Predictive Energy Management
for
Wireless Sensor Nodes

by

James E. A. Rodway

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

A wireless sensor network is a tool that can collect data, aiding in answering a number of different questions in research and industrial environments. When deployed in remote locations, it is often beneficial to use of energy harvesting technologies, allowing sensor nodes to replenish energy while in the field. This permits longer deployment times while keeping node size small. In order to make the best use of harvested energy, controllers can be used to adapt node activities to available energy. In this thesis, energy forecasts based on measurements of atmospheric pressure are created and included as inputs to fuzzy controllers. These controllers are applied to simulated sensor nodes and used to control node activity levels for effective use of available energy. They were tuned using differential evolution and simulated using measured meteorological data. The results were examined in terms of the networks overall activity level and the usage of reserve energy. With respect to the solar energy forecasts, a number of applied methods were able to achieve error levels comparable to other methods where more variables were included. The tuned fuzzy controllers represented an improvement over both the uncontrolled and human-created cases.

Acknowledgements

The completion of this thesis would not have been possible without the help others.

I wish to thank my supervisor Dr. Petr Musilek, for his support and guidance during my time working on this research project. Tomas Barton provided invaluable data regarding energy usage of sensor nodes, allowing for the simulation of nodes described in this document to be more realistic.

I would also like to thank a number of people for their encouragement and support, including Pawel Pytlak, Justin Fedor, and the members of my family Art, Bobbi, Rose and John. I am particularly grateful to Kelly, whose continued love and patience means so much to me.

Funding provided by the Queen Elizabeth II Graduate Scholarship is also gratefully acknowledged.

Contents

List of Tables

List of Figures

Symbols and Abbreviations

δ	Solar declination
Γ	Day angle
$\hat{\tau}_F$	Estimated transmissivity factor
ω	Hour angle
Φ	Latitude of a point
Ψ	Longitude of a point
τ	Estimate of clear sky atmospheric transmissivity
DHI	Diffuse horizontal irradiance
DNI	Direct normal irradiance
GHI	Global horizontal irradiance
E_L	Lost (unharvested) environmental energy
E_R	Node energy reserve usage
G_{SC}	Maximum solar irradiance before passing through the atmosphere
h	Site elevation in metres
M_D	Number of measurements per day
M_T	Total network measurements
N_A	Node activity level
$t_{\text{solar noon}}$	Solar noon time
t_{sunrise}	Sunrise time
t_{sunset}	Sunset time
DE	Differential evolution
MF	Membership functions
TS	Takagi-Sugeno fuzzy system
WSN	Wireless sensor network
WSU	Washington State University

Chapter 1

Introduction

1.1 Motivation

Wireless sensor networks (WSN) are composed of a number of sensor nodes connected wirelessly. These sensor nodes are generally small and relatively inexpensive, consisting of components including energy storage, sensors, a wireless transceiver, and a microcontroller to control operations. They are deployed in their target environment and used to collect desired measurements. Depending on the attached sensors, such networks can be used to monitor any number of different phenomena. WSN can also collect data with high spatial resolution when deployed densely, which is made possible by the inexpensive nature of individual nodes.

Sensor nodes may be constructed to be very robust, which makes a WSN an attractive solution for a number of industrial and research applications, especially in environmental fields. The high reliability of sensor nodes translates into long deployment times with few maintenance visits, which is ideal for remote locations. In cases where sensor nodes are deployed in remote locations, it is possible that the cost of deploying the network is comparable or greater than the cost of the network itself. This may mean that the number of future maintenance visits are limited. Here, maximizing the time the network is capable of operating without outside intervention reduces the total overall cost of a successful project. However, restrictions on the types of energy storage devices that can be used, as well as the cost of transporting larger, heavier devices to remote locations further complicates design of a successful WSN deployment.

To alleviate this shortcoming, energy harvesting technologies may be used to replenish a node's energy supply in the field. This allows for smaller energy storage devices to be used, which in turn reduces deployment cost associated with transport of larger devices, as well as potentially avoiding technologies inappropriate for the given site. Where energy harvesting is used, it must be managed effectively to capture the most useful data possible while still maintaining reliability. Measurements should be collected with sufficient frequency to capture changes in the variables of interest without large gaps. In order to support this strategic data gathering, a node must also have adequate energy to support the base network operations such as sending and receiving transmissions and maintaining operation during times of reduced energy harvesting opportunities due to reduced energy availability or hardware failure. Advanced knowledge of harvesting opportunities can be used to improve control of energy consumption, but the cost of providing such knowledge to a remotely deployed sensor node must also be considered.

Effective energy management in a wireless sensor network can result in a reduction in overall cost for a project in a number of ways. When energy is effectively managed, the number of maintenance visits required to replace drained energy is reduced. Inclusion of energy harvesting technologies also allows energy to be replenished in the field and thus to use smaller energy storage devices. This can also reduce the cost by lowering the weight that must be transported for deployments in remote locations. Decreasing deployment costs in this way, the economic hurdles faced by some WSN projects are reduced, meaning that more projects may be accessible, larger deployments may be undertaken, or more sensors could be purchased for the nodes.

1.2 Thesis Objectives

Deployment of a wireless sensor network in a remote location is a costly undertaking. Deployment cost can be reduced by including energy harvesting technologies that allow nodes to replenish energy in the field, reducing the size of energy storage devices. However, energy consumption must be managed to make the best use of the available environmental energy. The objective of the work described in this thesis is to support the success of remotely deployed WSN through effective energy management. This is accomplished by:

- devising methods to forecast daily availability of harvestable solar energy; the methods must be simple to allow implementation on limited hardware of typical WSN platforms;
- designing suitable energy management strategy and methodology for its site-specific tuning to maximize field performance of deployed WSN; given the already mentioned hardware restrictions and uncertainty of environmental conditions, these tasks are accomplished using evolutionary fuzzy control;
- validating the developed approaches through a battery of simulations of interconnected wireless sensor nodes in a network, using real environmental data and realistic models of a node hardware.

1.3 Contributions

This thesis develops a forecast method for the prediction of daily solar energy using atmospheric pressure measurements in order to support the energy management of a wireless sensor node. By using atmospheric pressure as the predictor variable, not only is the forecasting greatly simplified, but the potential sensor costs and deployment complexities are similarly reduced. The developed forecast method is simple enough for implementation on limited hardware and suitable for large scale usage.

The forecast models created with this method were integrated with a fuzzy logic controller for the duty cycling of measurement and transmission rates of wireless sensor nodes. Human-created and computationally optimized fuzzy sets were applied to the controller and simulated using measured meteorological data in order to examine the number of measurements taken and the reserve energy. These metrics served as indicators of collected data quality and potential deployment length.

The methodology and individual approaches developed in this thesis represent significant contributions towards the area of environmentally powered wireless sensor networks. Through the applications of intelligent methods, it provides novel solutions to energy prediction and management that improve the performance and lifetime of such networks, especially when deployed in remote locations without energy infrastructure.

1.4 Organization

This thesis is composed of five chapters. Background is presented in Chapter 2 and includes a brief explanation of wireless sensor networks and their applications, as well as a description of some common energy management techniques. The pressure-based solar energy forecast is developed in Chapter 3, while the results of optimizations tuning controller membership functions are presented in Chapter 4. Results of simulations combining developed forecasts with tuned controllers are shown in Chapter 5. Conclusions and future work are discussed in Chapter 6.

Chapter 2

Background

2.1 Wireless Sensor Networks

A WSN is a network of wirelessly connected, specialized sensor platforms called sensor nodes. Individual nodes consist of a number of components for energy storage, wireless communication, data measurement, and data storage. Different types of nodes may exist in a network, e.g. nodes primarily for sensing (termed sensing nodes), nodes primarily for relaying data (termed routers), and nodes for facilitating data exchange with other networks (termed base stations, or sink nodes). Typically, these sensor nodes are small and relatively inexpensive [2]. In order to remain physically small, sensor nodes normally use limited energy sources, as opposed to sources that may be large, heavy, expensive, or potentially damaging to the environment [3].

Sensor networks vary in size from a few nodes to thousands. Each node in the network is connected to one or more other nodes, allowing data to move through the network. Data move from the point of collection through the network until reaching a sink node where it can be retrieved for study. Retrieval can be achieved via manual download, wired transmission, or long range wireless transmission.

Depending on the application, wireless sensor nodes may also make use of energy harvesting, or scavenging, technology. Potential sources of environmental energy include solar radiation, wind, thermal gradients and acoustic noise. Environmental energy can be exploited using technologies including photovoltaic panels, wind turbines and flutter belts [3,4,5,6]. The inclusion of energy harvesting equipment may have a number of benefits over fixed energy sources. For example, with the ability to replenish energy during deployment, smaller batteries can be used. This may also lead to fewer maintenance visits, which can be costly or impossible depending on the remoteness of the site. Longer term deployments are also possible when energy harvesting is included as part the sensor platform, potentially approaching perpetual operation [7,8].

There are two major types of WSN applications: remote monitoring and mobile object location tracking. These types may be further divided into indoor and outdoor applications [2]. Monitoring applications require periodic sampling and transmission of data either at fixed intervals or in response to specific events. Some examples of remote monitoring include environmental and habitat monitoring [7,9,10,11,12], infrastructure monitoring, health monitoring, as well a number of smart grid related applications [13].

Examples of remote locations where WSN may be deployed include arctic locations [6], tropical regions [12], and inside glaciers [14].

There are a number of considerations that must be made for remote monitoring stations [6]:

- Access to the deployment location may be restricted due to time, weather, cost, or any combination of these factors
- Weather conditions may reduce the effectiveness of energy storage devices (e.g., effect of temperature extremes on batteries)
- Weather and other local conditions may reduce the effectiveness of energy harvesting devices (e.g., snow or dust covering solar panels)

Mobile object location tracking, while not the focus of this thesis finds applications in areas such as animal tracking for both agricultural and conservation purposes [9,11], child education [9,11], avalanche and fire rescue support, and support of product manufacturing and supply chain management [11].

2.1.1 Energy Management

Energy management balances deployment time and quality of service. That is, increasing the quality of service may have a negative impact on the deployment duration. Management strategies can seek to achieve this balance using factors like reduced measurement frequency, longer periods of time between transmissions, or lower network throughput [15,16]. The complexity of the management scheme is related to the quality of service required by the application with more critical applications having more stringent requirements on uptime and measurement frequency.

Energy management for wireless sensor networks can be broken down into two major components: management of energy provision and management of energy consumption. Energy provisioning consists of batteries, harvest and transference. The focus of this thesis, the management of energy consumption is broken down into three strategies: data driven, where energy usage is reduced by predicting data instead of measuring it, adapting the duty cycle to current conditions, and mobility based schemes involving mobile relays or sinks [17].

Energy transference involves transferring energy between nodes, or from a special purpose charging unit using methods such as lasers or other electromagnetic waves. However, the requirement for large and potentially expensive charging platforms with current technology may limit the use of energy transference to very specific applications [17].

Three levels can be considered when managing energy within a WSN: the microcontroller level, the node level, and the network level [18]. At the microcontroller level, energy is managed through proper selection of the microcontroller itself, as well as dynamic voltage and frequency scaling. These techniques require hardware created with these features, which means that they must be considered very early on during development and may be difficult to change afterwards [5,19]. Also, selection of appropriate energy storage technologies to avoid unnecessary losses, as well technologies to reduce energy conversion losses (e.g., maximum power point tracking for photovoltaic panels) should be considered in the early design stage.

Node level energy management includes tools such as adaptive sensing rates where measurements are taken considering the amount of energy available for use or scaled based on changing variability of the target variable. The bulk of energy consumption in a sensor node is due to the wireless communication, making the reduction of wireless transmissions and idle listening time an important node level power management techniques [3,20]. By reducing the number of transmissions a node sends to the rest of the network, the energy used by the entire network is reduced. This may take a number of different forms, from very simple schemes where transmissions are simply not sent, to more advanced techniques where redundant information is reduced through prediction of future values [21,22]. Sampling may also be reduced while recognizing the increase in error associated with less frequent measurements [23].

At the network level, energy management can be realized through schemes like communication scheduling and intelligent, energy-aware message routing, all of which reduce the number of required wireless transmissions [15,20,24]. For WSN with enough node density to support it, clustering can be used to improve deployment duration [25,26]. In these network topologies, nodes are clustered into different groups. Each cluster consists of at least one cluster head and a number of non-cluster heads. Cluster heads handle the processing and forwarding of data to the network base station. With this scheme, some of the energy consumption of non-cluster heads is shifted to the cluster head. This makes selection of the cluster head crucial, as energy harvesting opportunities and the power requirements of transmission distance to the base station must be considered. Base stations may also be moved, and a method of determining the optimal position is presented in [27]. Relocation of network base stations can reduce energy use and increase network lifetime by locating the base station such that the power intensity of transmission may be reduced.

Where energy harvesting is used, node level energy management becomes more complex [28]. The ability to replenish energy supplies during deployment can greatly extend the lifetime of a sensor node. For example, a design for a energy harvesting sensor node with both a battery and a supercapacitor used as energy storage was presented in [29]. The study estimated the lifetimes of the designed nodes and demonstrated the ability for a sensor node to remain active for long periods of time.

Energy neutral operation, i.e. operation of a system that may continue indefinitely if consumption does not exceed energy production, is an important concept for the energy management in wireless sensor nodes [30]. Energy neutral operation can take several forms, depending on the model used for energy storage present in the system. For the simplest case, where there is no energy storage available, energy harvested in excess of what is being consumed is lost. More complex cases involve non-ideal storage elements with limited capacity, non-unity round-trip efficiency, and leakage.

Duty cycling is a commonly used method of reducing power usage, owing to the usual support found for sleep modes on various components [5]. Simple static schemes of duty cycling match a fixed duty cycle to the average energy production. While setting a static duty cycle avoids a great deal of complexity, the downsides include missed opportunities for a higher duty cycle when harvested energy exceeds consumption and there is no more storage capacity available. Additionally, a static duty cycle does not provide opportunity to reduce the duty cycle when the harvested energy is less than future consumption. This issue supports the use of dynamic duty cycling to better match incoming energy.

With respect to duty cycle control of an energy harvesting node, three strategies of energy consumption control are defined in [31] as optimal harvesting, optimal consumption and an adaptive strategy. The optimal harvesting strategy aims to keep the amount of reserve energy near constant, while the consumption rate follows the rate of harvested energy. The defined optimal consumption strategy aims to keep the duty cycle constant, with the duty cycle calculated based on an average of the harvestable energy over a long period of time. The adaptive strategy is a variation of the optimal consumption strategy, when duty cycle is kept relatively constant with correction based on updates made to the average harvesting rate. The aim of the energy management strategy is to utilize the harvested energy to its fullest potential without large variances in duty cycle.

