capture the non-linearity and non-stationarity of the time series.

Paper[21], presents a comparison between multi linear analysis model, Persistence and Neural network model. Multi linear model is a regression of various parameters affecting the solar radiation like temperature, humidity, time of day, etc. Persistence model as the name suggests is obtained by keeping the actual value constant for the current hour and using it for the forecast. Though this technique works for very short-term forecasting but if the prediction window is large, this technique fails. The Artificial Neural Network models have better accuracy than the multi liner and Persistence models.

Paper [22] presents a Neural network model for PV output forecasting taking exogenous time series like temperature, humidity, Pressure, cloud cover and past observed PV output into consideration. These are also called Non-Linear Autoregressive Neural Network (NARX) models. The paper further studies the Mean Average Error (MAE) in different cloud conditions. The research presented in this thesis has adopted this technique for implementation of Energy Management System.

1.3. ORGANIZATION OF THE THESIS

In addition to the section 1, which represents the motivation of using battery systems in microgrids, need of energy systems and literature review, section 2 presents the formulation of the microgrid system and there mathematical constraints. Markov Decision Process is discussed in reference to the microgrids and mathematical framework behind the implementation of the same. The implementation of MDP in microgrid is discussed with zero forecasting errors to obtain a proof of concept and the theoretical limitations of the system.

Section 3 discusses load forecasting techniques like ARIMA (Auto Regressive Moving Average), SARIMA (Seasonal Auto Regressive Moving Average) & ARIMAX (Auto Regressive Moving Average with Exogenous time series) models. The implementation of the load forecasting models is presented and results of a one month load forecasting simulation is shown. Chapter 4 presents the solar forecasting techniques. Clear Sky model, Multi linear model and Auto Regressive Exogenous Neural Network Models are discussed and compared.

Section 4 shows the implementation of the Energy Management system by integrating MDP, Load & Solar forecasting. The Energy Management system presented in this thesis is compared with a Heuristic approach which does not require any intelligent control.

2. SYSTEM MODELING & MDP

2.1. SYSTEM MODELING

The system under consideration contains a local load, PV source, battery system and grid connection. System is shown in Figure 2.1. Arrows describe the flow of energy. Solar (P_{PV})and load power (P_L) flows are unidirectional and only take positive values. Whereas the power flows of Grid (P_G) and battery system (P_E) are bidirectional and can take both positive and negative values. Grid power (P_G) is positive when energy is drawn from it and negative when surplus energy is fed into the grid. Similarly, Battery power (P_E) is positive when it is discharged and negative when charged. Solar output power (P_{PV}) and Load demand (P_L) are always positive. The system operates with the following constraints:

$$P_L - P_G - P_E - P_{PV} = 0 \tag{2.1}$$

$$SoC_{min} \leq SoC(k) \leq SoC_{max} \tag{2.2}$$

$$P_E^{min} \leq P_E \leq P_E^{max} \tag{2.3}$$

The Battery power (P_E) is bounded by maximum and minimum powers P_E^{max} & P_E^{min} respectively. The State of Charge ($SoC(k)$) at any time step k cannot exceed minimum (SoC_{min}) and maximum (SoC_{max}) values based on battery specifications.

2.2. MARKOV DECISION PROCESS

Markov Decision Processes (MDPs) are a class of stochastic sequential decision processes in which the cost and transition functions depend only on the current state of the system and the current action [8]. As load demand and PV generation are uncertain in

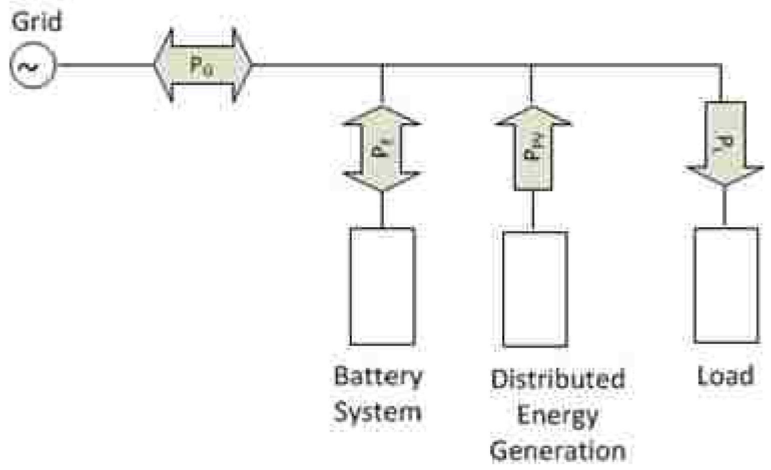


Figure 2.1. System architecture

nature, MDP is a good option to schedule the battery charging/discharging rate. A day is discretized into 15 minutes interval i.e. 96 epochs thereby reducing the problem statement to making 96 decisions of charging/discharging battery rate in 24 hours. Rolling Horizon MDP is implemented in the system to ensure that the control algorithm always keeps 24 hours into consideration while making a battery action.

The state, s_k, at epoch k has all the information necessary to define cost and transition probabilities

$$s_k = \{E, \hat{u}_{PV}, \hat{\sigma}_{PV}, \hat{u}_L, \hat{\sigma}_L, \theta_s, \theta_b, k\} \tag{2.4}$$

where,

E = Discretized energy of battery in kWh

$\hat{u}_{PV}$ = Point forecast of PV Generation in kW

$\hat{\sigma}_{PV}$ = Standard error of PV Generation in kW

$\hat{u}_L$ = Point forecast of Load demand in kW

$\hat{\sigma}_L$ = Standard error of Load demand in kW

θ_s = Electricity selling cost in \$/kW

θ_b = Electricity buying cost in \$/kW

k is the epoch

2.2.1. Battery Energy. During transitions between epochs,$\widehat{u}_{PV}$, $\widehat{\sigma}_{PV}$, $\hat{u}_L$, $\hat{\sigma}_L$, θ_s, and θ_b are independent of control actions and are deterministic. Therefore, state transitions are determined by the change in battery energy, $e(t)$. To facilitate calculation, energy is discretized into M bins of size.

$$E_{bin} = \frac{E_{\max} - E_{\min}}{M} \tag{2.5}$$

So, a battery can have energy (state) E_i as:

$$L_i = E_{min} + (i-1)\,E_{bin} \tag{2.6}$$

$$U_i = E_{\min} + iE_{bin} \tag{2.7}$$

$$L_i \leq e(t) \leq U_i \tag{2.8}$$

U_i is the upper limit of Energy at $E = i$ and L_i is the lower limit of Battery Energy at $E = i$. Therefore battery can have M discrete states during the scheduling and energy $e(t)$ of battery would be greater than L_i and smaller than U_i. In the simulations, M is taken as 200, E_{max} as 200 kWh and E_{min} as zero but during practical implementation, boundaries can be set on E_{max} & E_{min} to ensure that the battery is charged/discharged till certain levels.

2.2.2. Battery Actions. In MDP formulation for battery scheduling, actions(n) is considered to be Charging/Discharging rate in kWh. This is defined by the specification of the battery system. From each state E_i, some finite discrete actions are possible. These actions i.e. charge/discharge rates (n) are chosen such that there is an increment/decrement in energy E with a probability of 1 (Equation 2.9).

$$P_{ij} = \begin{cases} 1 & if\,(i-j) = n \\ 0 & if\,(i-j) \neq n \end{cases} \tag{2.9}$$

Where, i is the state at epoch k and j is the state at epoch $k+1$. The action in kWh can be computed by referring to the specification of the Battery system (Equation 2.10).

$$P_E = \begin{cases} \frac{nE_{bin}\eta_d}{\Delta t} & n > 0 \\ \frac{nE_{bin}}{\Delta t \eta_c} & n < 0 \end{cases} \tag{2.10}$$

where,

P_E is battery charging/discharging power

η_d & η_c are discharging and charging efficiency respectively

Δt is the time interval between two actions (15 minutes in our case)

2.2.3. Cost. Total cost accumulated at each state denoted by $C_{ij,k}$ of a certain epoch is a function of destination state of the next epoch and the action taken. It is to be noted that the cost mentioned here is not the actual cost of the day. Rather, it is divided into two parts:

1. Grid Cost
2. Energy Transfer Cost

2.2.3.1. Grid cost. This is the expected cost by transitioning from state E_i to E_j at epoch k.

Now, P_G is estimated by the following constraint:

$$P_G = P_L - P_{PV} - P_E \tag{2.11}$$

P_E is held constant. Distribution of P_G is Gaussian with following parameters:

$$\hat{\mu}_G = \hat{\mu}_L - \hat{\mu}_{PV} - P_E \tag{2.12}$$

$$\hat{\sigma}_G = \sqrt{\hat{\sigma}_L^2 + \hat{\sigma}_{PV}^2} \tag{2.13}$$

There are majorly two types of rate structures imposed by utilities on customers:
1) Time of Use (TOU) rate: In this structure, the prices are increased during peak load periods. For example, utilities might have peak period from 12:00 PM to 7:00 PM when the demand is the highest. Purchase price for the off peak hours would be lower than the on-peak hours thereby encouraging customers to consume less power during those periods. Cost associated with this scheme for a particular epoch with P_E held constant is as follows:

$$C_G = \begin{cases} P_G \theta_s \Delta t & P_G < 0 \\ P_G \theta_b \Delta t & P_G > 0 \end{cases} \tag{2.14}$$

Where, Δt is the size of one epoch which is 15 minutes in this study. The expected cost of C_G is given in Equation 2.15

$$E\left[C_G\right] = \int_{-\infty}^{0} \frac{\theta_s \Delta t x}{\sqrt{2\pi}\hat{\sigma}_G} \exp\left(\frac{-(x-\hat{\mu}_G)^2}{2\hat{\sigma}_G^2}\right) dx + \int_{0}^{\infty} \frac{\theta_b \Delta t x}{\sqrt{2\pi}\hat{\sigma}_G} \exp\left(\frac{-(x-\hat{\mu}_G)^2}{2\hat{\sigma}_G^2}\right) dx \tag{2.15}$$

$$E\left[C_G\right] = \frac{\hat{\mu}_G \Delta t}{\sqrt{2}\hat{\sigma}_G}\left[(\theta_b - \theta_s)\left(\frac{\hat{\sigma}_G}{\hat{\mu}_G}\sqrt{\frac{2}{\pi}} \exp\left(\frac{-\hat{\mu}_G^2}{2\hat{\sigma}_G^2}\right) + \operatorname{erf}\left(\frac{\hat{\mu}_G}{\sqrt{2}\hat{\sigma}_G}\right)\right) + \theta_b + \theta_s\right] \tag{2.16}$$

2) Demand charge: This charge is generally levied on Industrial/commercial facilities or customers with high power consumption. Some utilities in states like Alabama, Arizona, Wyoming, California and many more [23] have implemented demand rate structure for residential customers too. Bill is generated on the basis of two components, first is fixed rate where the purchase rate remains constants throughout the billing cycle. Second

is the demand rate which accounts for a large amount in the bill. Demand charge is based on the highest power consumed in a 15 min duration for one month billing cycle. The Cost associated with this rate structure is as follows:

$$C_{G=}\begin{cases} P_G\theta_s\Delta t & P_G < 0 \\ P_G\theta_b\Delta t & 0 < P_G < \gamma \\ (P_G-\gamma)\,\theta_d\Delta t + \gamma\theta_b\Delta t & P_G > \gamma \end{cases} \tag{2.17}$$

where,

θ_s = Selling price in \$/kWh

θ_b = Buying price in \$/kWh

θ_d = Demand charge in \$/kWh

γ = Threshold value for demand in kWh

Expected cost of C_G is given in Equation 2.18 and 2.19. In this research, Demand charge rate structure is going to be followed.

$$\begin{aligned} E\left[C_G\right] = {} & \frac{\theta_s\Delta t}{\sqrt{2\pi}\hat{\sigma}_G}\int_{-\infty}^{0} x\exp\left(-\frac{(x-\hat{\mu}_G)^2}{2\hat{\sigma}_G^2}\right)dx + \frac{\theta_b\Delta t}{\sqrt{2\pi}\hat{\sigma}_G}\int_{0}^{\gamma} x\exp\left(-\frac{(x-\hat{\mu}_G)^2}{2\hat{\sigma}_G^2}\right)dx \\ & + \frac{\theta_d\Delta t}{\sqrt{2\pi}\hat{\sigma}_G}\int_{\gamma}^{\infty} x\exp\left(-\frac{(x-\hat{\mu}_G)^2}{2\hat{\sigma}_G^2}\right)dx \end{aligned} \tag{2.18}$$

$$\begin{aligned} E\left[C_G\right] = {} & \frac{(\theta_d-\theta_s)}{2}\left(\hat{\sigma}_G\sqrt{\frac{2}{\pi}}\exp\left(\frac{-(\gamma-\hat{\mu}_G)^2}{2\hat{\sigma}_G^2}\right) + (\gamma-\hat{\mu}_G)\,\mathrm{erf}\left(\frac{\gamma-\hat{\mu}_G}{\sqrt{2}\hat{\sigma}_G}\right) + \gamma\right) \\ & + \frac{(\theta_b-\theta_s)}{2}\left(\hat{\sigma}_G\sqrt{\frac{2}{\pi}}\exp\left(\frac{-(\hat{\mu}_G)^2}{2\hat{\sigma}_G^2}\right) + \hat{\mu}_G\mathrm{erf}\left(\frac{\hat{\mu}_G}{\sqrt{2}\hat{\sigma}_G}\right)\right) + \frac{\hat{\mu}_G\left(\theta_d+\theta_s\right)}{2} \end{aligned} \tag{2.19}$$

2.2.3.2. Energy transfer cost. This is the cost of Energy transfer from/to battery to emphasize on the notion that the battery is degraded while charging/discharging it and enabling the MDP to determine whether it is financially advantageous to supply the load with grid energy or battery energy. Battery cost C_B is defined as :

$$C_B = E_{bin}\,(i - j)\,\theta_e \tag{2.20}$$

where,

θ_e = Energy transfer cost in \$/kWh

i = Initial energy state at epoch k

j = Final energy state at epoch (k+1)