Generally, where energy harvesting is used, node activity levels can be informed by the harvesting opportunities. These strategies may vary in complexity based on the information available for decision making. Foreknowledge of harvesting opportunities may be included. For example, adaptive duty cycling for energy harvesting sensor nodes is discussed in [32] with energy prediction performed using an exponentially weighted average. The implemented controller allowed the utilization of 58% more environmental energy compared to the case without harvest awareness. Adaptation of sensor node parameters based on a prediction of future energy is shown in [33]. Parameters informed by incoming energy included the sensing rate and the usage of local memory. A power estimator based on the output of a numerical weather forecast model and integrated into a dynamic power management scheme is discussed in [34]. This scheme affected node operations such as the duration of video transmitted back to the base station.

Two dynamic duty cycle scheduling schemes to balance energy consumption in energy harvesting WSN, primarily with respect to node transmissions, are presented in [35]. The performance improvements sought were a reduction of end-to-end delay and improvement in packet delivery ratios. One of the proposed schemes used only current residual energy, while the other used an estimate of the prospective residual energy in order to increase the duty cycle more aggressively.

Task scheduling for sensor nodes where properties of the energy source are considered is discussed in [36]. The developed algorithms allowed large reductions in battery size when compared to the required size when using Earliest Deadline First scheduling.

Work simulating the energy use of a wireless sensor platform is presented in [37]. In that work, statically controlled energy management methods are compared to a dynamic fuzzy controller that allows the system to adapt to the available energy for harvest. In the presented results, the dynamic controller outperforms static methods tested in both the size of the dataset collected and the lowest number of device failures. Further simulation of fuzzy controls used for adaptive duty cycling is presented in [38] and [39], where the status of a node's energy buffer is used as an input to determine the nodes sensing and transmission rates.

Simulation of a fuzzy controller using the 24 hour moving average of an installed energy buffer and the percentage of data in the data buffer as inputs to determine the measurement and transmission duty cycle of the node is discussed in [12]. The controller was simulated in tropical dry and boreal, forests and allowed the simulated node to match energy consumption to the amount of energy available in its environment.

Information from across a WSN may also be used to manage its energy consumption. Another study discusses a proposed scheme to set each node's sampling rates such that the network performance is maximized for the case where each node is capable of harvesting energy and has a limited energy storage capacity [40]. The desire for the distribution is not only to prevent node's energy depletion, but to also reduce missed harvesting opportunities caused by nodes being completely charged. A framework for a sensor network to learn the energy landscape that occurs due to unequal harvesting opportunities experienced by sensor nodes is presented in [41]. This knowledge is then used to allow task sharing between nodes for more effective energy consumption. The experiments presented showed large improvements in node lifetime.

2.2 Solar Energy and Atmospheric Effects

Photovoltaic (PV) panels are capable of transforming electromagnetic radiation into electrical energy. This power conversion technology has application spanning very small amounts of energy (e.g., small pocket calculators) to large grid-connected installations with capacities of hundreds of megawatts. Solar panels may be constructed using a number of different techniques and materials. Monocrystalline solar cells using single silicon crystals have the highest efficiency rates, cost, and have better performance when compared to their polycrystalline counterparts. The lowest efficiency PV materials are thin-film cells, which are currently the least expensive [42].

Solar irradiance is the amount of solar power striking a given area of the earth. It is measured in the SI units of W/m^2. PV power output is affected by the amount of radiation striking the panels. For outdoor panels, solar irradiance striking the panel determines the amount of energy that can be harvested. This includes direct incoming radiation, diffuse solar radiation, and solar radiation reflected by the earth's surface. Additional considerations for the amount of energy harvested include the position and orientation of the panel, its efficiency and the use of maximum power point tracking, which involves operating the panel at a certain voltage to ensure that the maximum power is harvested. There are also different amounts of power harvestable from different wavelengths of incoming radiation, owing to the band gaps of the different materials used in the construction of the solar cells [43].

The makeup of the atmosphere affects the type and spectrum of incident radiation experienced by a solar panel. Gases and water vapour in the atmosphere may absorb certain wavelengths of incoming light. Incoming light also experiences Rayleigh scattering as it enters the atmosphere. Although this type of scattering affects all wavelengths of light, the shorter wavelengths corresponding to the blue and purple regions of visible light are affected to a much greater degree. Furthermore, aerosols and clouds cause light to experience Mie scattering. This scattering contributes to a reduction of direct incident radiation and an increase in diffuse radiation striking a solar panel, resulting in the diffuse radiation being the greater contribution in certain conditions [44, 45, 46].

Considering the spectral distribution of incoming solar radiation, visible light wavelengths are between 0.4 μm to 0.8 μm. This range includes the greatest amount of relative power. Wavelengths greater than 0.8 μm are termed infrared energy and also make up a large portion. Wavelengths lower than 0.4 μm are ultraviolet rays and make up less of the relative power of the spectrum [45]. The degree of absorption of different incoming wavelengths is affected by the amount of atmosphere that the radiation must pass through, due to absorption by water vapour, oxygen, carbon dioxide, and other atmospheric gases. The effect of absorption and scattering by atmosphere is generally negative as far as the collection of photovoltaic power is concerned.

Light that has been scattered may still strike the Earth's surface, leading to equations for total, or global, irradiance. For example, the global horizontal irradiance *GHI* can be calculated as

$$GHI = DNI + DHI, \tag{2.1}$$

where the *GHI* can be calculated as the sum of direct normal irradiance (*DNI*) and diffuse horizontal irradiance (*DHI*) components [43]. There may also be a reflected portion of solar irradiance referring to the radiation that has struck a reflective surface, such as water or snow, and has been reflected back towards the point of interest, but in the case of surfaces horizontal to the Earth, this is expected to be small. In cases where the surface is tilted, or stands vertically, reflected light may be a larger factor.

The focus in this work is on the *DNI*, rather than *GHI* primarily due to the lack of available measurements of global irradiance. The use of *GHI* measurements necessitates the scenario where the solar panel is placed normal to the surface of the earth. In a more realistic scenario, the following changes would likely be observed:

- the presence of diffuse radiation would have a positive effect on the amount of energy harvested (the degree of which would depend on the type of solar panel used). This effect would be the greatest during overcast days when the diffuse radiation is the dominant factor in Eqn 2.1
- Tilt of the panel modulates the amount of energy harvested at different times of the year resulting in more collection during the winter at the cost of less collection during the summer.

Both of these effects would result in more collection during times of expected low energy harvest and would slightly improve the overall deployment length of the affected nodes.

2.3 Weather Forecasting

Accurate weather forecasts have applications in a number of different areas such as travel and agriculture. With the increasing usage of renewable energy sources, forecasting solar irradiance and wind has become of greater importance. Not surprisingly, forecasts of solar irradiance are of critical importance to PV power plants.

There are a multitude of reasons for a reduction in the amount of sunlight striking a solar panel. These can include dust, vegetation growth, or the presence of clouds. However, the reduction of direct solar radiation due to cloud cover is transient and cannot be mitigated. Clouds are composed of tiny water droplets water that condense from humid atmosphere when it is cooled such that its relative humidity exceeds 100%. Relative humidity is the ratio of water vapour in the air compared to the saturation amount at the current conditions. By extension, this value indicates how much evaporation may occur, with 100% meaning that no evaporation may occur at all [47]. Atmospheric pressure and temperature both affect the amount of water vapour that air can hold.

Static atmospheric pressure is the force caused by the random movement of air molecules in calm winds. The atmospheric pressure at sea level is 101.325 kPa (standard) and it decreases exponentially with increasing altitude. Regions of high pressure are associated with cold temperatures and low humidity. The boundaries between air masses are termed fronts, and can be the focus of low pressure, clouds and precipitation [47].

Pressure is a useful variable for the prediction of storms and cloud cover. Large changes in pressure can indicate significant changes in the weather. Large pressure drops are associated with increasing cloud cover and storms, while increases in pressure are associated with clearing skies and improving weather conditions [47, 48]. Some rules of thumb exist for weather prediction based on pressure tendencies. Ten rules from [49] are presented in Table 2.1.

In order to create the best forecast possible, meteorological values should be measured with care. For example, accurate measurement of air temperature requires that the thermometer be shielded from solar radiation, in order to avoid absorbing this radiation itself. With some instruments, this absorption may cause a measured value of up to 25 degrees higher than actual. The shield used must also allow free circulation while avoiding moisture on the thermometer itself, which depresses the measured value [50].

Accurate measurement of atmospheric pressure is hampered by the presence of gusting wind, which may cause changes in pressure on the order of 2-3 hPa. This effect can be mitigated through the use of a static head. Specific to electronic barometers, the temperature at which the device operates should be constant, or change slowly, and be near the calibration temperature [50].

Fuzzy inference can be applied to weather forecasting. In [51], a number of variables including temperature, humidity, dew point, amount and type of precipitation, pressure, wind, and clouds were used as inputs to fuzzy models to make a number of short term weather forecasts. The outputs of the developed systems included the estimate of the upcoming general weather condition, ranging from stormy to sunny, as well as the upcoming temperatures and dew points.

Table 2.1: Brewer's 10 Special Rules for Barometric Pressure and Weather

Rule	
1	"The barometer is highest of all during a long frost; and generally rises with a north-west wind"
2	"The barometer is lowest of all during a thaw, which follows a long frost and it generally falls with south or east wind"
3	"While the the barometer stands above 30, the air must be very dry, or very cold, or perhaps both – and no rain may be expected"
4	"When the barometer stands very low indeed, there will never be much rain; although a fine day will seldom occur at such times"
5	"In the summer-time (after a long continuance of fair weather) the barometer will fall gradually for 2 or 3 days before rain comes: But if the fall of the mercury is very sudden, a thunder-storm may be expected"
6	"When the sky is cloudless, and seems to promise fair weather – if the barometer is low, the face of the sky will soon be suddenly overcast"
7	"Dark dense clouds will pass over without rain, when the barometer is high; but if the barometer be low, it will often rain without the appearance of clouds"
8	"The higher the barometer, the greater the probability of fair weather"
9	"When the mercury is in a rising state, fine weather is at hand; but, when the mercury is in a sinking state, foul weather is near"
10	"If (in frosty weather) it begins to snow, the barometer generally rises to 30; where it remains, so long as the snow continues to fall: If, after this, the weather clears up, you may expect very severe cold"

2.3.1 Solar Energy Forecasting with Limited Information

Many models for estimating daily solar radiation are available, many of which focus on the use of commonly measured meteorological values with the aim of supporting agricultural and renewable energy models. Daily solar energy will often be a required input but will not have been measured due to the high cost of the instrumentation [52, 53, 54]. However, more common meteorological observations may be available and can be used to estimate solar energy. These indirect methods most often use the difference between observed maximum and minimum daily temperatures, with some models including other variables (e.g., minimum relative humidity) in order to improve the estimation [55,56,57,58,59]. The models require regression in order to tailor the formula constants to the specific site. Machine learning has also been applied to this problem. For example, support vector machines (SVM) have been applied to monthly estimates of solar energy in [52]. As most of these methods aim to use a relationship between an unmeasured daily solar irradiance and commonly observed meteorological variables, their power as forecasting tools extends only as far as previous values of solar energy correspond to current and future values.

Global solar radiation is estimated from common meteorological data in [58]. In the proposed multiple regression model, extraterrestrial solar radiation, saturation vapour pressures, rainfall data and daily minimums of relative humidity were used as predictors. The reported *RMSE* and *MAPE* values are 2.378 MJ/(m^2 day) and 19.3%, respectively.

Similarly, commonly measured meteorological values are used for the prediction of solar energy also in [59]. The values considered included average daily global solar radiation on horizontal surfaces, average daily relative humidity, average daily amount of rainfall, as well as minimum, maximum and averages of daily temperatures. They report errors of 3.62% *MAPE* and 0.257 kWh/(m^2 day) using the Angstrom-Page model. The two models based on relative humidity are the worst and have *MAPE* values of 7.72% and 8.25%.

Support vector machines are used to predict monthly solar radiation values using the inputs of minimum and maximum temperatures in [52]. They found that a polynomial kernel function outperformed both SVMs using other kernel functions, as well as methods using empirical relationships. The lowest *RMSE* value in that work is 0.833 MJ/(m^2 month).

An investigation of the estimation of solar transmissivity using observations of minimum and maximum temperatures and of total precipitation for western Canada was undertaken in [55]. The model includes four empirical coefficients that varied with the time of the year. This model performed poorly during late fall and winter, but performed better during the growing season. The *RMSE* values for the transmissivity estimates range from 0.107 in July to 0.157 in November.

Support vector machines and radial basis function kernels are used for the estimation of solar intensity three hours into the future with the goal of assisting the integration of solar power in [60]. The meteorological values used as inputs included day, temperature, dew point, wind speed, sky cover, precipitation and humidity. The lowest *RMSE* value reported in this study is 128 W/m^2.

Artificial neural networks are used to provide a forecast of power output of a photovoltaic power installation in [52]. The forecast, with a 24 hour forecast horizon, is created using average values of solar irradiance, humidity, wind speed and power production.

In [61], 24 hourly pressure measurements, as well as the differences between all of them are used as input features to the support vector machines. The best reported errors for the Fairview 2012 training data are 12.9% with respect to *MAPE* and 2.803 MJ/(m^2 day) with respect to *RMSE* using a radial basis function kernel. For the 2013 Fairview test data, the best errors are 26.7% with respect to *MAPE* and 3.160 MJ/(m^2 day) with respect to *RMSE* and used a polynomial kernel.

Fuzzy evolutionary rules are used to estimate the next day's incoming solar energy in [62]. In this work, 24 hourly pressure measurements and the differences between them are used as inputs. Three different fitness functions are used during the creation of the fuzzy rule sets. With respect to the training set, the lowest *MAPE* value is 18.80%, while the lowest *RMSE* value is 2.310 MJ/(m^2 day). For the test set, the lowest *MAPE* value is reported as 23.4%.

Support vector machines are used for short-term forecasting of PV power in [63]. In that study, features are created from time series of solar irradiances in order to create a weather classification. The features extracted include a clearness index, the root mean square deviation between the measured and analytical solar irradiances, the maximum value of the third derivative of the difference between measured and analytical solar irradiances, the ratio of maximum measured and analytical solar irradiances, the variance of the difference between solar irradiances, and an inconsistency coefficient. The created models result in high overall accuracy for the classification of a days weather.

A review of both empirical models and soft computing techniques, used to estimate solar energy is presented in [64]. Empirical equations using a number of meteorological values including clearness index, average daily temperature, ratio of minimum and maximum daily temperatures, relative humidity and relative sunshine duration resulted in *RMSE* values of the estimate ratio ranging between 0.1705 and 0.06996 MJ/(m^2 day) for a location in Nigeria. At another site in Nigeria, maximum and minimum daily temperature values are used and result in *RMSE* values between 1.59 MJ/(m^2 day) for a model with regressed coefficients and 4.55 MJ/(m^2 /day) without. This review cites a number of advantages and disadvantages of the methods examined. With respect to the soft computing methods, the main advantages are the ability to detect complex nonlinear relationships and to remain tolerant to input noise. The disadvantages include the greater computational cost of creating them and the lack of transparency with the resulting model.

Prediction of daily global solar radiation using TS fuzzy systems is presented in [65]. The proposed method does not require any transformation of the data and uses two fuzzy inputs, each with 4 Gaussian membership functions. The variables used as inputs are the preceding two measurements of daily global solar radiation. The training and test *RMSE* values reported for the presented model are 0.76 and 0.86 Kw/m^2, respectively.

Estimation of daily solar radiation using regularly measured meteorological values is presented in [56]. The variables included as predictors were daily minimum and maximum temperatures, daily average dew point temperature, fog and precipitation amounts, depending on the model being tested. All models involved the analytical estimate of the above-atmosphere solar radiation and multiplication with an algebraic formula involving the different features and empirically determined constants. Reported errors include the *RMSE* for the overall model at 2.522 MJ/m^2/day.

Further models were created for specific seasons, with the winter and fall models producing *RMSE* values of 1.491 and 2.037 MJ/m^2/day. The best model created for the summer has an *RMSE* value of 3.163 MJ/(m^2 day), while the best *RMSE* value reported for a spring model is 2.910 MJ/(m^2 day).

A calibrated Hargreaves-Samani equation, which uses minimum and maximum daily temperatures, as well the average atmospheric pressure at the site is presented in [57]. The reported standard estimates of error for this method ranged between 2.8 and 3.28 depending on the station considered.

Different types of adaptive neuro-fuzzy inference systems (ANFIS) and M5Tree models for the prediction of daily global solar radiation are discussed in [54]. Variables included for the prediction were sunshine durations, air pressures, minimum and maximum temperatures, average temperatures, water vapour pressure, and relative humidity. The reported *RMSE* values for the ANFIS based models range between 2.07 and 3.08 MJ/m^2/day while the M5tree model produced *RMSE* values between 2.79 and 3.87 MJ/m^2/day. These error values represented an improvement over the empirically calibrated Angstrom model.

With a more explicit goal of solar energy forecasting, a self-organizing map is used to classify the type of weather expected for the next 24 hours for the support of a photovoltaic power installation in [66]. In that work, a radial basis function network is used to directly estimate the power generation of a rooftop solar panel installation for 24 hours. The input values for this study were the daily means of solar irradiance, air temperature, relative humidity, pressure, and windspeed. The reported *MAPE* values for these prediction range between 0.0636 and 0.5444, depending on the classification of the weather, with cloudy weather having the lowest error, and rainy having the greatest.

A number of different solar radiation models using various meteorological variables is presented in [53]. The variables included air temperature, atmospheric pressure, relative humidity, vapor pressure and and sunshine duration. A number of different neural networks, including multilayer perceptron networks (MLP), generalized regression networks and radial basis networks (RBNN), were applied to data from stations from differing climates. The MLP and RBNN methods provided the better results, but these results varied between stations. The *RMSE* reported for MLP range between 1.94 and 3.27 MJ/m^2/day. Overall, considering all the models and stations investigated, the *RMSE* values for these predictions vary between 1.94 and 4.58 MJ/m^2/day.

Historical weather information, including the month, maximum temperature, probability of precipitation and a description of weather are used as inputs into a fuzzy system that selects a support vector regression model to use in order to predict hourly outputs of solar power production 1 day ahead in [67]. The proposed method reports *RMSE* values that are generally lower than 500 W, with the average error being 350.2 W.

A method of estimating the the amount of incoming solar energy specifically for wireless sensor nodes is presented in [68]. In this case, a variant of an exponentially weighted moving average is introduced. The new method includes solar conditions and also adjusts the estimation of available solar energy throughout the day. This method represents an improvement over regular exponentially weighted moving average, which has incidences of high error when sunny and cloudy days are intermingled. Their method includes seasonal variations in day length, as well as difference in solar power between seasons. The reported *MAPE* of the proposed prediction method is between 8% and 15% depending on the weather conditions, with the former being associated with consistent weather conditions and the latter being associated with alternating weather conditions.

A 'weather-conditioned moving average' is used to predict the amount of incoming solar energy [69]. Combined with an energy management algorithm, it greatly improved the energy utilization when compared to a regular exponentially weighted moving average based prediction. The weather conditioning factor introduced into the moving average was based on samples of previously collected solar energies. The reported *MAPE* values for these predictions was 9.80%.

2.3.2 Estimating Daily Direct Clear-Sky Solar Irradiance

In order to support a forecast of daily solar energy, an analytical estimate of the direct solar irradiance striking a surface can be made. This estimate requires the location of the site and the day of the year. While complex and more accurate forecasting methods are available, simple approaches also exist and provide estimates at lower computational cost. The method used here involves only three static values specific to the site, namely latitude, longitude and elevation. The daily estimates of incoming solar energy need to be precalculated only once [43, 46, 70, 71, 72]. First, the extraterrestrial solar radiation striking the Earth at the edge of the atmosphere varies over a year, and can be estimated from:

$$G = G_{sc}(1 + 0.033\cos(2\pi\frac{N}{365}), \tag{2.2}$$

where N is the day of the year and G_{sc} is the solar irradiance reaching the Earth at the edge of atmosphere.

The solar declination in radians is given by the Spencer formula:

$$\begin{aligned}\delta = {} & 0.006918 - 0.399912\cos\Gamma + 0.070257\sin\Gamma \\ & - 0.006758\cos 2\Gamma + 0.000907\sin 2\Gamma \\ & - 0.002697\cos 3\Gamma + 0.00148\sin 3\Gamma,\end{aligned} \tag{2.3}$$

where Γ is the day angle given by:

$$\Gamma_N = 2\pi\frac{N-1}{365}. \tag{2.4}$$

The sunset angle corresponds to the angle of the Earth's rotation where the sun dips below the horizon at a particular point. Using the solar declination δ and the site latitude Φ, the sunset angle can then be calculated:

$$\omega_{\text{sunset}} = \arccos(-\tan\Phi\tan\delta). \tag{2.5}$$

The hour angle is calculated as:

$$\omega = \frac{15^\circ}{\text{hour}}(t_{zone} - 12h) + \omega_{eq} + (\Psi - \Psi_{zone}), \tag{2.6}$$

where Ψ is the longitude of the site and Ψ_{zone} is the longitude of the local time meridian. The equation of time ω_{eq} is the difference between solar and local time and is defined as:

$$\omega_{eq} = 9.87\sin(2B) - 7.53\cos(B) - 1.5\sin(B), \tag{2.7}$$

where B is calculated as:

$$B = (N - 81)\frac{360}{364}. \tag{2.8}$$

Solar noon is defined as the local time when the sun is directly overhead and can be calculated as:

$$t_{\text{solar noon}} = 12 - (4(\Psi_{zone} - \Psi) + \omega_{eq}). \tag{2.9}$$

This sunset angle can then be converted into the time at which sunset occurs using:

$$t_{\text{sunset}} = t_{\text{solar noon}} + \frac{\omega_{\text{sunset}}(\pi/180)}{15}, \tag{2.10}$$

and using symmetry, sunrise time can be similarly calculated as:

$$t_{\text{sunrise}} = t_{\text{solar noon}} - \frac{\omega_{\text{sunset}}(\pi/180)}{15}. \tag{2.11}$$

The solar altitude angle α is calculated as:

$$\sin(\alpha) = \sin\Psi\sin\delta + cos\Psi\cos\delta\cos\omega, \tag{2.12}$$

which uses site longitude as well as the previously calculated values for declination and hour angle. The solar altitude angle is then integrated between the sunrise and sunset angles:

$$\begin{aligned}\int_{t_{\text{sunrise}}}^{t_{\text{sunset}}} \sin\alpha \mathrm{d}t &= \int_{\omega_{\text{sunrise}}}^{\omega_{\text{sunset}}} \frac{12}{\pi} \sin\alpha \mathrm{d}\omega \\ &= \frac{24}{\pi}(\cos\Phi\cos\delta\sin\Psi + \Psi\sin\Phi\sin\delta),\end{aligned} \tag{2.13}$$

which can be used to estimate the value of incoming daily solar energy. To obtain the final value of the estimate, a simple model of clear sky atmospheric transmissivity is added in order to take into account some of the absorption and scattering in the atmosphere [73]:

$$\tau = 0.75 + 0.00002h, \tag{2.14}$$

where h is the site elevation in metres. This model was developed through a linearization of Beer's Law (see [47]) with respect to elevation and is valid for site elevations of less than 6000 m with relatively clean air. It ignores some of the more complex factors including water vapour and atmospheric contaminants, making it an ideal estimate for cases where this additional information is not available and the assumptions are reasonable. A site specific model can be used instead, but would require calibration that may not be possible in practice. A different approximation for net sky transmissivity estimate is presented in [47] as:

$$\tau = (0.6 + 0.2\sin\theta)(1 - 0.4\sigma_H)(1 - 0.7\sigma_M)(1 - 0.4\sigma_L), \tag{2.15}$$

where σ_H, σ_M, and σ_L are cloud cover fractions for high, medium and low clouds. With cloud cover fractions varying between 0 and 1, this model suggests the degree to which the transmissivity is affected by the amount of clouds at varying altitudes.

The values obtained from Eqns. 2.2, 2.13, and 2.14 are then combined to produce an estimate of available daily solar energy, $\hat{E}_A$ in J/m^2:

$$\hat{E}_A = 3600\tau G \int_{\omega_{\text{sunrise}}}^{\omega_{\text{sunset}}} \sin\alpha. \tag{2.16}$$

This estimate of clear-sky daily solar energy depends only on the day of the year, and the elevation and location of the site. The lack of dependence on meteorological factors that affect the transmissivity make it attractive. It can be pre-calculated and stored as a lookup table on an embedded device.

For the case where the surface is tilted, there are additional factors in the estimation of total daily solar energy. For a flat solar collector tilted at an angle of β from the horizontal and turned γ from an axis passing through the poles, the angle of incidence θ may be calculated as:

$$\begin{aligned}\cos\theta = {} & \sin\delta\sin\Phi\cos\beta - \sin\delta\cos\Phi\sin\beta\cos\gamma \\ & + \cos\delta\cos\Phi\cos\beta\cos\omega + \cos\delta\sin\Phi\sin\beta\cos\gamma\cos\omega \\ & + \cos\delta\sin\beta\sin\gamma\sin\omega.\end{aligned} \tag{2.17}$$

This can again be integrated between the sunrise and sunset angles to estimate the solar energy:

$$\begin{aligned}\int_{t_{\text{sunrise}}}^{t_{\text{sunset}}} \cos\theta \mathrm{d}t = {} & \int_{\omega_{\text{sunrise}}}^{\omega_{\text{sunset}}} \frac{12}{\pi}\cos\theta \mathrm{d}\omega \\ & = \frac{24}{\pi}(\omega_{\text{sunset}}\sin\delta\sin\Phi\cos\beta \\ & - \omega_{\text{sunset}}\sin\delta\cos\Phi\sin\beta\cos\gamma \\ & + \cos\delta\cos\Phi\cos\beta\sin\omega_{\text{sunset}} \\ & + \cos\delta\sin\Phi\sin\beta\cos\gamma\sin\omega_{\text{sunset}}).\end{aligned} \tag{2.18}$$

With a tilted surface, extra care must be taken with the limits of integration. There may be cases where the sunset angle for the tilted surface may be larger than the angle obtained when calculated for a horizontal surface. For tilted surfaces, there may also be the case where the sun is behind the surface. Global solar irradiance can also be estimated, but these models are more numerous and slightly more complex [74]. Additionally, more complex models of direct solar irradiance are available, but also require many values that are difficult to measure [75].

2.4 Error measures

In order to compare the performance of different estimations, a number of different error measures may be used. While each have flaws, the more common measures have been used in this work in order to enable comparison with other works [76,77].

Mean absolute percentage error, *MAPE*, was used to evaluate the performance of the predictions:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - y_i|}{y_i} \times 100\%, \tag{2.19}$$

where n is the number of predictions, y_i is the true value (or more accurate value), and x_i is the predicted value. This error metric is easily understandable, but has downsides of not producing values when y_i is zero and is not symmetric, with higher values occurring for over-prediction than for under-prediction [78].

Root mean squared error, *RMSE*, is also calculated:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{2.20}$$

RMSE penalizes larger values of error more heavily, and higher values indicate potentially large errors. Mean absolute error, *MAE*, does not penalize the size of the error more heavily and can be calculated simply as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i|. \tag{2.21}$$

Skill scores can also be useful for comparing the improvement of a method with respect to some reference value. It may be used with any error metric and is calculated as:

$$SS = 1 - \frac{M_{\text{forecast}}}{M_{\text{reference}}}, \tag{2.22}$$

where M_{forecast} represents the value of the error metric that was produced from the forecast and $M_{\text{reference}}$ represents the value of the same error metric as produced by the reference forecast. The skill score represents the relative performance of a forecast with respect to a reference forecast. Scores closer to one represent a larger improvement, while numbers close to zero represent little or no improvement. Negative values represent a forecast that is worse than the reference with respect to the error metric M.

2.5 Computational Techniques

This section outlines the computational techniques used in this work. Regression techniques are used to relate changing values of atmospheric pressure to the amount of incoming solar radiation. The use of this single variable for solar energy forecasting presents some difficulty as it does not fully capture the impact of cloud cover and its development. Nonlinear regression methods have been chosen to find acceptable relationships for this purpose.

Fuzzy logic is used as the method of controlling node activities. This method of control is desirable for this application because it can provide control without a concrete mathematical model of the node being controlled [79]. Energy usages of individual nodes are changed by the behaviour of other nodes in the network and by differences in harvesting opportunities, which provides a need for control flexible enough to handle these uncertainties.

Differential evolution is introduced as a method of optimization, with the aim of using it to tune the membership functions of the fuzzy logic controller. There are two reasons why a controller may see benefit from tuning membership functions in the case of predictive energy management. The first is that the membership functions may be better matched to the prevalent profiles of incoming solar energies. The second is for cases where the forecast horizon is increased. In these cases the relationship between the forecasts further in the future and the currently selected duty cycle become far more difficult to divine (motivating the use of the membership function tuning).

2.5.1 CART, Random Forests, Neural Networks

A number of different methods are available for obtaining an estimate of a target value based on a number of explanatory variables. The focus here was on methods that would be capable of running with the limited computing resources present in a wireless sensor node. The ability for nonlinear relationships between the inputs and outputs is also desirable as it is not expected to be a linear relationship.

Classification and regression trees were originally developed by Brieman et al. in 1984 [80, 81]. This method is capable of dealing with nonlinear relationships and high order interactions while still remaining easily interpretable [82]. Using this regression method, the target variable is repeatedly split into smaller and smaller groups based on one of the explanatory variables. Splits are performed such that the members of the resulting groups are as similar as possible. The trees are grown to be very large, then pruned back such that the smaller tree has the lowest cross-validation estimate of error. Since values of the explanatory variables are only compared with values of the same variable, normalization is not required. The regression tree implementation used is taken from the R *tree* library.