From Equation 2.20, the cost is positive when $j < i$ and negative when $j > i$. The primary objective of the system is to attain least cost. Therefore, this cost term tries to attain higher SoC (State of Charge) of the battery which heavily dependent on θ_e.[24, 25], estimate the value of θ_e by considering the battery installation cost and lifetime. Though it is out of the scope of this study but SoC and SoH (State of Health) estimation can be implemented in this topology to calculate θ_e for considering the degradation and lifetime of battery system. The total cost for transitioning from state "i" to "j" at epoch "k" is described in Equation 2.21.

$$C_{ij,k} = C_G + C_B \tag{2.21}$$

2.2.4. Optimal Policy. Dynamic programming is used to calculate the optimal policy of the MDP. The tree diagram shown in Figure 2.2 summarizes the implementation of MDP in microgrid. The optimal policy of each state at epoch k is defined by the Bellman equation.

$$U^*_{i,k} = \min_{nA} \sum_{j=1}^{M} P_{ij}(C_{ij,n} + U^*_{j,k+1}) \tag{2.22}$$

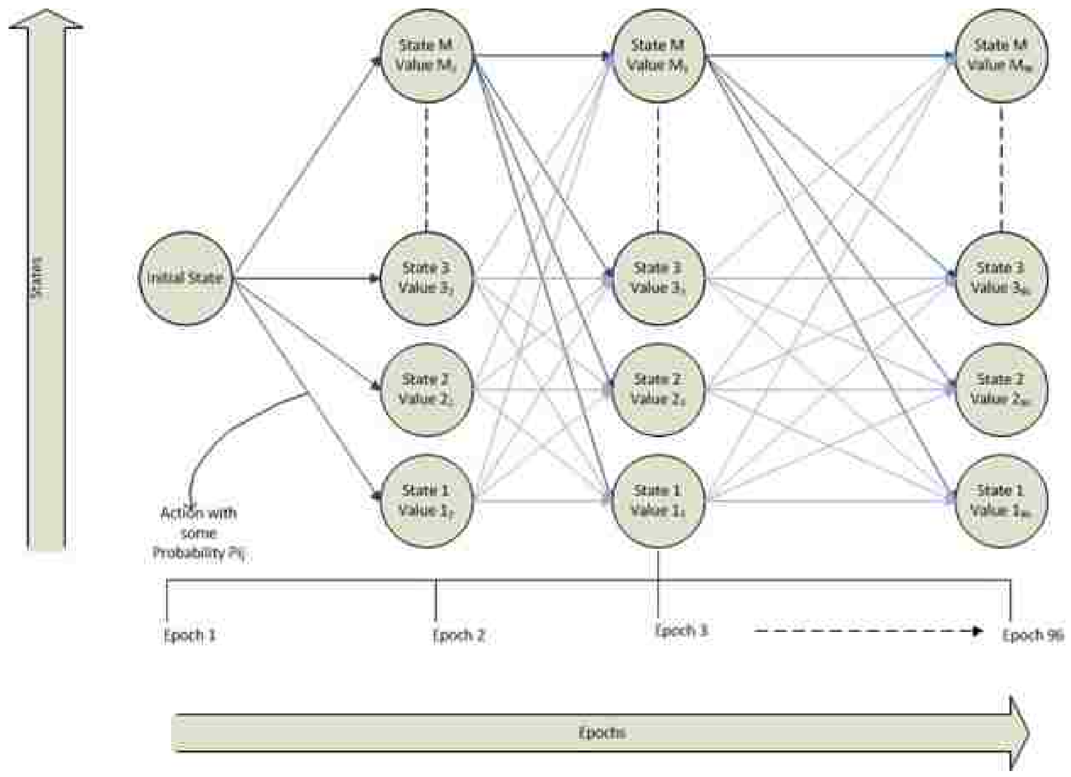


Figure 2.2. Implementation of MDP using Dynamic Programming

$$U^*_{i,K} = -\theta_e \left(E_{\min} + (i - 0.5)\, E_{bin}\right) \tag{2.23}$$

$$a^*_{i,k} = \underset{n \in A}{\operatorname{argmin}} \sum_{j=1}^{M} P_{ij}(C_{ij,n} + U^*_{j,k+1}) \tag{2.24}$$

Where, $U_{i,K}$ is the optimal utility at epoch k and state i, K is the last epoch (e.g. 96), and $a^*_{i,k}$ is the optimal action at epoch k and state i. Actions determined by Equation 2.24 determine the optimal policy for the battery controller. Since the terminal cost is based on the energy, the system prioritizes high battery energy at the end of the horizon while reducing the grid cost.

Rolling Horizon MDP ensures that there is always enough energy in the battery system to supply day ahead load demand taking into consideration both solar output and load demand.

2.3. IMPLEMENTATION

To validate the performance of MDP model and obtain a theoretical limit of the system, forecasting errors are ignored. In other words, MDP model is integrated with load and solar forecasting models which have zero forecasting errors. The load & solar data are taken from a research facility Pecan Street dataport situated in Austin. A community of 20 houses is considered with 10 houses having PV panels installed. demand charge rate structure is imposed using parameters from RMU (Rolla Municipal Utilities) given in Table 2.1. The battery system is assumed to have a capacity of 200 kWh with an initial SoC of 50%. The battery system has maximum charging power of 68.4 kW and maximum charging capacity of 75.7 kW.

Table 2.1. Demand rate structure used in simulations (Adopted from Rolla Municipal Utilities)

Parameter	Rate
Buying Rate (θ_b) (\$/kWh)	0.07009
Selling Rate (θ_s) (\$/kWh)	0.01326
Demand Rate (θ_d) (\$/kW)	14.5

It can be observed from Figure 2.3 & 2.4 that the solar output is stored in battery bank during surplus generation and the same energy is used during peak load. This helps to maintain constant power drawn from the grid and reduce the maximum peak power. This in turn reduces the demand charge in the system. Table 2.2 shows the cost comparison of system with & without MDP & battery system.

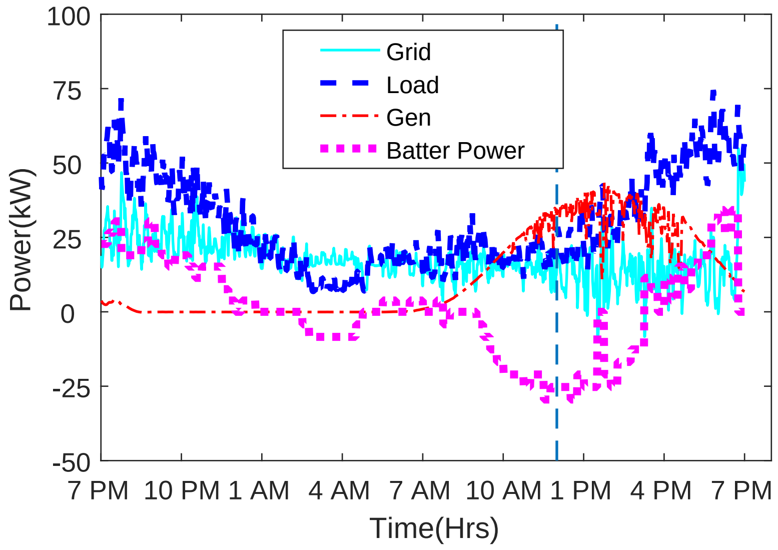


Figure 2.3. MDP simulation for one day

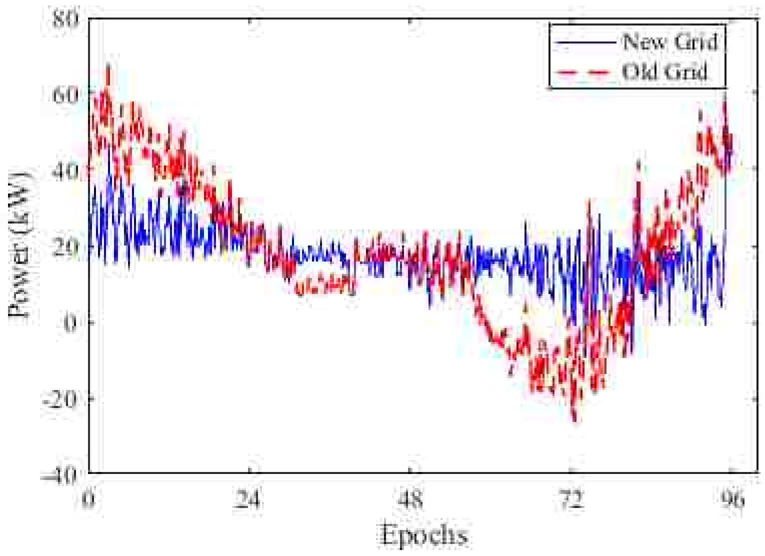


Figure 2.4. Comparison between the grid powers of system with & without MDP. Old grid is the system without MDP & battery system

Table 2.2. MDP one month simulation results

	Without Battery Storage	MDP without forecasting errors
Cost ($)	1944.9	1015.49
Savings ($)	NA	929.4
Maximum Load (kW)	74.4	34

It can be observed from Table 2.2 that the system with no battery pack has a maximum load demand of 74.4 kW. Whereas, a system with battery pack and MDP algorithm reduces the maximum load demand to 34.6 kW thereby saving $929.4 in one month. Though, this model only signifies the capabilities of MDP algorithm & a theoretical limit to the cost savings. The MDP model is incomplete without load and solar forecasting models. Therefore, forecasting models are discussed in proceeding chapters in order to integrate them with MDP.

3. LOAD FORECASTING

Load forecasting is a tool used by utilities and power companies for planning & operation of power systems. Increased acceptability of microgrids has encouraged researchers to forecast loads at a micro level for energy control purposes and increase the penetration of renewable energies in the microgrid system. Load forecasting captures the customer behavior and predicts load with a lead time from several minutes to even days depending on the application. Load forecasting can be categorized based on the forecast horizon:

1. Short term forecast: The prediction period ranges from several minutes to weeks. It is generally used for Network planning, supply/demand matching, load shedding strategy, etc.

2. Medium term forecast: The prediction period ranges from weeks to months. The main advantages of medium term forecast are network planning, power procurement & rate case development.

3. Long term forecast: The prediction window ranges from months to years. Long term forecasting is generally used by power companies for investment planning and projecting the need for infrastructure.

The system load in a microgrid is the sum of all individual loads of all houses forming a microgrid. In principle, if the characteristic consumption of individual house is known, load can be predicted easily. The total load in a microgrid results in distinct features which can be statistically predicted.The load pattern is influenced by a number of factors which are listed below:

1. Economic: Economic factors have different meaning for different sectors where load forecasting is performed. For microgrids, the rate structure imposed by the utilites is crucial. For example, in case of Time of Use (TOU) rate structure, consumer

will try to maintain low load demand during specific times. Whereas, in case of Demand charge structure customer will try to maintain low load demand peak (flatter consumption profile). Therefore, knowing the economic factors can play a key role in predicting the future load demand .

2. Time: The consumer load has a specific profile which can be extracted from the time of prediction. The load profile therefore repeats itself after every 24 hours. This is called seasonality. Even if the consumer's load consumption pattern changes, it is updated on the next day's forecasting results. Other time factors affecting the load profile are weekends, holidays like Christmas and special events like the Super Bowl.

3. Weather: Meteorological changes are major factors affecting load consumption due to the presence of weather sensitive components in the system, particularly air conditioning and space heating. The inputs for the load forecasting can be temperature, humidity, dew point & wind speed. However, not all weather parameters can have correlation with the load consumption. Analysis is needed to decide which weather parameters directly affect the load consumption.

4. Stochasticity: Random behavior of the customer or event can cause a change in load consumption which cannot be explained from the other factors discussed above. This property largely calls for the load forecasting models.

3.1. LOAD FORECASTING MODELS

Load forecasting can be performed with different models like multiple regression technique [13, 26], Time series model & Artificial Neural Networks. Time series models are the classical approach to load forecasting. It is a linear combination of past observed values of the load demand. Time series models are relatively simple compared to the Artificial intelligence (AI) methods and produces same accuracy as that of the AI models [27]. This research work focuses on developing a Time series model for load forecasting due to its simple mathematical framework and wide acceptability for practical applications.

3.1.1. Time Series Models. Time series is defined as the series of data generated sequentially in time [28]. The time series models assume that the future data is related to the past observed values of the time series.

Introduced by Box and Jenkins [29], ARIMA (Auto Regressive Integrated Moving Average) modeling has been a popular technique to predict the future load demand [30]. ARIMA model has three components: Autoregressive (AR), Moving Average (MA) and differencing.

The primary requirement of the ARIMA model is that the time series has to be stationary. In other words, a time series is stationary if the statistical properties (mean, variance, autocorrelation) do not change with time. Kwiatkowski−Phillipsi−Schmidti−Shin (KPSS), Dickey−Fuller test (ADF) and unit root test are the common methods to determine the stationarity of a time series. Mathematical operations are needed to be performed if the time series is not stationary. A non-stationary time series can be made stationary by differencing the data set with various methods like normal differencing, exponential smoothing, regressing on trends, etc. In most of the cases, one or two differencing is enough to make a series stationary. The order of differencing is denoted by “d”.