After their creation, simple regression trees are a set of branching comparisons with fixed values. Combined with their lack of requirement for normalized data, simple implementation and low computational requirements make them an attractive option for use on limited hardware.

A random forest consists of a number of regression trees, where random subsets of variables are considered at each split [83, 84]. The results of all created trees are averaged to create the final output. This ensemble prediction results in lower errors and an increased tolerance for noise. Additionally, internal estimates of variable importance are made during the forest creation process. As with single regression trees, since variables are never compared to one another, their normalization is not required. The random forest implementation used is taken from the R *randomForest* library.

Neural network (NN) methods considered in this work include a multilayer perceptron (MLP) neural network, an extreme learning machine (ELM) trained MLP NN, and Elman and Jordan recurrent NN. The recurrent neural networks make use of previous information in their prediction, which may improve predictive accuracy.

An MLP network consists of at least 3 layers of nodes: the input layer, a number of layers consisting of hidden nodes, and the output layer. Input values are multiplied by a weight value and added to a bias value before being evaluated with an activation function for each neuron. The activated values are then passed to the next layer of neurons, again being multiplied by a weight value, added to a bias and used to evaluate an activation function, until reaching the output layer [85].

The ELM network has the same structure as the MLP network but uses a different training method [86]. Using this method, there is no iterative network training. Weights of the hidden nodes are assigned randomly and the weights for the output are solved for using a matrix pseudo-inverse.

With both the Jordan and Elman networks, a number of context nodes are added to the input layer and receive values from previous network evaluations. With respect to Jordan networks, the past output is passed into the current evaluation, while the Elman network passes the outputs of each hidden node to its context inputs.

2.5.2 Fuzzy Logic and Control

Fuzzy logic has several benefits compared to binary logic, such as the ability to handle imprecise, incomplete, vague and uncertain information. It has been applied to many different control problems, including control of home appliances and traffic signals [85].

Fuzzy control systems are based on a set of if-then rules where controller inputs determine the outputs. Fuzzy controllers can be useful in cases where there is no available mathematical model of the system being controlled, experienced human operators can provide qualitative control rules, or uncertainty and variation exists in parameters [79]. Non-fuzzy, or crisp, inputs are first fuzzified and then the rule base is applied. The resulting fuzzy outputs are then defuzzified in order to obtain a crisp output that may be used by the controlled process.

Takagi-Sugeno (TS) fuzzy models are useful for control of nonlinear systems [87]. With this model, a fuzzy rule represents a local function input/output relationship of the system. The fuzzy model of the system is then achieved through blending of these individual relationships [79, 88]. The functions may be constants, not depending on the inputs at all (0^{th} order), linear (1^{st} order), etc., or other functions entirely.

With respect to defuzzification, a TS fuzzy system may have less intensive computation requirements when compared to a method like centroid. Additionally, since the controller outputs are functions of the input variables, much more dynamic control may be realized when compared to cases where controller outputs are static values.

Levels of activation of individual fuzzy sets are calculated using a possibility measure [89]:

$$\mathrm{Poss}(X(x), A_i) = \sup_{x \in X}(\min(X(x), A_i(x))), \tag{2.23}$$

where $X(x)$ is a fuzzy singleton corresponding to the value of X, and A_i is one of the fuzzy membership functions of set A defined on the universe of discourse X. For a two input case where the input fuzzy sets are termed X and Y and an output set of functions O, the activity of a rule r from the fuzzy rule base of the form IF X IS A_i AND Y IS B_i THEN O IS $O_i(\bullet)$ is calculated using an algebraic-product **t**-norm operation:

$$\lambda_r = \mathrm{Poss}(X(x), A_i)\mathbf{t}\mathrm{Poss}(Y(y), B_i), \tag{2.24}$$

where λ_r is the activation of the rule. The final output of the controller is calculated as the average of the output functions weighted by their activations:

$$C = \frac{\sum \lambda_r O_i(\bullet)}{\sum \lambda_r}, \tag{2.25}$$

where $O_i(\bullet)$ is a function of the inputs from O associated with rule r from the rule base and C is the controller output.

In this work, a TS type fuzzy controller was used rather than a Mamdani controller. Studies have shown that these different controller types offer similar performance (e.g. [90]). The process of creating the output fuzzy set used in Mamdani has a higher computational burden when compared to the weighting of output functions used in TS. This makes TS a more attractive option for use on the limited hardware of sensor nodes. One study comparing the two different types of fuzzy logic controllers noted that the processing time using a TS controller was always lower than that for the Mamdani controller [91]. Ideally, the lower computational burden translates to faster processing time and lower energy usage. Both these qualities are attractive for application in wireless sensor networks.

2.5.2.1 Constraints on Fuzzy Sets

In order to create robust and transparent fuzzy controllers, a few constraints should be placed on the fuzzy sets. Two possible constraints are coverage and distinguishability [89]. Ensuring proper coverage guarantees that at least one fuzzy membership function will be activated for every point in the input space. Without proper coverage of the input spaces, there may be cases where the controller fails. This also implies that there must be at least a small degree of overlap between membership functions. Distinguishability is associated to the semantic soundness of a fuzzy set, which relates to the set having a meaningful linguistic interpretation. This can be realized by having fuzzy sets where membership functions are unimodal and sufficiently disjoint, as well as not too great in number [92]. If two input fuzzy membership functions overlap a large amount, then they may be too similar to have a meaningful difference in the output space.

For the fuzzy sets discussed later in this thesis, the guidelines for semantic soundness are fulfilled in two ways. For the guidelines of a limited number of unimodal membership functions, these are ensured via the selection of fuzzy membership functions, where only triangular, trapezoidal, and Gaussian have been considered, and by the limited number of membership functions allowed. For the case where evolutionary computing is used to tune the fuzzy membership functions, a penalty is applied to the resulting fuzzy sets based on a measure of the distinguishability of membership functions using the possibility function from Eqn. 2.23.

2.5.3 Optimization of Fuzzy Controllers

One of the strengths of fuzzy control is the ability to create a usable controller from linguistic if-then rules. It may not always be the case that the object of control is straightforward enough for a human operator to create the necessary rules. However, the other benefits of fuzzy control are still desired. In this case, one method of obtaining a fuzzy controller may be through optimization, using a number of different methods, e.g., genetic algorithms (GA), differential evolution (DE), covariance matrix adaptation evolutionary strategy (CMAES), particle swarm optimization (PSO), or bee colony optimization (BCO). In the optimization of a fuzzy controller, numerous components or combination of components may be operated on, including input membership functions, output membership functions, the fuzzy rule base, and/or controller gain parameters. Where parameters of individual membership functions are operated on during optimization, additional constraints can be placed on the solutions with respect to coverage and distinguishability, but this is not strictly necessary.

Optimization of fuzzy membership functions using genetic algorithms is presented in [93]. The membership functions considered for optimization in this contribution were triangular and the parameters associated with each membership function were simply related to the center and spread of the triangle. Additional factors were introduced to the fitness of a potential solution relating to the coverage and overlap of the membership functions in order to produce sets with functions that cover the universe of discourse without being too redundant. The method of optimizing fuzzy membership functions for controllers was applied to a general industrial process. Two inputs and one output were used, each partitioned with 5 triangular membership functions. The resulting controller showed a small improvement when compared to the human-created reference controller.

A GA optimization of a fuzzy controller that uses a forecast of expected power and an estimate of the simulated battery charge state for a sensor node is presented in [94]. Five discrete values are used for the controller output, which was used to control the duty cycle of the simulated station. The resulting controller successfully maintains high duty cycle while reducing unnecessary changes in state.

Fuzzy control optimization using DE is demonstrated in [95]. The fuzzy controller is applied to the liquid level in a tank as part of a two-tank system. In this optimization, the input and output fuzzy sets were static and differential evolution was used to assign the rules. The resulting controller outperformed the reference controller, having lower response times and percentage overshoots.

Utilization of different optimization techniques, including DE, CMAES, and PSO, to optimize output singletons of a TS fuzzy controller for control of sensor nodes is explored in [1]. For this controller, the input partitions were fixed (Figs. 4.8). In that study, DE was found to produce the best solutions. CMAES may have difficulty because the output associated with a given input could have a far greater effect on the output because of recurring weather conditions.

The use of DE for optimization of a cascade fuzzy controller for inverted pendulum and ball and beam problems is presented in [96]. In these cases, the fuzzy input sets were made up of static triangular membership functions, while the outputs were fuzzy singletons obtained through the use of PSO. DE and GA were applied to the various gains in the system to create the final optimized controller. For the inverted pendulum problem, the controller created with DE had larger percentage overshoot, but faster overall response time, while for the ball and beam problem, the DE optimized controller had improved overshoot, delay, rise and settling times.

The application of a number of optimization methods for antiviral therapy for hepatitis is explored in [97]. In this study, the CMAES method produced the best performing controller.

PSO optimization of a TS fuzzy controller for maximum power point tracking in PV systems is shown in [98]. In the optimized controller, 5 membership functions are used for each of the two inputs and the outputs were the linear combinations of the outputs. The produced fuzzy controller outperformed a fuzzy logic controller created in the conventional way. A similar optimization was undertaken in [99] for the control of DC motor speed and resulted in similar improvements.

A modified BCO of a fuzzy controller is presented in [100]. In the proposed extension to BCO, two of the main control parameters are dynamically changed during optimization using a fuzzy inference system, where the inputs represent the current percentage of the maximum iterations and a measure of the diversity of the bees. The proposed method performed well with respect to the traditional variant of BCO when applied to a benchmark problems, including a water tank level control problem, autonomous vehicle control and temperature control. For these problems different numbers and types of membership functions were used. The fitness of the solutions was solely based on performance and did not include measures of overlap or coverage of the membership functions. The resulting controllers created with the BCO extension generally had better performance when compared to the basic version.

2.5.4 Differential Evolution

Differential evolution (DE) belongs to a family of algorithms known as evolutionary algorithms. The main features of these algorithms include the encoding of potential solutions, a function to determine the fitness of candidate solutions, a method of creating an initial population, a method of selecting potential solutions to create new ones, and methods of generating new solutions from previous ones [85].

Differential evolution is a fast and simple method of global optimization [85,101]. This method is attractive because of its performance, low number of control parameters, and low memory requirements [102]. As in other evolutionary algorithms, candidate solutions are created as a vector of randomly generated real numbers within the allowed interval for the variable. The initial solutions are passed to the fitness function and evaluated. There are different methods of creating a new potential solution using differential evolution, one of which is called the *DE/rand/1/bin* scheme. In this scheme, for each of the P members of the population v_o, 3 other potential solutions are selected for the creation of a new population member. The new vector v_n is created as follows:

$$v_n = v_1 + F(v_2 - v_3), \tag{2.26}$$

where v_1, v_2, v_3 are three randomly selected vectors such that $v_o \neq v_1 \neq v_2 \neq v_3$, and F is a scaling value. In this case, v_1 is called the base vector, while the subtraction of v_2 and v_3 is called the difference vector. Each individual value in the original vector is then replaced with the corresponding value of v_n such that:

$$v_n[i] = \begin{cases} v_n[i] & \text{if } rand(0,1) < C \text{ or } i = I \\ v_o[i] & \text{otherwise,} \end{cases} \tag{2.27}$$

where C is the crossover probability, I is a randomly selected index of the potential solution, and $rand(0,1)$ is a uniformly distributed random number between 0 and 1. Fitness is calculated for the newly created population member v_n, and if better than that of v_o, then v_n replaces v_o in the population. This procedure is repeated until predetermined stopping conditions are satisfied.

The label of *DE/rand/1/bin* signifies that this is differential evolution, that the base solution is randomly selected, that there is one difference vector, and that crossover occurs with a binomial probability distribution. Other methods of creating candidate solutions are possible, changing which vectors and how many are used. A few variants are shown in Table 2.2. In this table, v_{best} refers to the current candidate solution with the best fitness value, and $v_1 \neq v_2 \neq v_3 \neq v_4 \neq v_5$ are randomly selected solutions from the population.

Table 2.2: Differential Evolution Mutation Operators

Method Name	Formula
DE/best/1/bin	$v_n = v_{\text{best}} + F(v_1 - v_2)$
DE/current-to-best/2/bin	$v_n = v_o + F(v_{\text{best}} - v_o) + F(v_1 - v_2)$
DE/best/2/bin	$v_n = v_{\text{best}} + F(v_1 - v_2) + F(v_3 - v_4)$
DE/rand/2/bin	$v_n = v_1 + F(v_2 - v_3) + F(v_4 - v_5)$

Variations on the basic DE algorithm exist, including a number of more adaptive versions. The dependence of the basic DE algorithm on the selection of the different parameters for different problems motivates the addition of a degree of self-adaptation [103]. In one adaptive version of DE, termed WDE, all parameters are changed during execution, including the method used to create new candidate solutions [104,105]. Using this method, C and F values are randomly generated and assigned to each member of the population. Additionally, the method used to create new candidate solutions is also randomly assigned to each population member, initially with equal probability. During the course of optimization, the number of times that a new candidate solution replaces its prior parent solution is recorded, as well as the number of times each solution generation method is used to create an improved candidate solution. At a set frequency, the probabilities of each solution generation method are recalculated based on the number of successful replacements of each method. Simultaneously, the probability of an individual creating an improved solution is calculated. For cases where this value is less than the average, new values of *C* and *F* are randomly generated and assigned.

At a longer interval, the population is shrunk by a given percentage ρ, removing the bottom performing solutions. The population is not shrunk past the size of *P*/2 in order to preserve some diversity. Finally, at the longest interval, the fitness value of the best performer is compared to the best performer that existed at the beginning of the interval. If the fitness value has not improved by a selected threshold, the population is reinitialized to its initial size, only keeping the top ρ% performers.

With respect to population sizes for DE, an early guideline was to set the number of population to be 10 times the number of parameters being optimized [106]. However, for a large number of parameters, this may lead to enormous populations. Studies on the effect of population sizes on the performance of the DE algorithm point to the interaction between population size and the other parameters affecting the speed of convergence and the ability of the algorithm to avoid stagnation. In non-adaptive versions, the size of the population is selected based on the number of parameters for a solution and the nature of the problem with separable, unimodal problems requiring the smallest populations. Suggested population sizes range between 2 and 40 times the number of parameters [107]. However, populations that are either too large or too small can have a negative impact on the performance of DE, and the use of a population size smaller than the number of dimensions causes greater dependence on the *C* and *F* parameters [108].

An empirical study of the application of DE to high dimension problems suggests that the best results for multimodal non-separable problems may be obtained using the DE variants of *DE/best/2/bin* and *DE/rand/2/bin*, while the worst results may be expected from the variants of *DE/current-to-rand/1/exp* and *DE/current-to-best/1/exp* for 100 dimensions. For that class of problem, *DE/best/2/bin* also saw the best performance for the 500 and 1000 dimension cases [109]. Dimensions of

30, 100, 500 and 1000 were tested while the population size was fixed at 100. Another study tested different variants on small dimensional problems and arrived at a different conclusion, where the most all-around competitive variants were *DE/best/1/bin,* DE/rand/1/bin, *DE/rand/2/dir, and* DE/current-to-rand/1/bin [110]. The differences in outcomes for the various empirical studies highlight the importance in applying attempting more than one set of parameters.