If $d = 1$, the time series is stationary.

$$y_k = Y_k \tag{3.1}$$

If $d = 1$, the series is differenced with itself once.

$$y_k = Y_k - Y_{k-1} \tag{3.2}$$

If $d = 2$, the time series is differenced twice.

$$y_k = (Y_k - Y_{k-1}) - (Y_{k-1} - Y_{k-2}) = Y_k - 2Y_{k-1} + Y_{k-2} \tag{3.3}$$

where,

y_k is the stationary series obtained after differencing

Y_k is the original time series

k is the discretized time step

It can be observed that as the differencing order increases, the equations also become complex. Therefore, to make the time series equations easy to perceive, a lag operator "B" is introduced (Equation 3.4). Operator "B" is similar to the z^{-1} operator in discrete domain.

$$BY_k = Y_{k-1} \qquad B^m Y_k = Y_{k-m} \tag{3.4}$$

Therefore, a one and two differencing equations (Equation 3.5 & 3.6) can be represented as follows:

$$y_k = (1 - B)Y_k = Y_k - BY_k \tag{3.5}$$

$$y_k = (1 - B)^2 Y_k = Y_k - 2BY_k + B^2 Y_k \tag{3.6}$$

Autoregressive (AR) component stresses on the fact that the present value of the load demand is related to the past observed values. It is weighted sum of past observed/forecast values of the time series. The order of autoregressive model (p) is determined using partial autocorrelation. The Moving average (MA) term is a weighted sum of the forecast errors. The order of MA (q) is determined from auto-correlation plots.

Though, the partial correlation and auto correlation plots gives a ballpark numbers for the orders of AR & MA orders respectively.Figure 3.1 & Figure 3.2 shows the pacf & acf plots of a stationary time series.

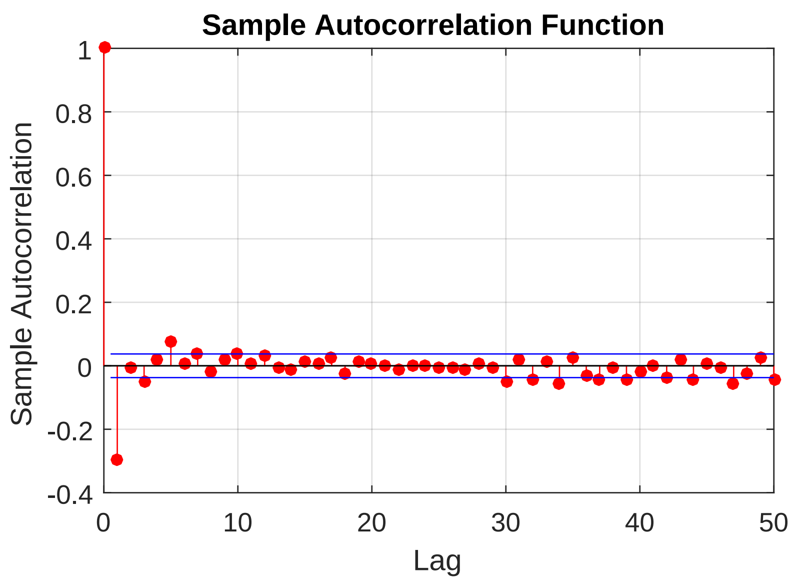


Figure 3.1. Auto correlation of stationary time series

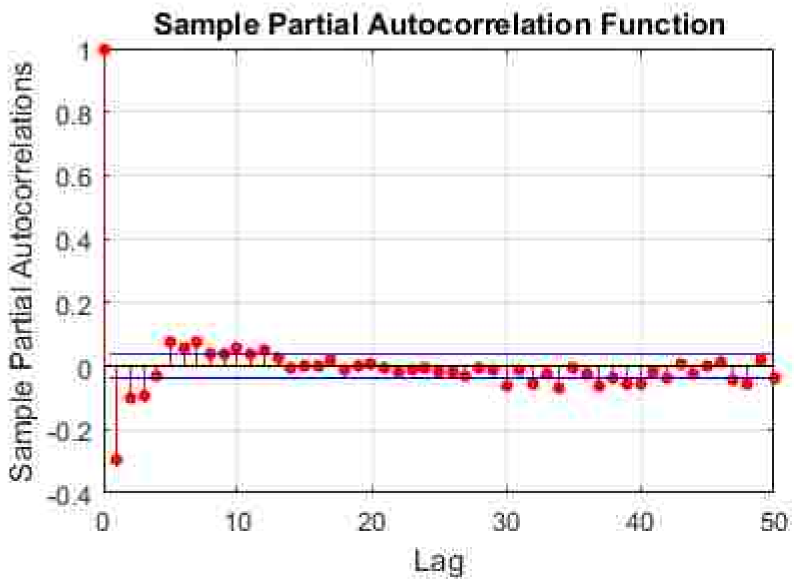


Figure 3.2. Partial correlation of stationary time series

It can be observed from Figure 3.1 & 3.2, that the auto correlation and partial correlation damps to zero after some lags. These lags can be used as ball park numbers to start with the AR & MA lags. Though these might not be the actual orders but it helps to look for the starting point. During the modeling phase, different combinations of MA & AR orders are tested and the model with highest accuracy is chosen. ARIMA equation is represented as follows:

$$\phi(p)(1-B)^d(Y_k-u)=\varphi(q) \tag{3.7}$$

$$\phi(p)=1-\sum_{i=1}^{i=p}\phi_i B^i \qquad \varphi(q)=1+\sum_{i=1}^{i=q}\varphi_i B^i \tag{3.8}$$

where,

Y_k is time series to be predicted

$\phi(p)$ is the Autoregressive function

ϕ_i is the autoregressive coefficient

φ_i is the moving average coefficient

$\varphi(q)$ is the Moving Average function

u is constant or intercept

d is differencing order

p is autoregressive order

q is moving average order

3.1.2. Seasonality. Load trends are seasonal in nature. Present data is correlated with previous day's data. Addition of seasonal terms in time series forecasting increases the accuracy of forecasting model. There is no formal method of calculating the order of seasonality. It can be observed by carefully studying the time series trend and looking for highest past cross correlation function (Figure 3.3). Plotting frequency spectrum of the time series can also help in investigating the seasonality of data set. In Figure 3.3, a load

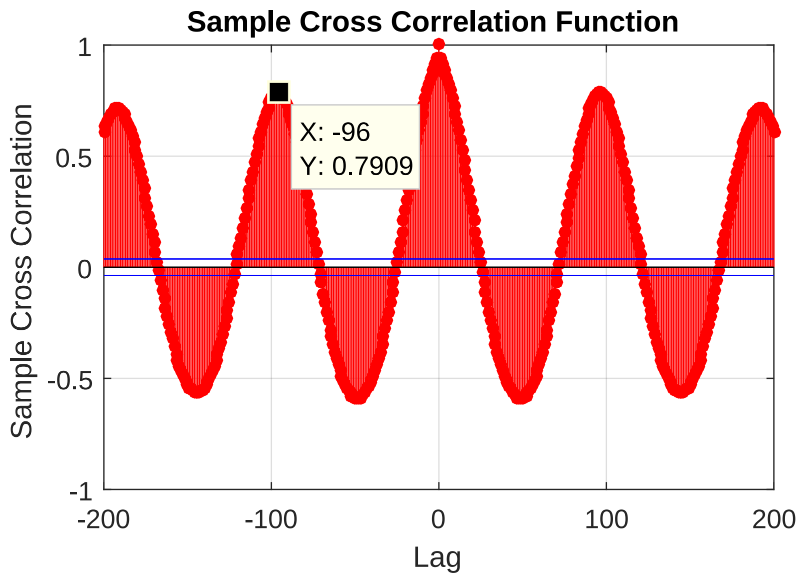


Figure 3.3. Cross-correlation of original time series

data set is discretized into 15 minutes intervals/epochs and shows that the current value is highly correlated with the past 96th epoch/past 24 hour value. This also makes sense as the typical load behavior of a consumer load does not change on a daily basis.

Therefore, the past 24 hours load value can also be used to predict the future load. Seasonality integrated with ARIMA model is called SARIMA (Seasonal Auto Regressive Integrated Moving Average) Model. The equation of SARIMA model is given below

$$\phi(p)\phi_s(P)(1-B^S)^D(1-B)^d(Y_k-u)=\varphi(q)\varphi_s(Q) \tag{3.9}$$

$$\phi_s(P)=1-\sum_{i=1}^{i=P}\phi_i^s B^i \qquad\qquad \varphi(Q)=1+\sum_{i=1}^{i=Q}\varphi_i^s B^i \tag{3.10}$$

where,

$\phi(P)$ is the Seasonal Autoregressive function

ϕ_i^s is the autoregressive coefficient

φ_i^s is the moving average coefficient

$\varphi(Q)$ is the Moving Average function

S is the seasonality of time series

D is the seasonal differencing of time series

P is the order of Seasonal Autoregressive function

Q is the order of Seasonal Moving Average function

The coefficients of SARIMA model is calculated using SAS (Statistical Analysis Tool). Other software tools like MATLAB & R can also be used for this purpose. Equation 3.9 is represented as $ARIMA(p, d, q)S(P, D, Q)$ in short. Different SARIMA were tried and model with $ARIMA(5, 1, 5)96(0, 1, 1)$ produced the best accuracy. Simulations for 24 hours ahead load forecasting is performed for one month data where the model is updated every 15minutes. Mean Average Percentage Error (MAPE) (Equation 3.15) histogram is plotted for the simulations. The histogram (Figure 3.4) represents the MAPE of 2500 day-ahead load forecasting. The minimum MAPE of the simulation is 9.7131% and the maximum MAPE is 81.9178%. As the SARIMA model predicts the future load by regressing the past value, it fails to predict the future load change due to external factors like weather. Therefore, weather parameters are important to be integrated in the system to accurately forecast the load demand.

3.1.3. Weather Correlation. Among all the dependent factor of load consumptions, weather dependent load plays a vital role in short-term load forecasting [19]. Weather components can constitute Temeprature, Humidity, Wind speed, dew point, etc. Though it depends on the consumer load and behavior which needs to be investigated before including the weather factors.

The data set being used in this research work only has correlation with temperature (0.8466) with a lag of 150 minutes. This means that the load demands react to the temperature changes after 150 minutes. Figure 3.5 shows that the load is linearly dependent on temperature but it is not necessary for all load demands. Some load demand might have a quadratic or even no relationship with load. Therefore, a thorough analysis is necessary while choosing the weather parameters.

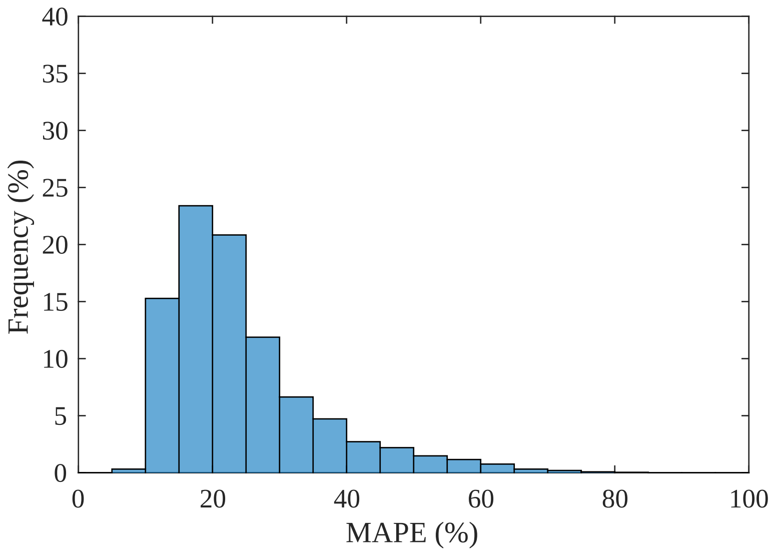


Figure 3.4. SARIMA model one month

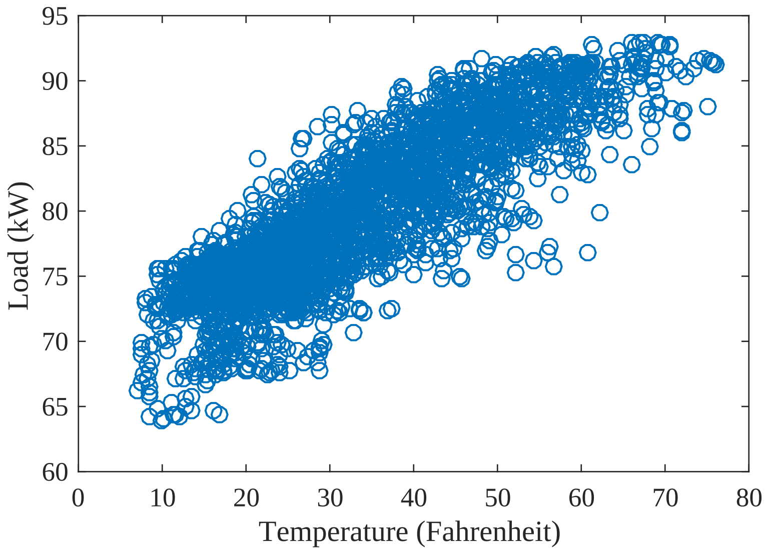


Figure 3.5. Load is linearly dependent on Temperature

3.2. LOAD FORECASTING MODELING

Load forecasting involves two steps i.e. formulating a relationship between load - weather parameters (Regression Method) and predicting the residuals regression method using SARIMA model. For example, the regression model obtained from the data set is a linear equation between load and temperature (Equation 3.11).

$$\widehat{X}[k] = u + a_{Temp} \times T[k - 10] \tag{3.11}$$

where,

$X[k]$ is the load predicted using regression method

u is the intercept equal to -129.698 for the data set taken into consideration

a_{Temp} is the regression factor of Temperature equal to 1.945

$T[k]$ is the temperature expressed in Fahrenheit

The temperature forecast is obtained from the weather stations which predict the weather for next 48 hours. Various weather stations provide with APIs which can be used to acquire the weather forecast (Appendix-A).

The residual error after applying (Equation 3.11) is predicted using SARIMA of the form SARIMA(5,1,5)X(0,1,0)96 (Equation 3.12). This form gave the best accuracy for the given data.

$$\phi(p)(1 - B^{96})(1 - B)(\widehat{e}_k - u) = \varphi(q) \tag{3.12}$$

$$\phi(p) = 1 - \sum_{i=1}^{i=5} \phi_i B^i \qquad \varphi(q) = 1 + \sum_{i=1}^{i=5} \varphi_i B^i \tag{3.13}$$

The error/residuals predicted (Equation 3.12) using SARIMA model are added to the load predicted using regression model (Equation 3.11). The addition of both the models results into the final prediction of the load (Equaiton 3.14).

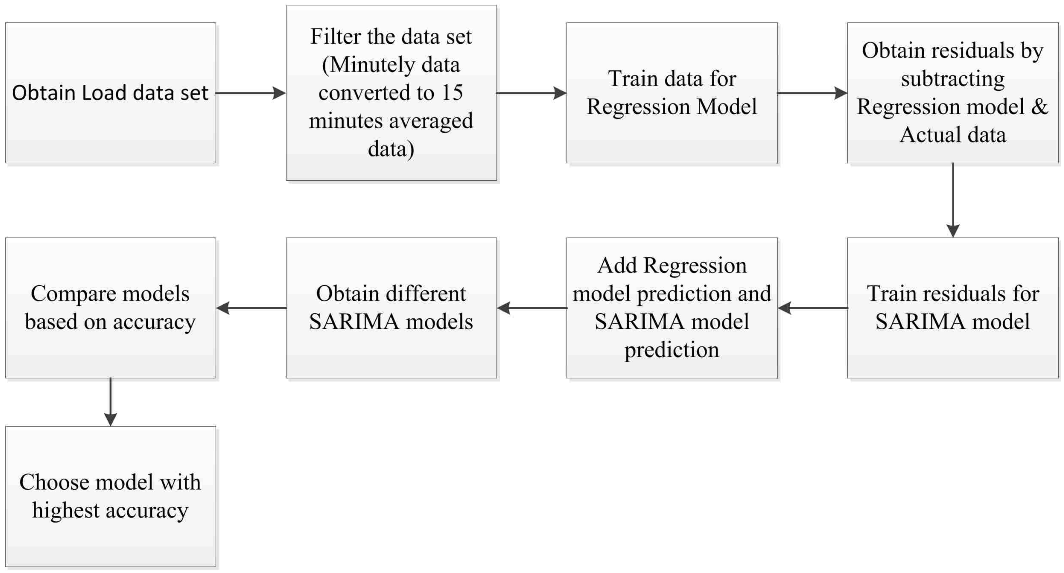

Obtain Load data set
Filter the data set (Minutely data converted to 15 minutes averaged data)
Train data for Regression Model
Obtain residuals by subtracting Regression model & Actual data
Train residuals for SARIMA model
Add Regression model prediction and SARIMA model prediction
Obtain different SARIMA models
Compare models based on accuracy
Choose model with highest accuracy

Figure 3.6. Load forecasting approach

The parameters in Equation 3.11 & 3.12 are obtained from SAS software. The flow diagram of the load prediction is depicted in Figure 3.6.

$$\hat{Y}[k] = \hat{X}[k] + \hat{e}[k] \tag{3.14}$$

where,

$Y[k]$ is the load forecasted at epoch/time step k

$X[k]$ is the load predicted using regression model

$e[k]$ is the residual predicted using SARIMA model

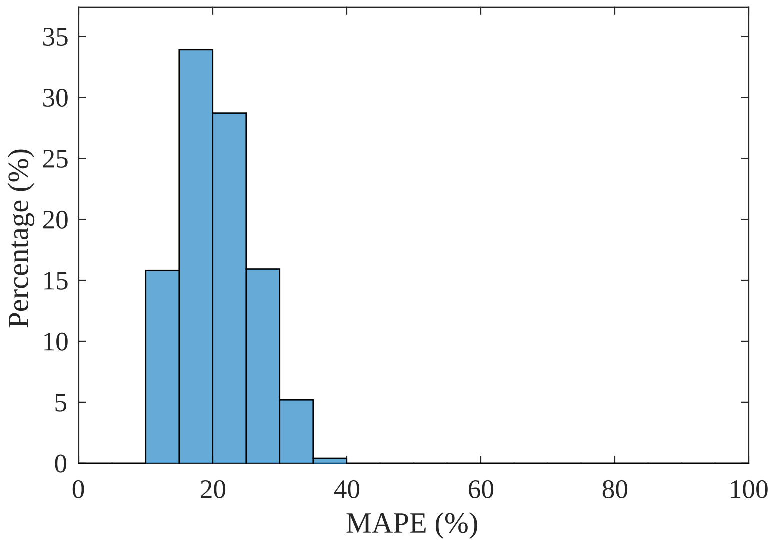


Figure 3.7. One month simulation MAPE result of load forecasting

3.3. RESULTS

The load data was obtained from Pecan Street data port. The minutely data is converted to 15 minutes averaged data. The aim of the forecasting model is to predict 24 hours ahead load demand from the current epoch. This reduces to predicting future 96 epochs of load demand. SARIMA model discussed in previous sections is used for load predictions . The forecasting model is analyzed based on MAPE (Mean Average Percentage Error) (Equation. 3.15)

$$MAPE = \frac{1}{96}\sum_{k=1}^{k=96}\frac{|Y[k]-\hat{Y}[k]|}{Y[k]} \times 100 \tag{3.15}$$

For day-ahead load forecasting updated every 15 min for 30 days, simulation results show MAPE (mean absolute percentage error) within 30% for 95% of the epochs (Figure 3.7), with a highest MAPE of 36.2% and lowest MAPE of 10%. It can be observed that the temperature parameters act as the backbone of load forecasting thereby effectively predicting the future load trend. SARIMA model is applied on top of the load predicted using weather parameter to improve the forecasting.

4. SOLAR FORECASTING MODELING

Solar energy is one of the most abundant renewable energy in the United States. Since 2008, US solar installation have grown from 1.2 GW to 30 GW which is enough to power 5.7 million American home. The PV panels have also become affordable as their price dropped by more than 60% in the past decade (Figure 4.1). Based on average prices, the system cost of installing solar panels in the US is $3.14/watt.

GTM (Green Tech Media) research suggests that the solar installation cost in India has reached as low as 65 cents per watt. Increased popularity, tax incentives, cutting edge technology are some of the reasons contributing to cheaper solar panels.

Solar energy integrated with microgrids improves the reliability of the system too. For example, in case of storms, the infrastructure of power systems is destructed due to which the solar panels directly tied to the power system stop working too. Microgrids have are more advanced and prepared for such scenarios and there smart software senses the incoming disruption. They isolate the microgrid from the power system and directly rely on their solar power and battery systems till the power system is again operating. Therefore, microgrids integrated with solar power not only provide green energy, cost reduction but also increases the reliability of the system.

All these factors contribute in encouraging the microgrid customers to install solar panels in their systems. Though, having solar panel in the system is not enough. In order to increase the penetration of solar energy, one needs to predict the upcoming solar power to better manage the energy output. There are a few solar terminologies which are needed to be discussed before discussing the forecasting models.

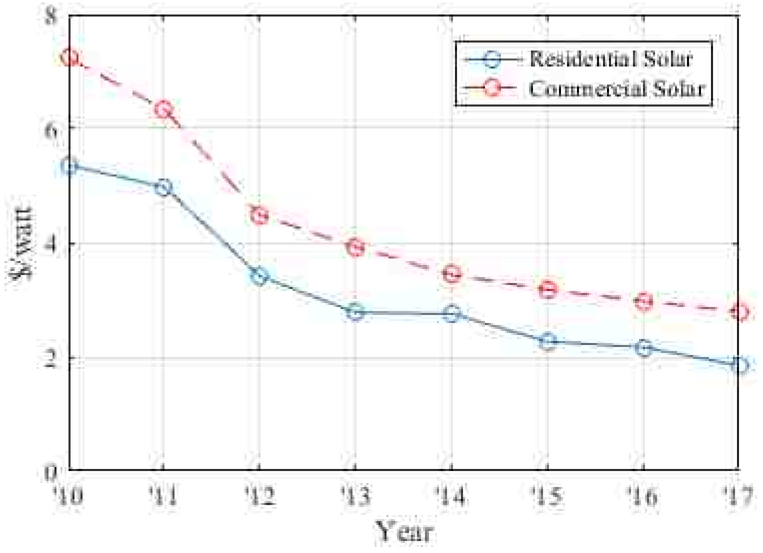


Figure 4.1. Solar panel cost reduction trend[4]

4.1. FUNDAMENTAL CONSIDERATIONS

Before diving into the solar forecasting methodologies, basic terminologies are discussed in this section.

1. Declination Angle (δ): Solar declination is the angular distance between the equatorial plane and the earth-sun line (Figure 4.2). Declination angle varies from +23.45° to −23.45° throughout the year. Solar declination can be defined as a function of day in a year as follows:

$$\delta = 23.45 \times \sin\left(360^\circ \frac{(n+284)}{365}\right) \tag{4.1}$$

where,

n=Day of the year (n=1 for 1st Jan, n=32 for 1st Feb, etc)

δ and angle inside sin are in degrees

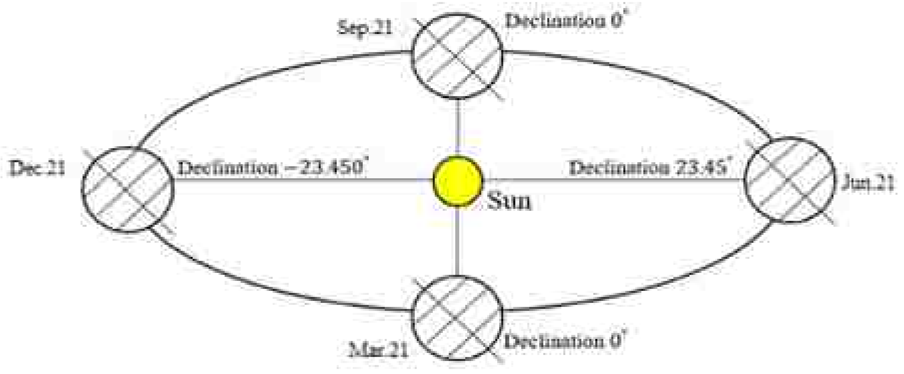


Figure 4.2. Declination angle of earth is different at different times of the day

2. Hour Angle (ω): Hour angle is an expression of observing the Sun from Earth throughout the day. It is expressed in degrees. At solar noon, hour angle is equal to 0 degrees. Time before solar noon is expressed in negative and after solar noon is expressed in positive. Sunrise/Sunset hour angle equation is described in Equation(4.2).

$$\cos(\omega_o) = -\tan(\varphi)\tan(\delta) \tag{4.2}$$

where,

ω_o is Sunset/Sunrise hour angle (Negative for sunrise and positive for sunset)

φ is Latitude of the location of interest

Earth rotates at an angular velocity of 15° hour angle/hour. Therefore, if the time of the day is known, hour angle can be calculated accordingly. Hour angle has the following constrain at a particular day:

$$-\omega_o \leq \omega \leq \omega_o \tag{4.3}$$

3. Solar Altitude (β): Solar Altitude is the angle between the horizontal plane and the line which joins the point of interest with the Sun (Figure 4.3). Expression for solar angle is given in Equation(4.4).

$$\sin(\beta) = \cos(\varphi)\cos(\delta)\cos(\omega) + \sin(\varphi)\sin(\delta) \tag{4.4}$$

where, β is Solar altitude

4. Zenith angle (ϕ): It is the angle between the vertical plane and line which joins the point of interest with the Sun (Figure 4.3). Therefore, Zenith angle and Sun altitude angle are co-dependent (Equation 4.5).

$$\phi = 90 - \beta \tag{4.5}$$

5. Azimuth Angle (θ): Azimuth angle is defined as the angular displacement of the projection of earth-sun line with south on the horizontal plane (Figure 4.3). Azimuth angle can be computed from Equation 4.6.

$$\cos(\theta) = \frac{(\cos(\omega)\cos(\delta)\sin(\varphi) - \sin(\delta)\sin(\varphi))}{\cos(\phi)} \tag{4.6}$$

where, θ is the Azimuthal angle

6. Extraterrestrial Solar Radiation: Extraterrestrial Solar Radiation E_o is the solar radiation flux just outside the EarthâĂŹ‘s atmosphere. Due to the elliptical path of Earth‘s orbit, E_o is not constant throughout the year, but can be approximated by

$$E_o = E_{sc}\left\{1 + 0.033 * cos\left[360^{\circ}\frac{n-3}{365}\right]\right\} \tag{4.7}$$

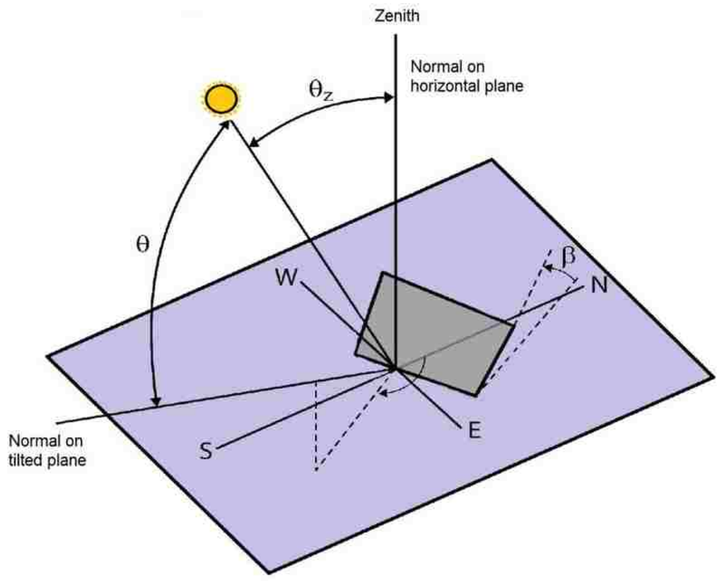


Figure 4.3. Solar Angles for horizontal and vertical surfaces[5]

where,

E_{sc} = Solar constant (1367 W/m^2)

n = Day of the year (n=1 for 1st Jan, n=32 for 1st Feb, etc)

7. Air mass (m): It is the ratio of the actual air mass present in the atmosphere to the air mass that would be present when the sun was directly overhead. Air mass is the function of solar altitude (Equation 4.8)

$$m = \frac{1}{\sin(\beta) + 0.50572(6.07995 + \beta)^{-1.6364}} \tag{4.8}$$

Where, β is the solar altitude in degrees.