2.6 Tools and Data

2.6.1 Software

The development of solar energy forecasts was done using *R* [111]. The multilayer perceptron (MLP), Elman, and Jordan neural networks used were from the *R RSNNS* library [112] (available: CRAN). The random forest implementation used was from the *R randomForest* library (available: CRAN) [83].

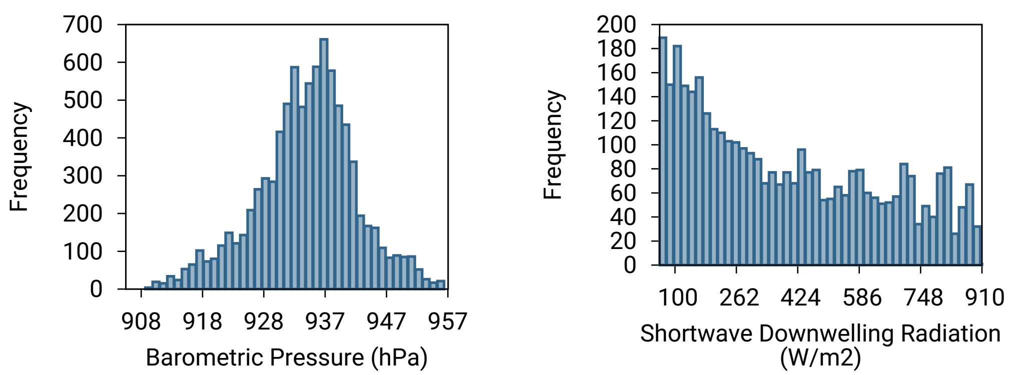
Network simulations were carried out using the *shawn* network simulator [113] (available: GitHub). This simulator is fast and extensible, allowing energy considerations, like the charging and discharging operations of a sensor node, to be added to the simulation. Outputs of the nodes were taken from transmissions logged during simulation and were then post-processed to extract information regarding battery reserve and supercapacitor charge levels, as well as the measurements themselves. Fuzzy logic was implemented using the *fuzzylite* library [114] (available: GitHub). In order to support sampling the time series at arbitrary points, they were fit with interpolating splines using the GNU Scientific Library [115].

2.6.2 Available Data

Two meteorological data sources were used for this work, including both measured data and the output of a numerical weather prediction model.

The site initially examined was located just outside of Fairview, Alberta, Canada at 56.0815° latitude, -118.4395° longitude with an elevation of 665 m. This location was selected due to its proximity to a forest ecosystem monitoring program, allowing for further development and field testing in the future. Synthetic meteorological data were created using the Weather Research Forecasting system (WRF) numerical model, with input data from the Global Forecasting System (GFS). GFS data for the entire year of 2012 were processed and the exact location was extracted. The WRF output variables *PSFC* (surface pressure), *SWDOWN* (downward shortwave flux at ground surface), and *SWDNBC* (instantaneous downwelling clear sky shortwave flux at bottom) were used. Unfortunately, while measurements of solar energy were available, actual surface pressure measurements were not available at this site.

For the data generated for the Fairview site, the histogram of hourly pressure values is shown in Fig. 2.1a and the histogram of daytime solar irradiance values is shown in Fig. 2.1b. While the distribution of pressure measurements is closer to a normal distribution, the distribution of the *SWDOWN* for this site is skewed towards lower values, as would be expected given the sinusoidal nature of this variable.



(a) Pressure values. (b) SWDOWN (nighttime values excluded).

Figure 2.1: Histograms of hourly values from the Fairview site.

The second site used data from a network of automated meteorological stations operated by Washington State University (WSU). In this network, stations take a variety of measurements including temperature, pressure, and solar irradiance. Data are measured once every five seconds, summarized by the data logger and reported every 15 minutes. While the WSU network has many stations, not all of them have the necessary instrumentation to measure atmospheric pressure. Four stations providing the necessary measurements were selected for use. Locations of the selected stations are shown in Fig. 2.2 and tabulated in Table 2.3. Distances between the stations, based on the latitudes and longitudes, are shown in Table 2.4.

Figure 2.2: Locations of WSU meteorological stations.

Table 2.3: Locations for University of Washington automated meteorological stations.

Station Name	Latitude (°)	Longitude (°)	Elevation (m)
Garfield East	47	-117.06	849
Lind	47	-118.57	491
Moxee	46.54	-120.35	341
WSU Prosser	46.26	-119.74	265

The distributed measurements of the WSU data allowed for additional forecasting schemes to be explored. One of these schemes was to use measurements

Table 2.4: Distances between WSU Stations (km).

	Garfield East	Lind	Moxee	WSU Prosser
Garfield East	0	114.5	255.7	210.4
Lind	114.5	0	144.9	102.7
Moxee	255.7	144.9	0	46.66
WSU Prosser	210.4	102.7	46.66	0

taken from individual stations as predictor inputs to attempt forecast improvement by leveraging the movement of weather systems. The second scheme was to investigate whether lower errors could be achieved by creating a forecast from a combination of all of the measurements (i.e., using the entire data set to create a more general forecasting model for the network), or if creating forecast models for the individual nodes using only data measured at those locations results in better error values.

The primary differences between these two data sets are as follows:

- location, Fairview is more northerly, while the WSU stations are closer to the coast and have variable elevations,
- data frequency, with Fairview WRF output frequency being hourly, as opposed to 15 minute frequency from WSU stations,
- measured values versus the output of a numerical weather prediction model, and
- single location versus spatially distributed values.

2.7 Chapter Summary

In this chapter, wireless sensor nodes and the problem of energy management for energy harvesting nodes were introduced. The computational techniques employed to both estimate future values of daily solar energies (Chapter 3) and to tune membership functions of fuzzy logic controllers to obtain the best performance with the provided information (Chapter 4) were presented, as were the software tools employed. The data used to create forecasting models and as inputs for simulations were presented and briefly examined.

Chapter 3

Solar Energy Forecasts

Atmospheric pressure was explored as a variable for the prediction of upcoming daily solar energy. The use of only atmospheric pressure for the forecast of solar energy removes many of the necessary inputs used in more typical forecasts. Other meteorological values (e.g., relative humidity) hold valuable information relating to cloud cover. However, using pressure as the sole predictor variable removes the need for other measurement instruments in cases where they are not associated with the ultimate goal of the monitoring network. This reduces the power requirements that would be associated with such instruments, as well as the amount of data storage required. There are a number of low power sensor options for the measurement of pressure, for example [116]. Use of atmospheric pressure also avoids potential forecast errors associated with measurement error from certain variables, such as temperature, which may arise due to sensor placement (e.g., shading) [50]. Pressure and cloud cover are generally considered on a spatial scale of 100-300 km.

The guidelines shown in Table 2.1 suggest that atmospheric pressure and its variation over time can be very indicative of changes in the weather pertaining to cloud cover, especially rules 3, 4, 8, and 9. Weather forecasts based solely on pressure have been used in consumer products (e.g., home weather stations) for a number of years [117, 118, 119]. The forecasting accuracy for such devices is claimed to be roughly 75% [120].

3.1 Perfect Energy Forecast

The primary object for comparison with the developed forecasts was a perfect forecast. The perfect forecast was created using the same measurements that were later used in the simulations. The values were created by numerically integrating the incoming solar energy curves with a step size of 1 second. This step size was chosen to match the time resolution used in the simulation. Comparisons with the perfect forecast allowed various error values to be calculated for the developed forecasts. It also allowed the tolerance of different controllers to error in the forecast to be examined.

3.2 Data Preprocessing

3.2.1 Solar Energy Preprocessing

The tilt of the Earth causes periodicity in the diurnal solar energy. In order to remove some of this periodicity from the time series, estimates of daily diurnal solar radiation are made using Eqn. 2.16. A plot of the daily incoming solar energy derived from meteorological data and the analytical estimate of solar energies is shown in Fig. 3.1.

The measured daily solar energy values E_{DOWN} were divided by the calculated value $\hat{E}_A$ to estimate transmissivity:

$$\hat{\tau}_F = \frac{E_{DOWN}}{\hat{E}_A}, \tag{3.1}$$

where E_{DOWN} is the calculated daily solar energy. Using this multiplicative decomposition, $\hat{\tau}_F$ can be thought of another transmissivity factor corresponding to cloud

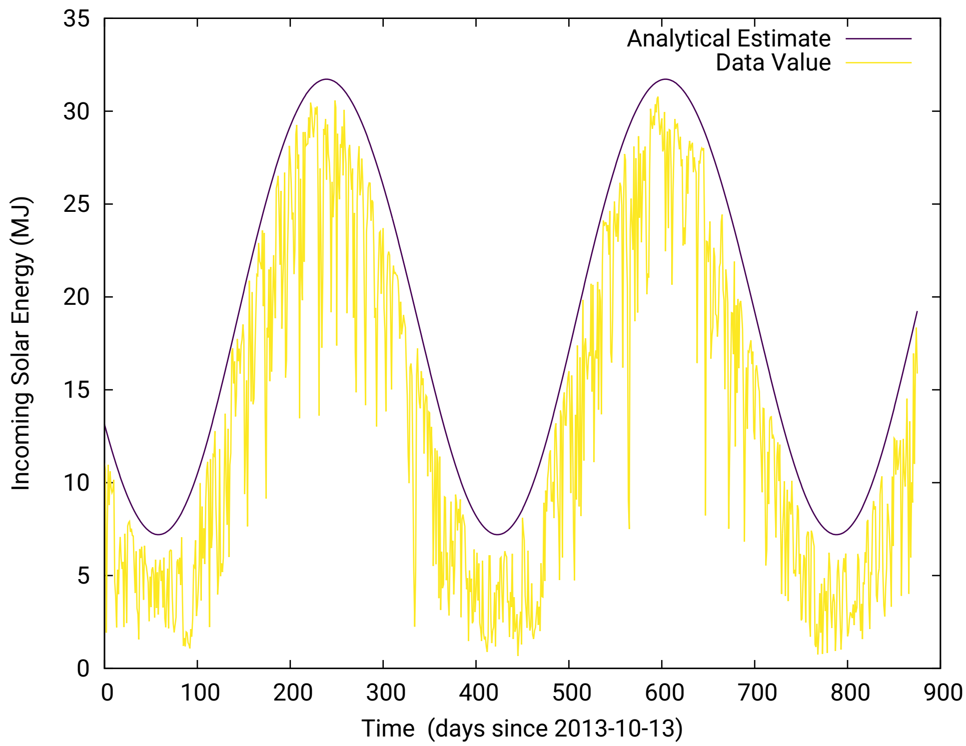


Figure 3.1: Estimated empirical and analytical (Eqn. 2.16) daily solar energy for WSU Moxee station.

cover. This estimate accounts for only the incoming light blocked by clouds and cannot account for other factors like atmospheric dust. The values of $\hat{\tau}_F$ were used as the target variable to be predicted. By predicting this ratio, dependence on the day of the year is be partially removed and allowed for the applied forecasting methods to omit it as a predictor variable. A plot of this estimated transmissivity value for the Moxee site is shown in Fig. 3.2.

To create the solar energy forecast, $\hat{\tau}_F$ was predicted and multiplied by the calculated value of $\hat{E}_A$. Using visual inspection, it can be seen that Fig. 3.2 still contains a periodic component. This is not unexpected, as some seasons experience more cloud cover compared to others. Other methods of removing seasonality may be used. For example, decomposing a log transformed solar energy time series using STL (Seasonal-Trend Decomposition based on Loess) decomposition to create a multiplicative decomposition [121, 122]. The plot of this decomposed time series

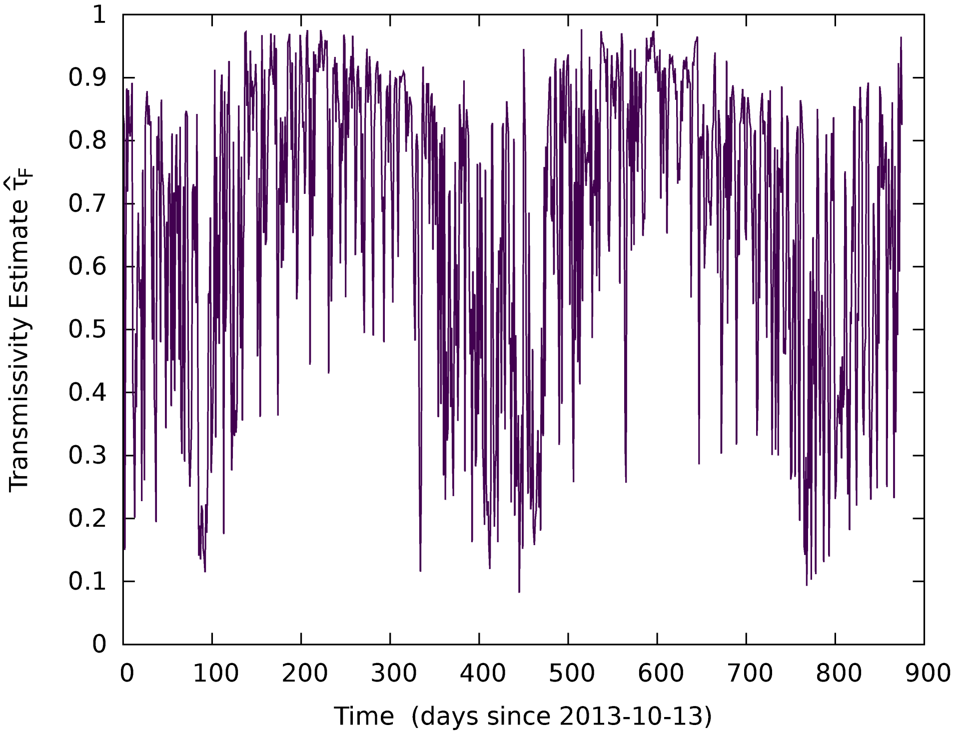


Figure 3.2: Transmissivity estimates $\hat{\tau}_F$ for daily solar energy for WSU Moxee station.

for seasonal, trend, and remainder are shown in Fig. 3.3 (note the different scales for seasonal and remainder components). This decomposition detected a similarly shaped seasonal component to the analytical estimate. The scales of the components of this decomposition can be changed by moving portions of different components (e.g., to reduce the size of the trend component while increasing the size of the seasonal component). However, the drawback of requiring enough data from a site in order perform this estimation, and to perform online updates to the trend component may not be justified. The simpler method of estimating transmissivity has been used in the rest of the work.

The histogram of the estimated transmissivity of solar energy reaching the Fairview site is shown in Fig. 3.4. The estimated transmissivity for this data set had an average value of 0.7794 and a variance of 0.0395.

The histogram of the estimated transmissivity for all of the WSU sites is shown

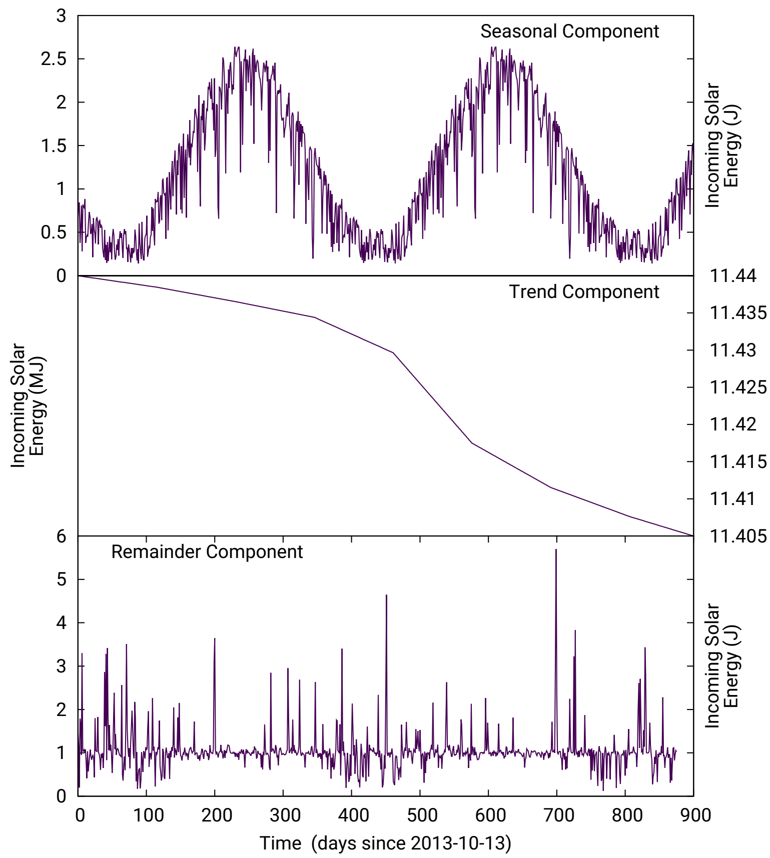


Figure 3.3: Naive multiplicative STL decomposition results for WSU Moxee site.

in Fig 3.5. The average and variance of the estimated transmissivity value for this data set were 0.6717 and 0.06074, respectively.

Both histograms show a similar pattern of higher frequencies of values in the range of approximately 0.8 to 1.0, indicating that these locations frequently receive solar energy close to the analytical estimate. The tails of the distributions show that there are many days when sites receive less than the analytical estimate. However, these occurred across a wide range of values. As expected, there were no instances of transmissivity values 0, as this would indicate that the sun was completely blocked out (or that there were instrument errors). The presence of transmissivity values above one indicated imperfection in the creation of the analytical estimate $\hat{E}_A$. Improvements could potentially be made through the use of a more

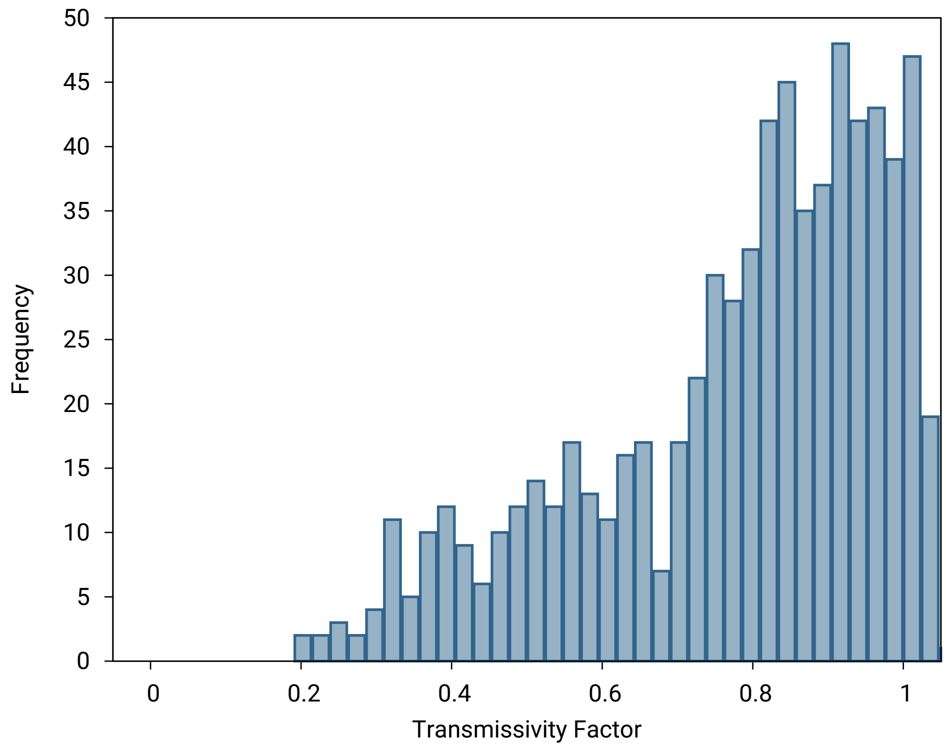


Figure 3.4: Histogram of daily averages of transmissivity factors ($\hat{\tau}_F$) for the Fairview site.

sophisticated model.

Table 3.1 shows the calculated *MAPE* values (Eqn. 2.19) for the case when the analytical estimate of total daily solar energy is used as the predictor of actual energy. These values highlight a number of differences between data sets. Firstly, the higher error of the 2013 Fairview data when compared to the 2012 data shows that there was likely more cloud cover during 2013. When compared to the WSU data, the Fairview *MAPE* are much lower. There are a number of possible explanations, including:

- The calculation method for $\hat{E}_A$ may not be as representative for the WSU locations,
- Fairview may experience more clear, cloud free days than central Washington,
- WSU data have additional measurement error, and

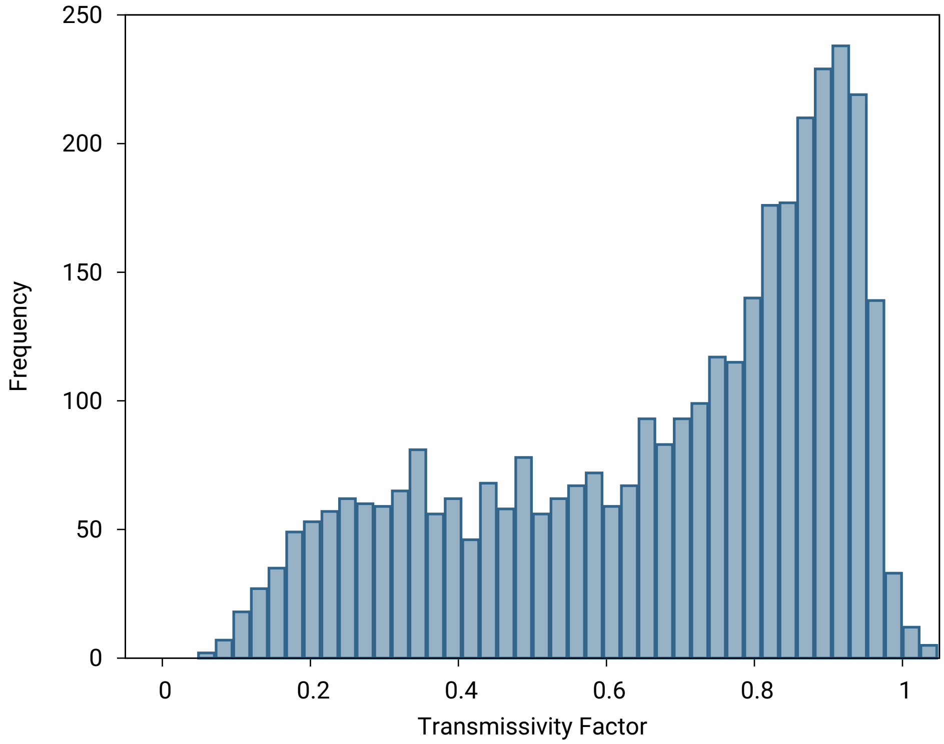


Figure 3.5: Histogram of daily averages of transmissivity factors ($\hat{\tau}_F$) for the WSU sites.

- Fairview data, coming from WRF, do not account for dust, ash, or other detrimental factors that may be present in the measured WSU data.

Table 3.1: *MAPE* values when $\hat{E}_A$ is used as a predictor.

Dataset	*MAPE* (%)
Fairview, 2012	44.05591
Fairviw, 2013	48.95873
Fairview, All	46.50732
WSU-Moxee	86.79939
WSU-Lind	85.61352
WSU-Prosser	84.91675
WSU-Garfield East	113.1569

Since the expected amount of solar energy changes over the course of a year, directly predicting its value may lead to poor performance during winter months. In order to avoid the larger errors possible when the amount of incoming energy is high, prediction methods may sacrifice performance during periods when expected energy is low. Unfortunately, these times are the most crucial to a sensor node, as there is a greater chance for usage to exceed the energy available for harvest, as well as less chance to replenish energy during the following days. This motivates using the transmissivity factor as the prediction target as it allows for errors to be spread more evenly across a year.

3.2.2 Pressure Preprocessing

Atmospheric pressure depends on temperature and site elevation. Measurements of atmospheric pressure at a given site P_{ST} can be converted to an equivalent sea level value P_{SL} using:

$$P_{SL} = \frac{P_{ST}}{e^{\frac{-h}{29.240T}}}, \tag{3.2}$$

where the pressure values are measured in hPa, h is the station elevation in metres, and temperature T is measured in Kelvin [47]. Performing this conversion did not produce statistically significant differences in the error values of the forecasts that follow and were thus omitted. This is likely because the differences in station elevations were not too great.

3.3 Forecast Development

3.3.1 Temperature-Based

One potential model for estimating the incoming daily solar energy is given by:

$$\hat{E}_{SH} = K_r \sqrt{T_{max} - T_{min}} \hat{E}_A, \tag{3.3}$$

where $\hat{E}_{SH}$ is an energy estimate using the Samandi-Hargreaves relationship from [57], K_r is an empirical constant (a typical value of 0.17 is used here), T_{max} and T_{min} are the measured maximum and minimum temperatures of the day, and $\hat{E}_A$ is the analytical energy estimate. Applying Eqn. 3.3 to the available data sets for the prediction of the next-day solar energy results in the *MAPE* and *RMSE* values given in Table 3.2. The *MAPE* are an improvement when compared to using solely $\hat{E}_A$ as the predictor (Table 3.1), but also suffer from the large values of *RMSE*.

Table 3.2: Error values for next-day solar energy prediction using max. and min. temperatures.

Station	*MAPE* (%)	*RMSE* (MJ/m^2/day
Moxee	34.67	3.74
Prosser	35.17	4.63
Lind	37.06	4.52
Garfield East	41.34	4.87
All WSU Stations	37.06	4.46
Fairview, all	38.41	7.50

3.3.2 Pressure Pairs

Initially, predictions were made at sunrise with the aim of supporting a fuzzy logic control scheme operating at that time [123]. Five pressure measurements were used as possible predictors of the estimated transmissivity factor $\hat{\tau}_F$. These pressure measurements were made with relation to sunrise and sunset at the location of interest. The relationship between the sunrise, sunset, and the pressure measurements is shown in Fig. 3.6. The times between pressure measurements (i.e., lags) were varied in order to estimate the most important period during which the changes in atmospheric pressure impact the solar energy experienced by the site in the near future. The forecasts were made when the last pressure measurement in the set was taken.

A number of methods were employed, each with a different trade-off between complexity and prediction accuracy. The methods considered in this study included regression trees (CART), random forest regression (RF), multilayer perceptron neural networks (MLP), Elman recurrent neural networks and Jordan recurrent neural networks. The possible lag times, shown in Fig. 3.6, were varied between 1 and 18

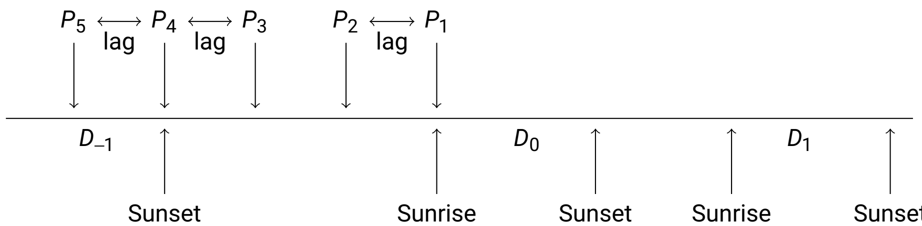


Figure 3.6: Measurement and prediction timeline for pressure pair based energy forecast showing relative timing of pressure measurements P to the sunrise and sunset of days D.

hours. 30 trials were executed for each scheme and lag time. For neural network methods, 6 hidden neurons were used. For networks requiring iterative training, 750 iterations were used.

Initially, only 2012 data from the Fairview site were available (see Sec. 2.6.2). Here, the training set was comprised of two-thirds of the data, with the test set comprised of the remaining third. The sets were selected as continuous blocks to allow the use of recurrent neural networks. Later, data for 2012 and 2013 became available. The same scheme was applied, but the training set included all the data from 2012 and the test set was comprised of the entire year 2013.