4.2. FORECASTING MODELS

Total extraterrestrial solar radiation is the solar radiation received by the outer atmosphere of the Earth which fluctuates around the average value of 1360 W/m^2. This radiation is attenuated in the atmosphere by complex reflections, refractions and absorptions by various factors like clouds, aerosols, air mass, etc. Therefore, there involves a challenge in predicting the solar radiation received by the surface of the due to its complex attenuation by the atmosphere. This research work discusses three major methods to predict day ahead solar radiation / solar power output in the system i.e. Clear Sky Model, Regression Model and Non-Linear Autoregressive Neural Network Model.

4.2.1. Clear Sky Model. ASHRAE (The American Society of Heating, Refrigerating and Air-Conditioning Engineers) [16] clear sky model estimates the Global solar radiation assuming that there are no clouds present in the sky[31, 32, 33]. Broadly, Global solar radiation on a clear sky day is defined as the sum of direct solar beam and the diffused beam from the sun. These two radiances are described as follows:

$$E_b = E_0 \exp[-\tau_b m^{ab}] \tag{4.9}$$

$$E_d = E_0 \exp[-\tau_d m^{ad}] \tag{4.10}$$

$$ab = 1.454 - 0.406\tau_b - 0.268\tau_d + 0.021\tau_b\tau_d \tag{4.11}$$

$$ad = 0.507 + 0.205\tau_b - 0.080\tau_d - 0.190\tau_b\tau_d \tag{4.12}$$

Where,

$\underline{E_b}$ is the direct radiation. It is the radiation directly coming from the Sun in a straight line to the surface of the Earth (Measured perpendicular to Sun rays).

$\underline{E_d}$ is the Diffused solar radiation which is scattered by the atmospheric particles (Measured horizontal to the surface)

$\underline{E_o}$ is Extraterrestrial radiation (Equation 4.7)

m = Air Mass

$\underline{\tau_b \& \tau_d}$ are the Pseudo optical depths. These values also considers the atmospheric effects of the location. ASHRAE handbook-fundamentals provides these values for each month based on the site. These values are updated every year for better accuracy.

$\underline{ab \& ad}$ are the beam and diffused air mass exponents.

4.2.1.1. Calculation of incident radiation on a surface. Total Global radiation on a surface inclined to the horizontal plane is defined as the sum of the Direct radiation, Diffused radiation and Reflected radiation.

$$E_t = E_{t,b} + E_{t,d} + E_{t,r} \tag{4.13}$$

where,

E_t = Global solar radiation

$E_{t,b}$ = Radiation directly originating from the sun

$E_{t,d}$ = Radiation diffused by the EarthâĂŹs atmosphere

$E_{t,r}$ = Radiation after getting reflected from the ground

$$E_{t,b} = E_b cos(\theta_i) \tag{4.14}$$

E_b is direct beam radiation described in Equation 4.9 and θ_i is the angle of incidence which can be calculated from the relationship given in Equation 4.15

$$\cos(\theta_i) = \cos(\beta)\cos(\theta)\sin(\alpha) + \sin(\beta)\cos(\alpha) \tag{4.15}$$

where,

θ is Effective Azimuth Angle

α is Angle of tilt of the surface from ground

β is Solar altitude

$$E_{t,d} = E_d\left(Y sin(\alpha) + cos(\alpha)\right) \tag{4.16}$$

$$Y = max\left(0.45, 0.55 + 0.437cos(\theta) + 0.313cos^2(\theta)\right) \tag{4.17}$$

It is to be noted that Equation 4.14 and 4.16 are a modified versions of Equation 4.9 & 4.10 respectively to calculate the Solar radiations in case surface/PV panel is placed in a certain orientation w.r.t. to the horizontal plane

$$E_{t,r} = \frac{\left(E_b sin(\beta) + E_d\right)\rho_g\left(1 - cos(\beta)\right)}{2} \tag{4.18}$$

Where, ρ_g is the coefficient of ground reflectance. It has been empirically calculated for different surfaces which can be found in ASHRAE handbook-fundamentals.

4.2.1.2. Clear sky model results. Solar power output is directly proportional to the solar radiation incident on the PV panel (Equation 4.19)

$$OutputPower = Solar\_Radiation \times Area \times Efficiency \tag{4.19}$$

Preliminary requirement in case of clear sky model is to predict the solar radiation accurately. Core assumption of clear sky model is that there should be no clouds in the sky. Therefore ASHRAE model gives best performance on days without no clouds but fails to predict the solar radiation on cloudy days.

Therefore, it is necessary to add some external factors in the solar forecasts. ASHRAE model alone cannot be used to predict the solar output but it can act as a backbone structure to predict the solar radiation which will be discussed in following sections.

4.2.2. Regression Model. Solar radiation is dependent on Earth-Sun geometry, the cloud cover and the weather[34, 35, 36]. Therefore, if the weather is known beforehand, solar radiation can also be predicted. Weather stations have past years weather data base, satellite imageries and complex algorithms to predict the weather with up to one week

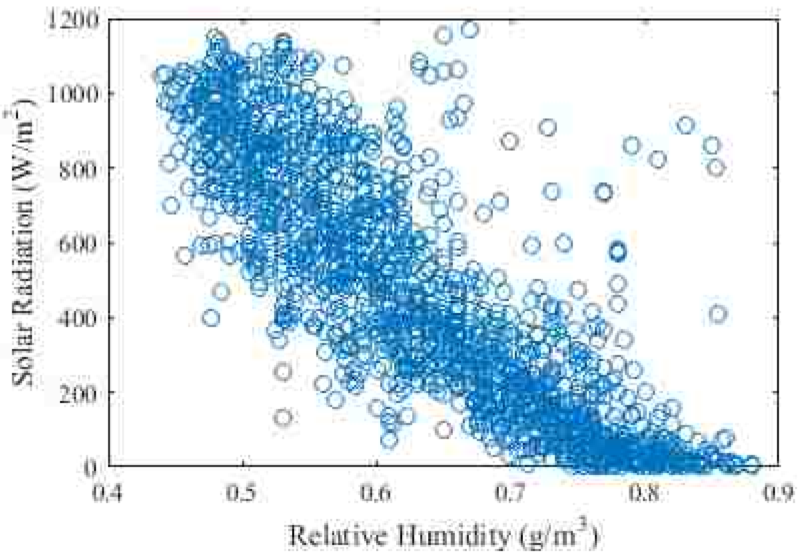


Figure 4.4. Solar radiation & Relative humidity are linearly correlated with a correlation coefficient of -0.8402

of prediction window. Most of the weather stations provide APIs to give 48 hours ahead weather forecast with a time step of one hour. Though, the weather is affected by the solar radiation but if we can use the reverse technique and predict the solar radiation from the predicted weather. Weather stations mostly provide with temperature, relative humidity, dew point, wind speed, precipitation, etc. The correlation between solar radiation/solar power output and other weather parameters are needed to be analyzed before moving to the modeling part. From the available dataset, temperature (Figure 4.4) and humidity (Figure 4.5) had the highest correlation with solar radiation.

Clear sky model gives the solar radiation available on a clear sky day. This means that it is the maximum solar radiation that is incident on the surface of solar panel. Therefore, ASHRAE clear sky model is taken as a reference and regressed against temperature and relative humidity forecasts. ASHRAE model consists all the necessary information for predicting the solar radiation such as latitude, time of day, air mass, reflectivity around the PV panel, etc. Rest of the information like weather parameters, efficiency of PV panel, area

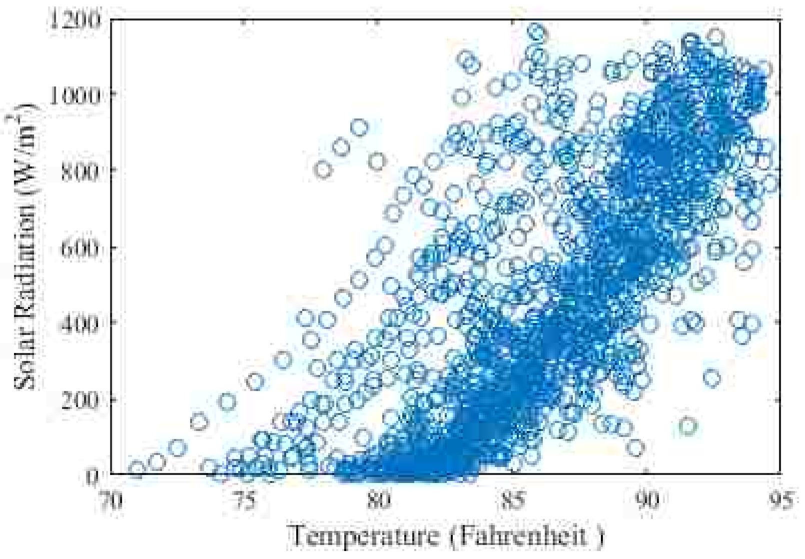


Figure 4.5. Solar radiation & Temperature are linearly correlated with a correlation coefficient of 0.7429

of PV panel is included in the model using Regression model (Equation 4.20)

$$S_k = \mu + ASHRAE_k \times \alpha + Temperature_k \times \beta + Humidity_k \times \chi \qquad (4.20)$$

where,

S_k is the predicted solar radiation at epoch k

$ASHRAE_k$ is the ASHRAE model prediction at epoch k

$Temperature_k$ is the temperature forecast at epoch k acquired from weather station

$Humidity_k$ is the relative humidity forecast at epoch k acquired from weather station

μ is the intercept equal to 345.36433 for available dataset

α is the regression factor of ASHRAE model 0.59383 for available dataset

β is the regression coefficient of temperature forecast 2.44879 for available dataset

χ is the regression coefficient of relative humidity -743.69570 for available dataset

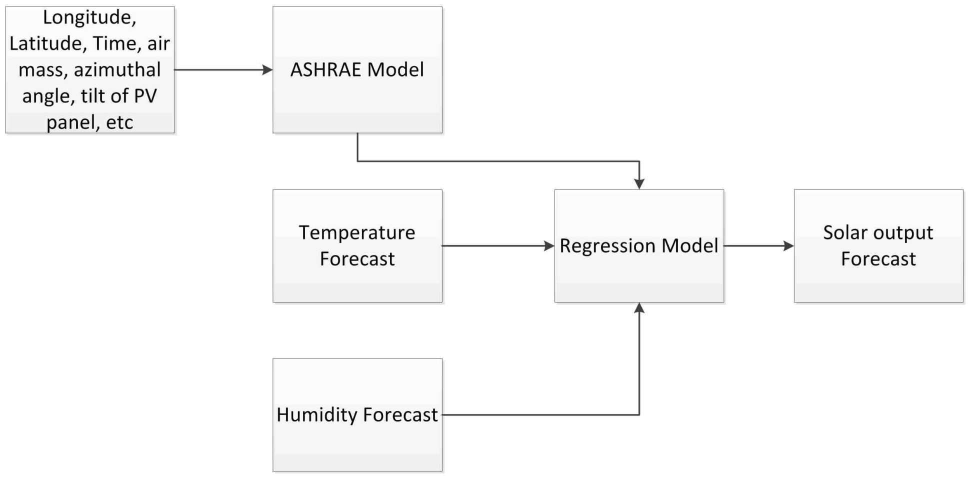


Figure 4.6. Regression model flow diagram

Regression factors are calculated using SAS software. The performance of solar output is analyzed based on NAPE (Normalized Average Percentage Error) and MAE (Mean Average Error) Equation 4.21 and Equation 4.22 respectively.

$$NAPE = \frac{1}{96}\sum_{k=1}^{k=96}\frac{\left|S[k]-\widehat{S}[k]\right|}{\max\left(S[k]\right)} \times 100 \tag{4.21}$$

$$MAE = \frac{1}{96}\sum_{k=1}^{k=96}\left|S[k]-\widehat{S}[k]\right| \tag{4.22}$$

The errors are normalized because it is difficult to analyze the performance of solar radiation when the output is near to zero; small errors can generate high error percentage which is insignificant in practical applications but reflects poor models in histograms. Therefore, errors are normalized to analyze the models in an efficient way. Regression model flow diagram is depicted in Figure 4.6.

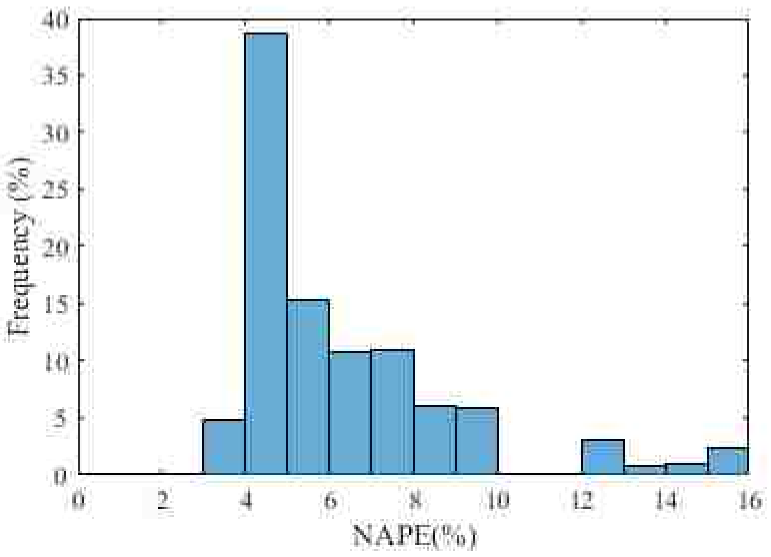


Figure 4.7. Regression NAPE histogram

Simulation for one month is performed where solar radiation is predicted from sunrise to sunset and updated every 15 minutes (1 epoch). Therefore, there are 1300 simulations. In order to analyze the performance of model, NAPE & MAE histogram are plotted (Figure 4.7 & 4.8) to visualize performance of model.

4.2.3. Neural Network Model. NARX (Nonlinear Autoregressive exogenous model) is a nonlinear autoregressive process which uses both past values of the time series being predicted and current & past values of the exogenous series (temperature, clear sky prediction, relative humidity). NARX combines the properties of both Autoregressive and Neural networks[37, 38, 33]. PV output is not only dependent on solar radiation but area and efficiency of solar panel too. Though the area of solar panel is constant but the efficiency is not. The efficiency of the solar panel depends on a lot of factors like temperature, type/magnitude of load, material of silicon, aging, etc. Therefore, NARX not only predicts the solar radiation but the efficiency of the system too. NARX model has been developed with the help of MATLAB Neural Network Time Series Tool Box. The tool box requires the user to input the number of lags of both time series being predicted and exogenous

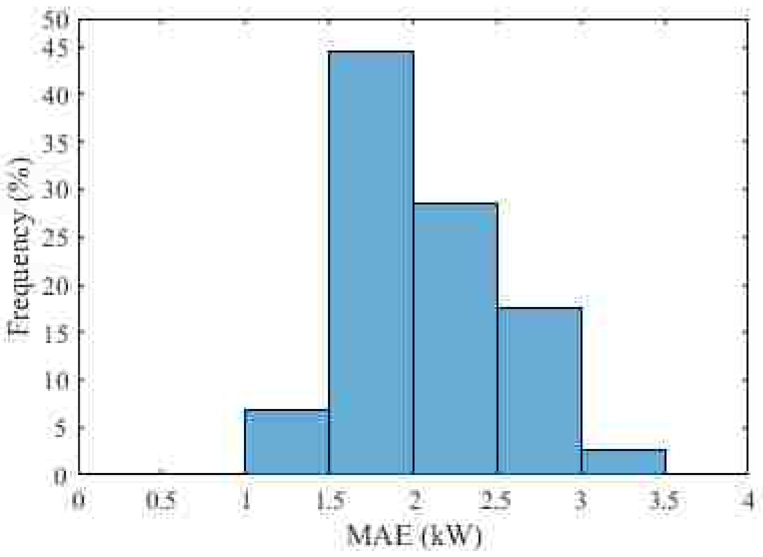


Figure 4.8. Regression Mean Absolute Error (MAE) histogram

series, the number of neurons in the hidden layer and dataset of all the time series to train the Neural network. The activation function of neurons in the hidden layer is sigmoidal (Equation 4.23)

$$F(n) = \frac{2}{1 - e^{2n}} - 1 \tag{4.23}$$

$$n = \sum_{i=1}^{d_y} y\,(t - i)\, W_i^y + \sum_{i=1}^{d_x} x\,(t - i + 1)\, W_i^x + b \tag{4.24}$$

where,

d_y is the delay of predicted series

d_x is the delay of exogenous series

W_{yi} are the weights of predicted series to the neuron

W_{xi} are the weight of the exogenous series to the neuron

b is the offset of the neuron

All the offsets and weights are calculated by the tool box. The flow diagram of NARX model is shown in Figure 4.9. The number of neurons in hidden layer & number of lags are decided by trial & error method. Different models are compared based on their

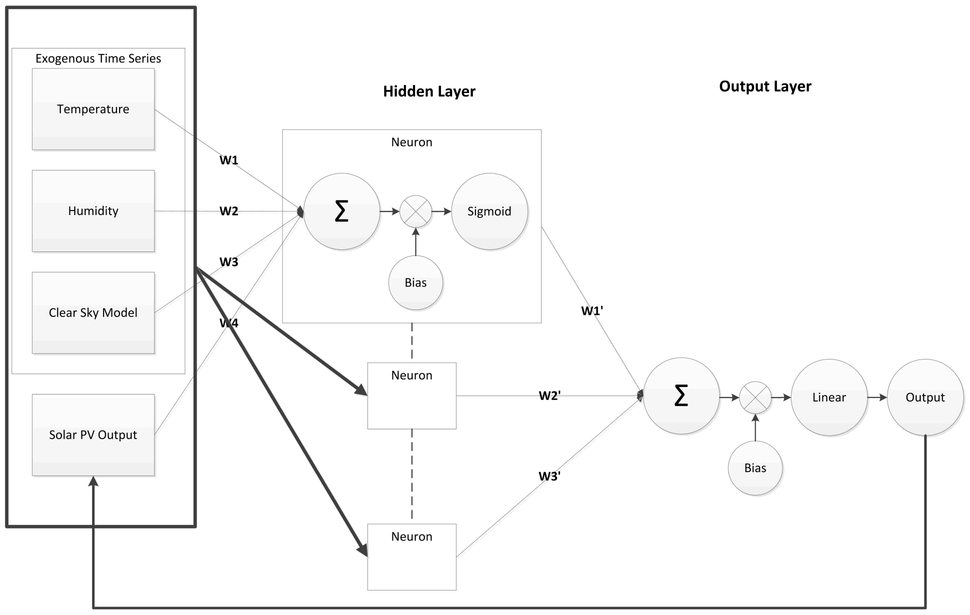


Figure 4.9. Flow diagram of NARX model

NAPE histogram. Model with 15 neurons in hidden layer and 1 lags are chosen. It can be observed that the performance of NARX method (Figure 4.10 & 4.11) outperforms the performance of Regression model. The NAPE & MAE histogram plot in case of NARX model have converged towards zero indicating better performance.

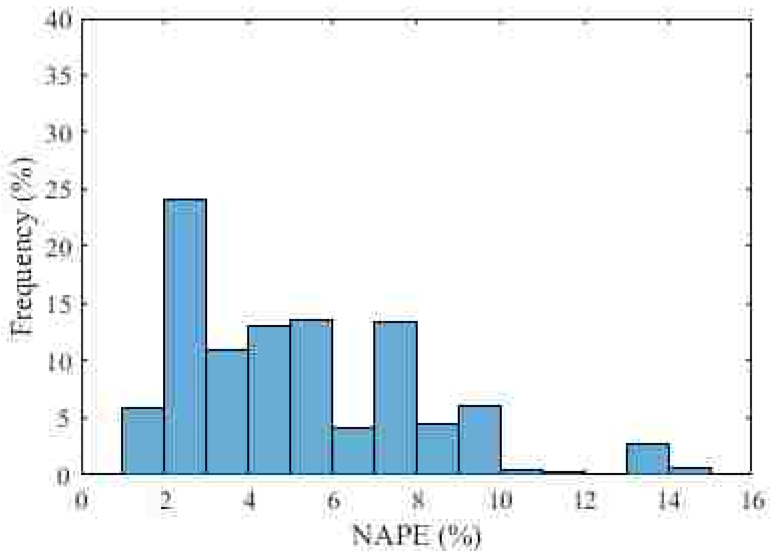


Figure 4.10. NARX NAPE histogram

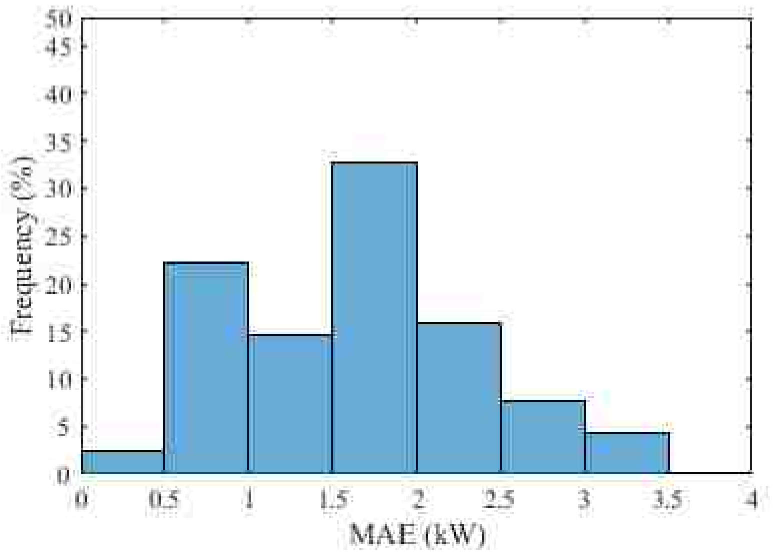


Figure 4.11. NARX MAE histogram

5. RESULT

In previous chapters, formulation of Markov Decision Process (MDP) for battery scheduling in microgrids is discussed. The results of MDP show a reduction of maximum load demand from 74 kW to 34 kW. This results in cost reduction of \$929.40 in one month. Though these figures are theoretical limits of the model. In other words, control algorithm implemented with MDP model cannot achieve further accuracy. When the Load and PV forecasts are integrated with MDP model, there errors are also induced. Load demand and PV output are highly stochastic in nature therefore, 100% accuracy cannot be achieved in practical applications. Therefore, for realistic implementation of MDP model, load and solar forecasts are necessary to realize a battery management system. This chapter discusses the effects in cost and maximum load demand when forecasting models are added to the system. Also, there is a need to compare the MDP model with a heuristic approach to prove that the microgrid system requires a stochastic approach in order to control the battery systems. The following section discusses the formulation of a heuristic approach to schedule the battery system in microgrid.

5.1. HEURISTIC METHOD TO SCHEDULE BATTERY

Battery scheduling method is developed to compare it with the MDP control method. The heuristic method does not forecast load and solar data. The modules tries to maintain the grid power to 33 kW. The only boundary condition imposed on the system is that the energy stored in the battery bank cannot exceed minimum (0 kWh) and maximum (200 kWh) energy.Following steps demonstrates on calculating a scheduling policy of battery using a heurisitc approach.

Step 1: Assume that the maximum power drawn from the grid should not exceed 33 kW. Therefore the battery actions can be found from the following equation:

$$BatteryPower(k) = Load(k) - Solar(k) - 33 \tag{5.1}$$

Equation 5.1 shows how should the battery charge/discharge into order to consistently draw 33 kW from the grid. This policy is calculated from the first day of the month and would be repeated everyday keeping into considerations the boundary conditions of the battery bank energy.

The equation with which the policy is derived does not consider the energy stored in the battery. Therefore, battery boundary conditions are applied in the next step.

Step 2: Following is the Pseudocode to ensure that the energy is within limits in the battery

1. Initialize energy stored in battery "E" before starting the simulation (100 kWh in our case) and epoch(k)=1

2. Convert Battery power to energy stored/withdrawn from battery using Equation 5.2

$$E(k) = \begin{cases} P(k)\frac{dT}{\eta_c} & P(k) > 0 \\ 0 & P(k) = 0 \\ P(k)\eta_d dT & P(k) < 0 \end{cases} \tag{5.2}$$

3. Compute energy stored in battery after 1 iteration $E = E - E(k)$

4. Check if $0 \leq E \leq 200$

   (a) If Yes, $Action(k) = E(k)$

   (b) If No, Increase or decrease E(k) according to the sign of E(k) till $0 \leq E - E(k) \leq 200$ and store $Action(k) = E(k)$

5. Increment k=k+1

6. Goto 2

After running the simulation for 30 days, the maximum load demand is 62.27 kW whereas the maximum load demand without any battery system is 74.4 kW.

5.2. SIMULATION RESULT

The MDP integrated with load and solar forecast is simulated for one month. The load and solar data are acquired form Pecan Street Dataport. A community of 20 houses is assumed with ten houses having PV panels installed. Demand charge rate structure is assumed which is adopted from Rolla Municipal Utility (RMU) with parameters given in Table 5.1.

In demand rate structure, the customer is penalized based on the maximum load drawn

Table 5.1. Demand rate structure used in simulations (Adopted from Rolla Municipal Utilities)

Parameter	Rate
Buying Rate (θ_b) ($/kWh)	0.07009
Selling Rate (θ_s) ($/kWh)	0.01326
Demand Rate (θ_d) ($/kW)	14.5

from the grid during a 15 minutes time period in a month. The demand rate structure is typically above $12 due to which it becomes the major part of the electricity bill. Therefore, peak shaving plays in an important role in reducing the billing cost.

30 days simulation captures the performance of the battery scheduling system as it covers all types of days like cloudy, partly cloudy and clear sky which play an important role in affecting the PV output and load demand. The MDP without forecasting errors gives a cost saving of 47.7% which is a theoretical limit of the model. Therefore, load and solar forecasting models are important to be integrated with the MDP framework to come up with

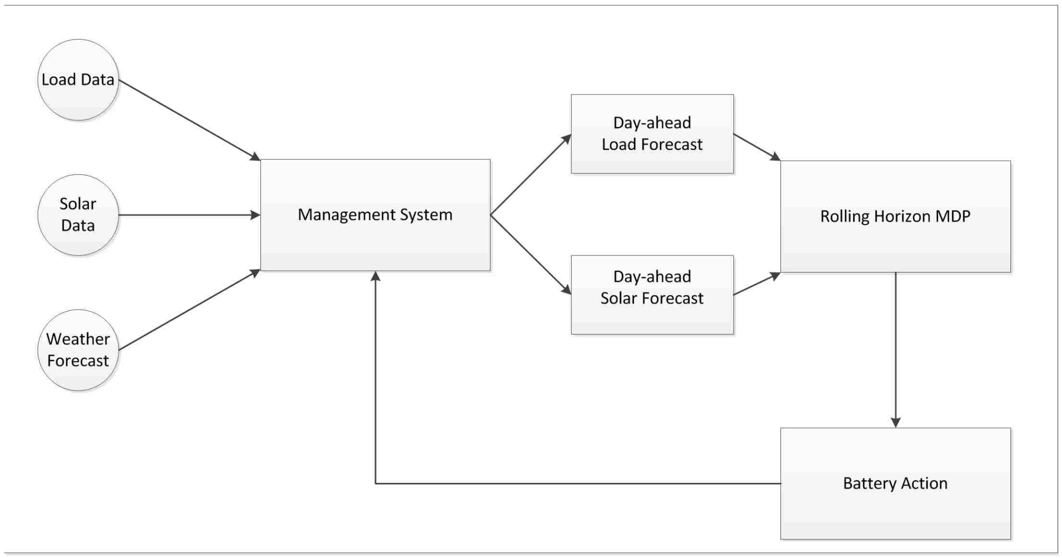


Figure 5.1. Implementation of Management system topology

a realistic system. The improved SARIMA model for load forecasting and NARX model for Solar forecasting are integrated with MDP to realize a battery management system. The block diagram of the proposed energy is system is shown in Figure. 5.1.