Two estimates of daily solar energy are made based on the WRF output variables SWDOWN (shortwave downwelling solar irradiance) and SWDNBC (shortwave downwelling solar irradiance for a clear sky). Using these outputs, estimates of the total daily incoming solar energy were created, $\hat{E}_{SWDOWN}$ and $\hat{E}_{SWDNBC}$. Since $\hat{E}_{SWDOWN}$ takes into account clouds and other factors that reduce solar radiation reaching the surface, it was considered as the true value. Though not as accurate as $\hat{E}_{SWDOWN}$, the estimate $\hat{E}_{SWDNBC}$ still considers more factors than $\hat{E}_A$ and should be considered 'more true'. The *MAPE* between $\hat{E}_{SWDNBC}$ and $\hat{E}_A$ for the Fairview site during 2012 was calculated to be 10.56%. For that same time period and location, $\hat{E}_{SWDNBC}$ and $\hat{E}_A$ had *MAPE* values of 32.13% and 44.06%, respectively, relative to $\hat{E}_{SWDOWN}$. While $\hat{E}_{\mathrm{SWDNBC}}$ is a better estimate, using numerical weather prediction to arrive at the value is computationally intensive. Plots of the available solar energy estimates

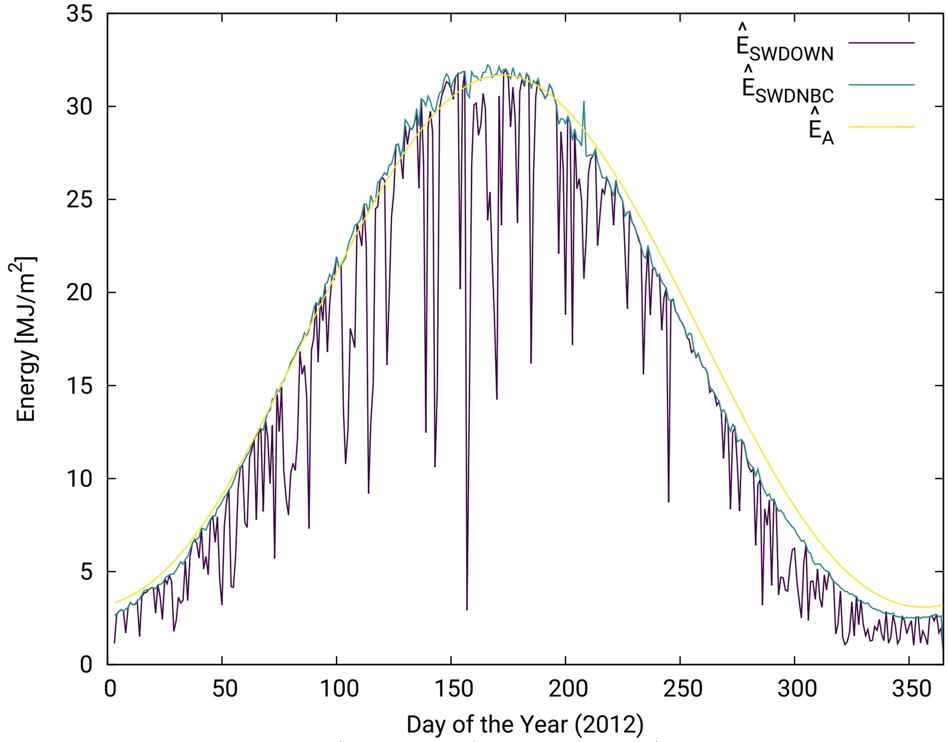


Figure 3.7: Plot of $\hat{E}_{SWDOWN}$, $\hat{E}_{SWDNBC}$, and $\hat{E}_A$ (Fairview 2012).

are shown in Fig. 3.7.

A number of schemes were used to predict the daily solar energy based on barometric pressure at various points prior to the day in question. Using the pressure measurements shown in Fig. 3.6, four schemes (pre-sunrise PreSR, post-sunset PostSS, pre-sunset PreSS, combined pre-sunrise post-sunset C.PreSRPostSS) were defined (Table 3.3) and used to make solar energy predictions. In addition to the pressure values, pressure differences (defined as differences between the earlier pressure and the later pressure in the pair) were also provided to the prediction methods.

Since the pressure values were selected in relation to prediction day sunrise and to the sunset on the previous day, the forecasting horizon varied with the time of year. For the PreSR and C.PreSRPostSS schemes, the amount of energy available for harvest on a particular day was predicted at sunrise the same day. For the

Table 3.3: Pressure scheme names and variables

Scheme Name	Variables Used
PreSR	P_1, P_2
PostSS	P_3 , P_4
PreSS	P_4, P_5
C.PreSRPostSS	P_1, P_2, P_3, P_4

PostSS scheme, the energy was predicted several hours in advance. The lead time was equal to the number of lag hours subtracted from the number of night hours. The PreSS scheme made solar energy predictions for each day at sunset on the previous day (one full night ahead).

For all methods, except the random forest and the regression tree, all input variables were normalized and the output value was scaled to vary between 0 and 1. *MAPE* values for the six methods are shown in Table 3.4. In all cases, the random forest performs the best, possibly owing to the lack of normalization, training iteration and selection of the number of hidden neurons. Comparing the values in Table 3.4 with the *MAPE* values for predicting $\hat{E}_{SWDOWN}$ using $\hat{E}_{SWDNBC}$ (32.13%) and $\hat{E}_A$ (44.06%), showed that all methods were capable of providing some improvement over the use of $\hat{E}_A$. However, in order to improve on $\hat{E}_{SWDNBC}$ as a solar energy forecast, tree-based methods worked best. The MLP network made a more modest improvement. For this case, most methods had lag values between 7 and 10 hours. This led to some overlap between pressures P_2 and P_3, during times of the year with short nights.

The time series for the best performing methods, as ranked by *MAPE*, are shown in Fig. 3.8. These series show that the recurrent neural network methods significantly and consistently under-predicted the target solar energy. All methods appeared to under-predict the highs and lows during times of higher energy (summer), while they seemed to over-predict when the amount of energy is low (winter). The over-prediction during winter may be due to the large difference between $\hat{E}_A$ and $\hat{E}_{SWDOWN}$ at this time. Improving the analytical calculation to more closely match

Table 3.4: Minimum and maximum training *MAPE* values for different prediction methods, with scheme and lag noted for the minimum (Fairview 2012).

	Min (%)	Max (%)	Best Scheme	Best Lag (hours)
RF	13.67	16.99	C.PreSRPostSS	9
CART	20.87	31.82	C.PreSRPostSS	8
MLP	26.90	32.89	C.PreSRPostSS	9
ELM	30.92	35.76	C.PreSRPostSS	10
Elman	32.64	44.28	C.PreSRPostSS	3
Jordan	34.81	40.02	C.PreSRPostSS	7

Table 3.5: *RMSE* values corresponding to the predictions from Table 3.4 (Fairview 2012).

Method	*RMSE* (MJ/m^2/day)
RF	1.645
CART	2.757
MLP	3.515
ELM	3.599
Elman	6.488
Jordan	6.978

the WRF output values may improve forecasting during these time periods.

A plot of the best performing regression tree is shown in Fig. 3.9, which indicates the thresholds for the included pressures, and the cutoffs for the pressure differences. The plot of this tree highlights the ease of implementation of the CART prediction method.

Table 3.6: Minimum and maximum *MAPE* (%) values for different prediction methods, with scheme and lag noted for the minimum training error (2012 Fairview).

Method	Min. (%)	Max. (%)	Best Scheme	Best Lag (hours)
RF	12.83	16.09	C.PreSRPostSS	9
CART	20.81	29.78	C.PreSRPostSS	5
MLP	27.52	35.14	C.PreSRPostSS	3
ELM	29.64	35.63	C.PreSRPostSS	1
Elman	24.97	31.09	C.PreSRPostSS	11
Jordan	29.13	33.98	C.PreSRPostSS	9

In terms of *MAPE* values, random forests consistently had the lowest error of all methods. Compared to the best scheme, C.PreSRPostSS, with a *MAPE* value of

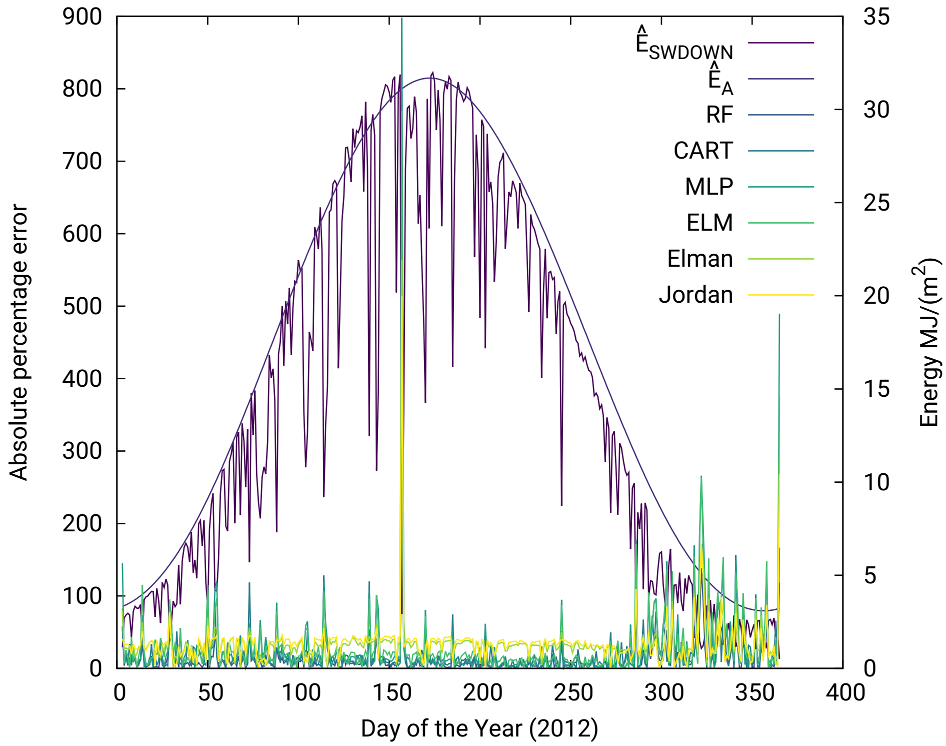


Figure 3.8: Absolute percentage error, daily solar energy and analytic daily solar energy (Fairview 2012). Gaussian shaped curves correspond to the solar energy estimates and use the scale on the right, while the lower values correspond to error values and use the scale on the left.

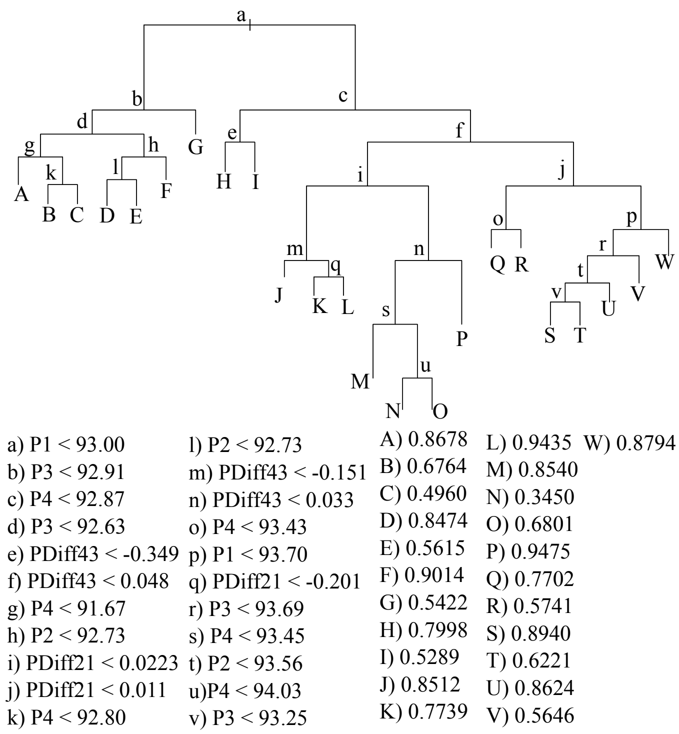


Figure 3.9: Plot of the best performing regression tree (Fairview 2012).

13.67%, the PreSS and PostSS schemes had MAPE values of 15.29% and 15.22%, respectively, with corresponding lags of 7 and 9 hours. This represented a modest increase in error, which may be acceptable given the increased prediction horizon in the case of the PreSS scheme.

In terms of minimum *MAPE* values for the regression trees, the best scheme was C.PreSRPostSS with 20.87%, while the PreSS and PostSS schemes had values of 23.97% and 24.65%, respectively, with corresponding lags of 6 and 10 hours.

In cases where an increased prediction horizon is desirable, the PostSS scheme could first be used to get an early initial estimate. This could then be refined later using a C.PreSRPostSS scheme, since the PostSS pressure values would have already been measured. Such a combined approach would require the use of both trained models and may only be feasible when using the simpler methods.

Table 3.7: Error values corresponding to the predictions from Table 3.6

Method	*RMSE* of training set ($MJ/m^2/day$)	*MAPE* (%) of test set	*RMSE* of test set ($MJ/m^2/day$)
RF	1.643	29.63	3.262
CART	2.802	31.17	3.835
MLP	4.695	34.30	5.917
ELM	5.181	34.44	5.060
Elman	3.428	30.96	3.547
Jordan	3.703	30.82	3.368

3.3.3 24 Hourly Pressure Measurements and Longer Prediction Horizons

Using the same measurements taken to create the single day horizon forecast, a forecast for the daylight hours of the next day and beyond can also be made. The larger forecasting horizon was considered as it could potentially improve performance of the controller to be developed later. However, because of the limited capacity of a sensor node's energy buffer, there was expected to be a limit to the size of a useful forecasting horizon.

At sunrise on a given day, the incoming solar energy for that day and the solar energy for the next five days were predicted. As opposed to the selection of variables shown in Fig. 3.6, 24 hourly pressure measurements were supplied, as shown in Fig. 3.10. The differences between the individual pressure measurements were calculated and supplied as inputs as well, since the change in pressure over time was expected to be the major indicator of weather changes. In the previous exploration there was no input corresponding to the time of the year or to the expected amount of solar energy. However, a dependence between the time of the year and the size of the prediction error was noted when longer time series were used (Fig. 3.2). In an attempt to lower these errors, the analytical estimate of the total incoming energy (Eqn. 2.16) was also provided as an input. For these predictions the CART, RF, ELM, and MLP methods were used. The sizes of the ELM and MLP neural networks were expanded to 50 hidden nodes, and the maximum number of iterations was increased to 1000. The inputs for the neural network methods were again scaled to lie between 0 and 1, while the CART and RF inputs were untouched. The models trained for the Fairview site had constant training and test sets corresponding to the data from 2012 and 2013, respectively, while the WSU models were trained on randomly selected training sets comprising 50% of the total available days. Ten trials were run for each prediction method and data set.

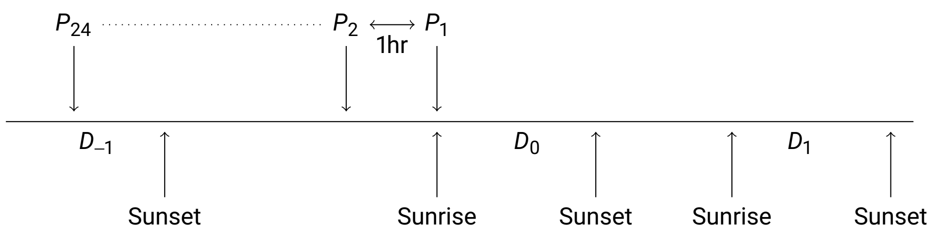


Figure 3.10: Measurement and prediction timeline using 24 hourly pressure measurements, $P_1 \ldots P_{24}$, in relation to the sunrise of day D_0.

Tabulated summaries of the training errors for the different methods are show in Tables 3.8- 3.11. Error values for the test sets using the best performing individuals are shown in Tables 3.12 and 3.13 for the Fairview and WSU data sets, respectively. Standard deviations for the CART prediction of the Fairview data are zero because the training set remained the same across all trials. With respect to *MAPE*, the CART prediction performed the best on the Fairview training set, but resulted in the highest test error, pointing to over-fitting. This difference was much less for the WSU dataset, likely because of the random selection of training set for this data set resulting in a more generally representative collection of measurements. For the Fairview dataset, all error metrics saw the best performance from the CART regression. The neural networks had similar error values for both *MAPE* and *RMSE*, but the MLP regression had better performance with respect to *MAE*. This suggests that with respect to the Fairview training set, the MLP prediction had better overall tracking, but more instances of very large errors.