The initial condition of Battery SoC is assumed to 50% and the day is discretized into 15 minutes time interval. The battery system is assumed to have 200 kWh capacity which is discretized into 200 bins while the charging and discharging efficiency of bi-directional inverter connected to the battery system is 95%. A total of 37 charging and discharging actions in kWh $\in [-17, 18]$ are available in the system. A one month simulation results are given in Table 5.2.

It can be observed from Table 5.2, that the MDP model integrated with solar and load forecasting is able to shave the maximum peak to 54.01 kW thereby saving a total cost of \$687.6 for one month billing cycle. MDP model without load and solar forecasting saves 47% in one month while model with load and solar forecasting saves 35% which is a difference of 12%. This change is caused because of the stochastic nature of load demand and PV output.

Table 5.2. One month MDP simulation with solar and load forecasting

	No Algorithm	MDP Algorithm with zero forecasting errors	MDP with Load and Solar forecasting errors
Cost ($)	1944.9	1015.49	1257.3
Savings ($)	NA	929.4	687.6
Maximum Load (kW)	74.4	34	54.01

5.3. CONCLUSION

Markov Decision Process is a mathematical framework which proves to be effective in peak shaving and lowering the system cost in microgrids. The rolling horizon MDP is implemented using dynamic programming which is invoked every 15 minutes in order to make sure that the next 24 hours load demand and PV output are taken into consideration. During one month simulation, MDP framework successfully shaves the maximum peak from 74.4 kW to 34 kW thereby saving 47% in the monthly billing cycle.

The MDP is incomplete without introducing Load and Solar forecasting it. Therefore, different modeling techniques have been discussed in this work to generalize a method to generate load and solar forecasting models for different locations. Load forecasting is implemented using improved SARIMA model and PV forecasting is performed using Neural Network with exogenous inputs. Both the forecasting models are supported by weather forecasts acquired from weather stations which help to increase the forecasting accuracy of the models. After introducing forecasting models with MDP, the maximum load in the system is 54 kW and monthly saving of 35%. The difference of 12% between the models with and without forecasting models is due to the errors in predicting load demand and PV output. Load and PV output are highly stochastic in nature therefore, errors are bound to be introduced in the system which reduces efficiency of MDP framework. In order to achieve the high accuracy in the system, better forecasting models can be integrated thereby, reducing the errors in the system and monthly billing cost.

APPENDIX A

API FOR ACQUIRING WEATHER FORECAST FROM DARKSKY.NET

API FOR ACQUIRING WEATHER FORECAST FROM DARKSKY.NET

```
import requests


class forecastio(object):

#define the forecast class to initialise api key, longitude
    and, latitude

    def __init__(self,lat=37.958534,long= -91.774461,api='
        daa69534a0f7809fbb190745b647717b'):

        self.latitude=lat
        self.longitude=long
        self.forecastio_api=api
        self.url='https://api.darksky.net/forecast/'
        self.data=0       #Initialise vaiable to store all the
            data


#Generate URL based on longitude, latitude and, API key
    def url_gen(self):
        url=self.url+self.forecastio_api+'/'+str(self.
            latitude)+','+str(self.longitude)
        return(url)
```

```
#Fetch data from the server
#This includes all the information i.e. hourly, daily
   current, etc
      def get_data(self):

          url=self.url_gen()
          resp=requests.get(url)
          self.data=resp.json()



#Inhereted from forecastio class to manipulate data
class forecast(forecastio):

          def __init__(self,lat=37.958534,long= -91.774461,api
             ='daa69534a0f7809fbb190745b647717b'):
                    forecastio.__init__(self,lat=37.958534,long=
                         -91.774461,api='
                       daa69534a0f7809fbb190745b647717b')



#Generic function to write data to files
          def write_file(self,param,type):
                    string=param+'_'+type+'.txt'
                    file=open(string,'w')
                    for m in self.data[type]['data']:
                              file.write(str(m[param]))
```

```
                file.write('\n')
            file.close()

#Fetch hourly data. Can be any desired arguments present in
   the json file.
#Arguments are case sensitive
    def hourly(self,*args):

        for arg in args:

            self.write_file(arg,'hourly')
#Fetch minutely data. Can be any desired arguments present
   in the json file.
#Arguments are case sensitive

    def minutely(self,*args):

        for arg in args:

            self.write_file(arg,'minutely')

#Fetch daily data. Can be any desired arguments present in
   the json file.
#Arguments are case sensitive

    def daily(self,*args):
```

```
                for arg in args:

                        self.write_file(arg, 'daily')
```

IMPLEMENTATION OF WEATHER FORECASTING API

```
from forecastio import forecast

lat='37.951935'      #Latitude of Solar Village
long='-91.779811'    #Longitude of Solar Village
api_key='daa69534a0f7809fbb190745b647717b'  # API key
    receive from www.Darksky.net

forecast_object=forecast(lat,long,api_key)  # Create object
    with required parameters

forecast_object.get_data() # Fetch data from server




forecast_object.hourly('temperature','cloudCover','icon') #
    Write desired data to a text file
forecast_object.minutely('precipIntensity')
forecast_object.daily('temperatureMin')
```

APPENDIX B

IMPLEMENTATION OF MARKOV DECISION PROCESS

IMPLEMENTATION OF MARKOV DECISION PROCESS (MDP.C)

```c
#include <stdio.h>
#include <stdlib.h>
#include "MDP.h"
#include "init.h"
#include "Cal.h"
#include "ARMA.h"

#define M 200       // Maximum energy of battery
#define K 96        // Epoch
#define N 37        // Number of Actions available
#define Ebin 1      // Minimum Energy change
#define Emin 0      // Minimum Energy
#define theta 0.5 // Cost-to-go
//#define Grid_avg 18.4473 //Avg power set for load demand
#define Grid_avg 33 //Avg power set for load demand
//Initialize Set of Actions Available
int At[37]={-17,-16,-15,-14,-13,-12,-11,-10,-9,-8,-7,
-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
15,16,17,18,19};

//ALL Avg data (Data can be entered here for convinience)
double Load[1440]={};
double Load_si[1440]={};
//ALL Avg data
//double Solar[1440]={};
```

```
double Solar_si[2880]={};


int main(){
        double Load_roll[96]={0};  \\Forecasted Load will be
            stored here
        double error_map[97]={0};  \\ Past 5 errors of
           forecasted load are stored here
        double *load_ptr;          \\ Pointer pointing
           forecasted load
        double *error_maptr;       \\ Pointer pointer Load
           errors
        double Solar_roll[96]={0}; \\ Forecasted Solar Power
            sotred here
        double *Solar_ptr;
        Solar_ptr=Solar_roll;

        int i,j,k,n,a,horizon,count,epoch;

load_ptr=Load_roll;
error_maptr=error_map;

FILE *ar;    //Forecasted load stored in file system for
   analysis
FILE *er;    // Past 5 errors stored in file system for
   simulaiton
```

```
FILE *sol; //Forecasted solar stored in file system for
    analysis

ar=fopen("Load.txt","w+");
sol=fopen("Solar.txt","w+");
er=fopen("error.txt","r");

double m;
Optimum m1;    //Structure used for MDP
Optimum *ptr;
ptr=&m1;

Val_min mini;   //Structure stores minimum values
Val_min *m2;
m2=&mini;

double *ptr_ar;

ptr_ar=Load;
double Cost;

Var_init(ptr);   //Initialize Variables



char line[100];
char line1[255];
```

```
int pos=99;    // Define Initial SoC of Battery


for(horizon=96*2+76-1;horizon<96*28+76;horizon++){    //
   Specify how many horizons to run.


i=0;
j=0;


//Read Load errors from file system. Initialized to zero
while(fgets(line,sizeof(line),er2)){


for(i=0;i<100;i++){


     if((line[i])=='\n'){
          (line[i])='\0';
          i=257;
     }
}
*(error_maptr+j)=atof(line);


j++;
}
fclose(er2);


ARMA(horizon,load_ptr,error_maptr); //Predict Load using
   ARMA
```

```
Neural_Network(horizon,Solar_ptr);   //Predict Solar Power
   using NN

//Enter errors from forecasted data
er2=fopen("error.txt","w");
for(i=0;i<5;i++){

    fprintf(er2,"%f\n",*(error_maptr+i));
}

fclose(er2);

Var_init(ptr);   //Initialize Variables
//Set Cost-to-go function
 for(i=0;i<M;i++){
    ptr->U[i][K-1]=-theta*(Emin+(i+1)*Ebin+Emin+(i)*Ebin)/2;
 }
epoch=94;
count=0;

while(epoch>=0){

    for(i=0;i<M;i++){

        for(n=0;n<N;n++){

            a=At[n];
```

```
                    if((i-a)>=0 && (i-a)<M){

                        j=i-a;
                        Cost=Temp_Cost(i,j,a,epoch,horizon,load_ptr,
                            Solar_ptr);  //Calculate utility cost

                        ptr->Ut[i][n]=Cost+ptr->U[j][epoch+1];

                    }
                }
                minimum1(m2,ptr->Ut,i);     //Calculate minimum
                    Utility
                ptr->U[i][epoch]=m2->minimum;
                ptr->Final_action[epoch][i]=At[m2->pos];   //Store
                    final action

        }
    epoch--;
    count++;

    //Initialize Utility for next MDP simulaiton
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){

            ptr->Ut[i][j]=1000000;

        }
```

```
}

  }

  printf("%d\n",ptr->Final_action[0][pos]); //Print Aciton
     taken on Console
     pos=pos-ptr->Final_action[0][pos];

}

return(0);
}
```

MDP.H

```
#ifndef MDP_H_INCLUDED
#define MDP_H_INCLUDED

#define M 200        // Maximum energy of battery
#define K 96         // Epoch
#define N 37         // Number of Actions available
#define Ebin 1       // Minimum Energy change
#define Emin 0       // Minimum Energy
#define theta 0.5 // Cost-to-go
//#define Grid_avg 18.4473f //Avg power set for load demand
#define Grid_avg 33.1844 //Avg power set for load demand

extern int At[37];
extern int i;
extern double Load[2880];
extern double Solar[2880];
extern double Load_si[2880];
extern double Solar_si[2880];
extern double stad_dev[96];
typedef struct {
double U[M][K];      //Final utility function
double Ut[M][N];     //Temp Utility function
double Cost[M][K]; //Cost function
int actions[K][M]; //Actions with different Utilities
int Final_action[K][M]; //Optimum Actions
}Optimum;

typedef struct{
double minimum;
int pos;
}Val_min;

#endif // MDP_H_INCLUDED
```

INITIALIZATION OF MDP VARIABLES (INIT.C)

```
#include <stdio.h>
#include "MDP.h"
#include "init.h"

int i,j,k,count;
```

```
//Initialise all variables
void Var_init(Optimum *ptr1){

//====Final Utility & Cost Initialisation==========
//printf("Final Utility & Cost Initialised...\n");
for(i=0;i<M;i++){

      for(j=0;j<K;j++){


            ptr1->U[i][j]=0;
            ptr1->Cost[i][j]=0;

                  }
}

//====Temp Utility Initialisation==========
//printf("Temp Utility Initialised...\n");
for(i=0;i<M;i++){
      for(j=0;j<N;j++){

            ptr1->Ut[i][j]=10000;


      }
}
//====Action & Final_actions Initialisation==========
//printf("Final Action & Action Initialised...\n");
for(i=0;i<K;i++){
    for ( j =0; j <M; j ++) {

        ptr1 ->actions [ i ][ j ]=0;
        ptr1 ->Final_action [ i ][ j ]=0;


    }

}
}
```

APPENDIX C

IMPLEMENTATION OF SARIMA MODELING

IMPLEMENTATION OF SARIMA MODELING (SARIMA.C)

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "MDP.h"
#include "Cal.h"

double prev_error;
double prev_value;

void ARMA(int horizon, double *ptr, double *error_maptr){
int i,j,index_count,line;
char data[512];
double sum=0;
double P[2689];
double *ptr_P;
ptr_P=P;
double AR1=0.08651;
double AR2=0.38199;
double AR3=-0.31334;
double AR4=0.39401;
double AR5=0.05516;
double MA1=-0.57553;
double MA2=-0.44625;
double MA3=0.38522;
double MA4=-0.54557;
double MA5=0.19079;
double u=0.002048;

double error[2689];

double Intercept[2871]={};
double noise[2871]={};

//Implement SARIMA==============================

j=4;
int t;
t=0;

for (i=horizon1;i<horizon1+96;i++){

P[i]=-(AR1*(P[i-2]+ P[i-97] - P[i-98]-P[i-1]) - P[i-1] + AR2
   *(-P[i-2]+ P[i-3] + P[i-98]  -P[i-99] )     + AR3*(-P[i-3]
    +P[i-4]      + P[i-99]  - P[i-100]) +AR4*(-P[i-4] + P[i-5]
    + P[i-100]  - P[i-101]  )    +AR5*(-P[i-5]  + P[i-6]    +
   P[i-101]  - P[i-102])  - P[i-96]  + P[i-97]   );

P[i]=P[i]+MA1*error[j]+MA2*error[j-1]+MA3*error[j-2]+MA4*
    error[j-3]+MA5*error[j-4];
j++;
t++;
```

```
 }

 error[5]= -P[horizon1]+noise[horizon1];
index_count=97;

for(i=0;i<=95;i++){

            *(ptr+i)=P[horizon1+i]+Intercept[horizon1+i];

  }
}

for(i=0;i<5;i++){
    *(error_maptr+i)=error[i+1];

}

}
```

APPENDIX D

IMPLEMENTATION OF NEURAL NETWORK MODEL

PV OUTPUT PREDICITON USING NEURAL NETWORK MODEL

```
#include <stdio.h>
#include <math.h>



double mapminmax_apply(double x,double settings_gain ,double
    settings_xoffset , double settings_ymin){


double y;
y=x−settings_xoffset;
y=y∗settings_gain;
y=y+settings_ymin;

return(y);
}

void transig_apply(double ∗n, double ∗out){
int i;

for(i=0;i<15;i++){

  ∗(out+i)=(2/(1+exp(−2∗(∗(n+i)))))−1;

}
}
```

```
double mapminmax_reverse(double y,double settings_gain,
   double settings_xoffset,double settings_ymin){

double x;
x=y−settings_ymin;
x=x/settings_gain;
x=x+settings_xoffset;
return(x);

}



void Neural_Network(int horizon,double *solar){

double Temp_new2[2672]={};
double Hum_new2[2672]={}
// Input 1 Normalize the Inputs
double x1_step1_xoffset_temp = 70.84;  \\temperature offset
double x1_step1_xoffset_Et = 6.07953724213161; \\Clear Sky
   offset
double x1_step1_xoffset_hum = 0.36; \\Humidity offset

double x1_step1_gain_temp = 0.0903342366757001; \\
   Temperature gain
double x1_step1_gain_Et = 0.00190970262240737; \\Clear sky
   gain
double x1_step1_gain_hum = 3.3195020746888; \\Humidity gain
double x1_step1_ymin = −1; \\boundry

//Input 2
double x2_step1_xoffset = −0.160733333333333; \\Solar
   radiation offset
double x2_step1_gain = 0.0510055749093376; \\PV output gain
double x2_step1_ymin = −1; \\boundry

//Layer1
double b1[] = {}; \\Define b1 from model
double IW1_1_temp[] ={}; \\Define Temperature weights from
   MATLAB
double IW1_1_Et[] ={}; \\Define Clear Sky weights from
   MATLAB
double IW1_1_hum[] ={}; \\Define  Humidity weights from
   MATLAB
double IW1_2[] = {}; \\Define PV Output weights from MATLAB

// Layer 2
double b2 = −1.8322820572359226; \\Layer 2 offset
double LW2_1[] = {}; \\define layer 2 weights form MATLAB

// Output 1
double y1_step1_ymin = −1; \\Output boundary
double y1_step1_gain = 0.0510055749093376; \\Output gain
double y1_step1_xoffset = −0.160733333333333; \\output
   offset
```

```
double xd1_temp , xd1_Et , xd1_hum , xd2_pv , a2 ;
int i , ts , count ;
double tapedelay1_temp , tapedelay1_Et , tapedelay1_hum ,
    tapedelay2 , final_result [15] , result_out [15] , y1 [96] , Gen3
    [96];
double *final_result_ptr ,*out ;
final_result_ptr=final_result ;
out=result_out ;
double c ,d ,e ;
        xd1_temp=mapminmax_apply (Temp_new2[ horizon ] ,
            x1_step1_gain_temp , x1_step1_xoffset_temp ,
            x1_step1_ymin ) ;
        xd1_Et=mapminmax_apply ( Et_new2 [ horizon ] ,
            x1_step1_gain_Et , x1_step1_xoffset_Et , x1_step1_ymin )
            ;
        xd1_hum=mapminmax_apply (Hum_new2[ horizon ] ,
            x1_step1_gain_hum , x1_step1_xoffset_hum ,
            x1_step1_ymin ) ;
        xd2_pv=mapminmax_apply ( pv [ horizon −1] , x2_step1_gain ,
            x2_step1_xoffset , x2_step1_ymin ) ;
        d=horizon /96;

        for ( ts =0; ts <96; ts ++) {
                tapedelay1_temp=xd1_temp ;
                tapedelay1_Et=xd1_Et ;
                tapedelay1_hum=xd1_hum ;
```

```
tapedelay2=xd2_pv;

  for(i=0;i<15;i++){
   (final_result[i])=b1[i]+IW1_1_temp[i]*
      tapedelay1_temp+IW1_1_Et[i]*tapedelay1_Et
      +IW1_1_hum[i]*tapedelay1_hum+IW1_2[i]*
      xd2_pv;

  }

transig_apply(final_result_ptr ,out);
   a2=b2;
   for(i=0;i<15;i++){

       a2=a2+LW2_1[i]*result_out[i];



   }
 y1[ts]=mapminmax_reverse(a2,y1_step1_gain ,
    y1_step1_xoffset ,y1_step1_ymin);

    if(y1[ts]<0  ||  y1[ts]>40){
       y1[ts]=0;
    }
 xd1_temp=mapminmax_apply(Temp_new2[horizon+ts
    ],x1_step1_gain_temp ,x1_step1_xoffset_temp ,
    x1_step1_ymin);
```

```
                        xd1_Et=mapminmax_apply(Et_new2[horizon+ts],
                            x1_step1_gain_Et,x1_step1_xoffset_Et,
                            x1_step1_ymin);
                        xd1_hum=mapminmax_apply(Hum_new2[horizon+ts],
                            x1_step1_gain_hum,x1_step1_xoffset_hum,
                            x1_step1_ymin);
                        xd2_pv=a2;

                    }
                    ts=80+(ceil((d))-1)*96-(horizon%96)+1-((ceil(d)
                        )-1)*96;
                    for(i=ts;i<96;i++){

                        if(((horizon+i)%96)<28 || ((horizon+i)%96)
                            >80)
                        {
                            y1[i]=0;
                        }

                    }
                    count=0;
    for(i=0;i<96;i++){

        *(solar+i)=y1[i];

    }
}
```

REFERENCES

[1] CarbonTrack, “Overview of solar cell system design.” https://carbontrack.com.au, 2018.

[2] P. Denholm, M. O’Connell, G. Brinkman, and J. Jorgenson, “Overgeneration from solar energy in california. a field guide to the duck chart,” tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2015.

[3] GreenTechMedia, “3 Trends That Show Why Microgrids Are About More Than Just Resiliency.” https://www.greentechmedia.com/articles/read/3-trends-that-suggest-microgrids-are-for-more-than-just-resiliency#gs.FcG7IwI, 2016.

[4] R. Fu, D. J. Feldman, R. M. Margolis, M. A. Woodhouse, and K. B. Ardani, “Us solar photovoltaic system cost benchmark: Q1 2017,” tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2017.

[5] V. Tolieng, B. Prasirtsak, J. Sitdhipol, N. Thongchul, and S. Tanasupawat, “Identification and lactic acid production of bacteria isolated from soils and tree barks,” *Malaysian Journal of Microbiology*, vol. 13, no. 2, pp. 100–108, 2017.

[6] R. B. Diddigi, D. S. K. Reddy, and S. Bhatnagar, “Multi-Agent Q-Learning for Minimizing Demand-Supply Power Deficit in Microgrids,” 2017.

[7] L. Raju, S. Sankar, and R. S. Milton, “Distributed optimization of solar micro-grid using multi agent reinforcement learning,” in *Procedia Computer Science*, vol. 46, pp. 231–239, 2015.

[8] M. L. Puterman, *Markov Decision Processes*. 1994.

[9] S. Grillo, A. Pievatolo, and E. Tironi, “Optimal Storage Scheduling Using Markov Decision Processes,” *IEEE Transactions on Sustainable Energy*, vol. 7, no. 2, pp. 755–764, 2016.

[10] P. Harsha and M. Dahleh, “Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy,” *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1164–1181, 2015.

[11] Y. Lan, J. Wu, and X. Guan, “Rollout strategies for real-time multi-energy scheduling in microgrid with storage system,” *IET Generation, Transmission & Distribution*, vol. 10, no. 3, pp. 688–696, 2016.

[12] R. Palma-Behnke, C. Benavides, F. Lanas, B. Severino, L. Reyes, J. Llanos, and D. Saez, “A microgrid energy management system based on the rolling horizon strategy,” *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 996–1006, 2013.

[13] G. Gross and F. D. Galiana, "Short-Term Load Forecasting.," *Proceedings of the IEEE*, vol. 75, no. 12, pp. 1558–1573, 1987.

[14] M. T. Hagan and S. M. Behr, "The Time Series Approach to Short Term Load Forecasting," *IEEE Transactions on Power Systems*, vol. 2, no. 3, pp. 785–791, 1987.

[15] M. Espinoza, C. Joye, R. Belmans, and B. De Moor, "Short-term load forecasting, profile identification, and customer segmentation: A methodology based on periodic time series," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1622–1630, 2005.

[16] M. Cho, J. Hwang, and C. Chen, "Customer short term load forecasting by using ARIMA transfer function model," *Proceedings 1995 International Conference on Energy Management and Power Delivery EMPD '95*, vol. 1, no. 95, pp. 317–322, 1995.

[17] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: Literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, 2002.

[18] P. Bacher, H. Madsen, and H. A. Nielsen, "Online short-term solar power forecasting," *Solar Energy*, vol. 83, no. 10, pp. 1772–1783, 2009.

[19] D. Shi, R. Li, R. Shi, and F. Li, "Analysis of the relationship between load profile and weather condition," in *IEEE Power and Energy Society General Meeting*, vol. 2014-Octob, 2014.

[20] S. Pelland, J. Remund, J. Kleissl, T. Oozeki, and K. D. Brabandere, "Photovoltaic and Solar Forecasting: State of the Art," *International Energy Agency: Photovoltaic Power Systems Programme, Report IEA PVPS T14*, pp. 1–40, 2013.

[21] M. Abuella and S. Member, "Solar Power Forecasting Using Support Vector Regression," *Proceedings of the American Society for Engineering Management International Annual Conference*, 2016.

[22] X. Yan, B. Francois, E. C. D. Lille, C. Scientifique, and V. Ascq, "Solar Radiation Forecasting Using Artificial Neural Network for Local Power Reserve," pp. 2–7, 2014.

[23] R. Hledik, "Rediscovering Residential Demand Charges," *Electricity Journal*, vol. 27, no. 7, pp. 82–96, 2014.

[24] Y. Riffonneau, S. Bacha, F. Barruel, and S. Ploix, "Optimal Power Flow Management for Grid Connected PV Systems With Batteries," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 3, pp. 309–320, 2011.

[25] X. Kong, L. Bai, Q. Hu, F. Li, and C. Wang, "Day-ahead optimal scheduling method for grid-connected microgrid based on energy storage control strategy," *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 4, pp. 648–658, 2016.

[26] G. A. Mbamalu and M. E. El-Hawary, "Load Forecasting Via Suboptimal Seasonal Autoregressive Models and Iteratively Reweighted Least Squares Estimation," *IEEE Transactions on Power Systems*, vol. 8, no. 1, pp. 343–348, 1993.

[27] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 798–805, 2001.

[28] J. D. J. Deng and P. Jirutitijaroen, "Short-term load forecasting using time series analysis: A case study for Singapore," *Cybernetics and Intelligent Systems (CIS), 2010 IEEE Conference on*, no. 1, pp. 1–6, 2010.

[29] E. J. McCoy and D. A. Stephens, *Bayesian time series analysis of periodic behaviour and spectral structure*, vol. 20. 2004.

[30] L. Suganthi and A. A. Samuel, "Energy models for demand forecasting - A review," 2012.

[31] K. Bakirci, "Estimation of solar radiation by using ASHRAE clear-sky model in erzurum, Turkey," 2009.

[32] M. Abouhashish, "Applicability of ASHRAE clear-sky model based on solar-radiation measurements in Saudi Arabia," in *AIP Conference Proceedings*, vol. 1850, 2017.

[33] D. Thevenard, S. Cornick, and P. Rp, "Revising ASHRAE Climatic Data for Design and Standards âĂŤ Part 1 : Overview and Data," in *ASHRAE Transactions*, vol. 119, pp. 194–209, 2013.

[34] S. Ibrahim, I. Daut, Y. M. Irwan, M. Irwanto, N. Gomesh, and Z. Farhana, "Linear regression model in estimating solar radiation in perlis," *Energy Procedia*, vol. 18, pp. 1402–1412, 2012.

[35] K. Gairaa and Y. Bakelli, "A Comparative Study of Some Regression Models to Estimate the Global Solar Radiation on a Horizontal Surface from Sunshine Duration and Meteorological Parameters for Ghardaïa Site, Algeria," *ISRN Renewable Energy*, vol. 2013, pp. 1–11, 2013.

[36] H. Suehrcke, R. S. Bowden, and K. G. Hollands, "Relationship between sunshine duration and solar radiation," *Solar Energy*, vol. 92, pp. 160–171, 2013.

[37] A. Alzahrani, P. Shamsi, C. Dagli, and M. Ferdowsi, "Solar Irradiance Forecasting Using Deep Neural Networks," *Procedia Computer Science*, vol. 114, no. 1, pp. 304–313, 2017.

[38] S. Mohanty, P. K. Patra, and S. S. Sahoo, "Prediction of global solar radiation using nonlinear auto regressive network with exogenous inputs (narx)," in *Proceedings of the 2015 39th National Systems Conference, NSC 2015*, 2016.

VITA

Prateek Jain received a Bachelor of Engineering in Electrical Engineering from Thapar University, Patiala, India, in 2015. He briefly worked as an Electronics Design Engineer at GreyOrange, Gurgaon, India. He completed his Master of Science in Electrical Engineering at Missouri University of Science & Technology in Rolla in May, 2018. His research interest included Microgrids, Power Electronics and Battery management systems.