For the WSU training data, the RF and CART regression had the worst performance with respect to *MAPE*, but had better performance compared to the neural network methods for the other two metrics.

Table 3.8: *MAPE* (%) training values for current day (D_0) prediction (Fairview).

Method	Min.	Average (σ)	Max.
CART	13.22	13.22 (0)	13.22
RF	22.66	22.71 (0.0330)	22.76
ELM	21.51	21.57 (0.0515)	21.67
MLP	21.28	21.40 (0.0763)	21.56

Table 3.9: *RMSE* (J/m^2/day) training values for current day (D_0) prediction (Fairview).

Method	Min.	Average (σ)	Max.
CART	2151498	2151498 (0)	2151498
RF	3013183	3019578 (4258)	3027674
ELM	2425244	2455591 (27261)	2523193
MLP	2299414	2330031 (20104)	2362040

Using the models with the lowest respective training errors, values for the test set were obtained and are tabulated in Table 3.12 and 3.13 for Fairview and WSU, respectively. The neural networks methods performed better for both data sets with respect to the *MAPE* metric.

Comparison of the results are shown graphically using the skill scores for the different error measures. The reference forecast used was the analytical solar energy, $\hat{E}_A$. A single constant value of that $\hat{\tau}_F$ minimized the *MAPE* value was determined for each of the datasets and included for comparison.

The test set skill scores (see Sec. 2.4) for different error values for the models that performed best on the training set are shown in Figs. 3.11, 3.12, 3.12, 3.14, 3.15, and 3.16.

For the Fairview data set and with respect to *MAPE*, the MLP method showed the largest improvement over $\hat{E}_A$. Unsurprisingly, given the randomness involved in the training method, ELM was the most erratic. The tree-based methods had more moderate improvements, with RF showing a slight improvement over the CART tree. Aside from the ELM prediction, the forecasts were fairly stable for the increasing forecasting horizons. For the constant $\hat{\tau}_F$ selected to minimize the *MAPE* of the energy forecast, it unsurprisingly provided some improvement over the use of $\hat{E}_A$,

Table 3.10: *MAPE* (%) training values for current day (D_0) prediction (WSU).

Method	Min.	Average (σ)	Max.
CART	32.76	34.29 (0.09874)	36.23
RF	32.62	33.81 (0.09608)	35.72
ELM	29.37	29.41 (0.00278)	29.46
MLP	32.07	32.86 (0.06170)	34.35

Table 3.11: *RMSE* (J/m^2/day) training values for current day (D_0) prediction (WSU).

Method	Min.	Average (σ)	Max.
CART	3006957	3132965 (60020)	3194933
RF	2816788	2927624 (38734)	2957873
ELM	3818006	3906135 (69619)	4016836
MLP	3327944	3437591 (49894)	3496477

as the average transmissivity value would certainly be less than one. For *RMSE* and *MAE*, the obtained constant results in worse values than the $\hat{E}_A$ estimate. Overall, for this dataset the most improvement over the basic estimate was seen by MLP.

With respect to the constant value used for the WSU dataset, the case was the same as with the Fairview set, with the constant-based forecast being an improvement over $\hat{E}_A$ with respect to *MAPE*, but providing higher error values with respect to *RMSE* and *MAE*. As with the other dataset, the neural network based methods had a better improvement than the tree based methods with respect to *MAPE*. However, the tree based methods showed improvements that were equal to or better than the neural networks when *RMSE* and *MAE* were used for the comparison. As with the Fairview data, the forecasts for this dataset showed a fairly consistent improvement over $\hat{E}_A$ across the different forecast horizons.

Overall, all of the methods generally showed an improvement when compared to the $\hat{E}_A$ forecast, regardless of the error metric used. The ELM network was usually more erratic in its performance, pointing to the need for a larger number of test trials in order to select the best model. No model stood out as being the consistent best performer. Relative performance depended more on the dataset and chosen error metric. Not surprisingly, the constant transmissivity determined to minimize *MAPE* led to worse performance with respect to the other two error metrics consid-

Table 3.12: Test values for current day (D_0) prediction using minimum training model (Fairview).

Method	*MAPE* (%)	*RMSE* (J/m^2/day)	*MAE* (J/m^2/day)
CART	25.66	3547483	2328119
RF	23.84	2953170	2091640
ELM	21.33	2695043	2079435
MLP	19.42	2624678	1898017

Table 3.13: Test values for current day (D_0) prediction using minimum training model (WSU).

Method	*MAPE* (%)	*RMSE* (J/m^2/day)	*MAE* (J/m^2/day)
CART	38.34	3410469	2596852
RF	38.37	3251470	2500198
ELM	32.95	6893808	3174793
MLP	33.79	3594110	3143653

ered. This highlights the challenges of selecting the error metric on which to base optimization efforts. Final selection of a model to provide a sensor node with solar energy forecast will also depend on the tolerance of the controller to error. Due to the limitations of the nodes with respect to memory, computing power, etc., the simpler models may be preferred.

Comparison with reported errors from other works was difficult because of the differences between sites, including the variability of solar energy due to clouds. The best comparisons could be made with other works where the Fairview site was the focus. The best errors reported in [61], where fuzzy evolutionary rules and support vector machines were used to predict the next day solar energy were 2.960 MJ/m^2 day *RMSE* using fuzzy rules and a *MAPE* of 22.40%. The best error values reported from [62], where evolutionary fuzzy rules were used where a *MAPE* value of 23.40% (test) and an *RMSE* of 2.310 MJ/m^2 day (training set). Both used different fitness functions during the creation of the rules. Where the same training and test sets were used, the RF, ELM, and MLP predictions made here performed slightly better than the support vector regression and evolutionary fuzzy rule predictions.

The error values obtained for the Fairview site also seemed comparable to the values obtained for other sites, e.g. an *RMSE* of 2.75 MJ/m^2/day and a *MAPE* of

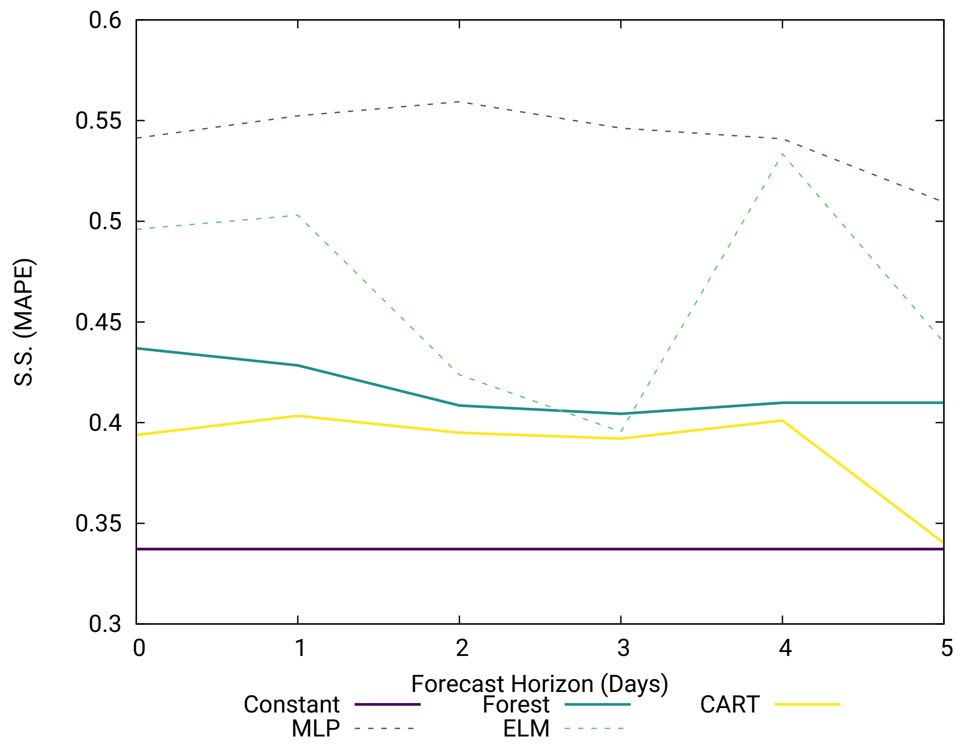


Figure 3.11: *MAPE* based skill scores for Fairview test set.

19.30% as reported in [58].

3.3.4 Distributed Measurements

The field of weather forecasting made large strides as communication technology allowed weather reports to travel faster than the weather itself. This allowed the reports to be used in a weather forecast. For a WSN with sufficiently large spatial coverage, the differences in measured meteorological values could be leveraged to provide a more accurate forecasted solar energy, especially for longer time horizons. In order to examine this possibility, forecasts were created using regression tree and random forest methods. For both of these techniques, 3 schemes were investigated: the case where all measurements across the network were used in forecast creation (ignoring the spatial separation), the case where only measurements local to the given nodes were used to create the forecast, and the case where

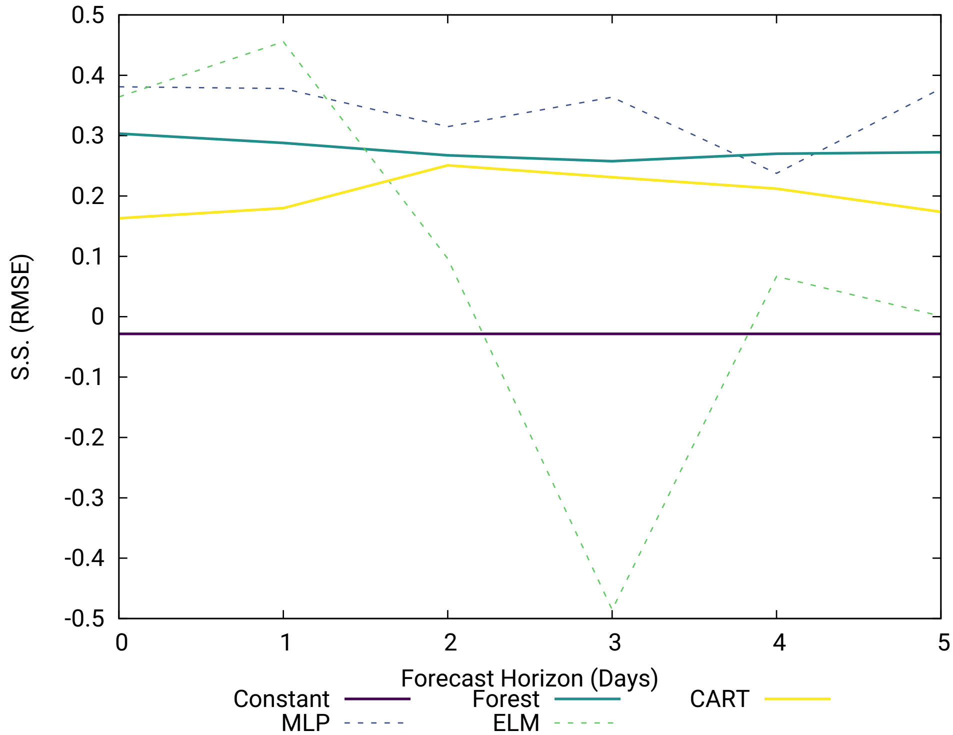


Figure 3.12: *RMSE* based skill scores for Fairview test set.

measurements made at other nodes were available for use in prediction across the entire network. The errors obtained on the test sets for the various combinations were compared using single tailed *t*-tests to determine if significant improvement existed.

For the case where pressure measurements across the entire network were available for use in the energy forecast, there were many possible variables to use as each location in the network has its own time series of meteorological values to draw upon. Therefore, a determination of the most important values for use in prediction should be made in order to reduce the number of required measurements. Additionally, because of the movement of weather systems, variables important for prediction at one location will likely not be as important to another. For this case each node should receive its own forecast, instead of using a single forecast for the entire network.

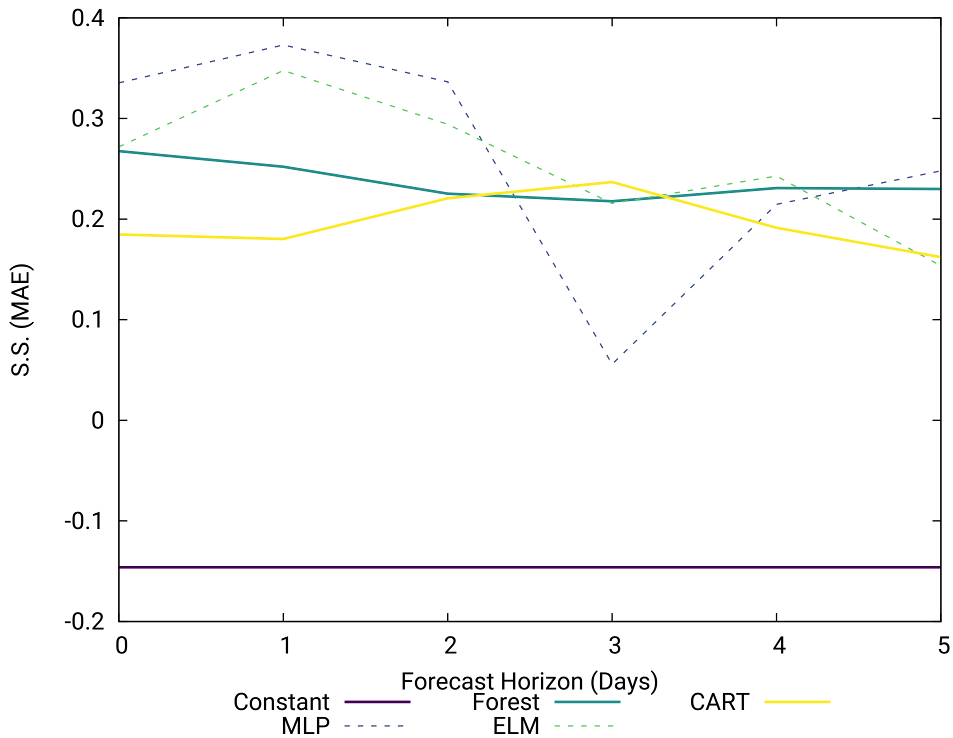


Figure 3.13: *MAE* based skill scores for Fairview test set.

Regression trees have an embedded variable selection built into the algorithm, as the limited allowed tree size forces the most predictively powerful variables to be selected. Variable importance ranking may also come from methods like random forests as the number of times a variable is selected for use in a tree [124].

Using the variable importance plots provided by the R randomForest package, the importance of pressure measurements from different stations for the prediction of transmissivity was investigated. For the distributed case, and withholding the analytical estimate from the regression, the top 5 variables for different forecasting horizons are shown in Tables 3.14, 3.15 and 3.16 for the Moxee, Lind, and Garfield East stations, respectively. Names of the stations having taken the measurements have been abbreviated in the tables as M, P, and EG, for Moxee, Prosser and East Garfield, respectively. If there was a consistent movement of weather across the network, it may have been expected to see measurements made at specific stations

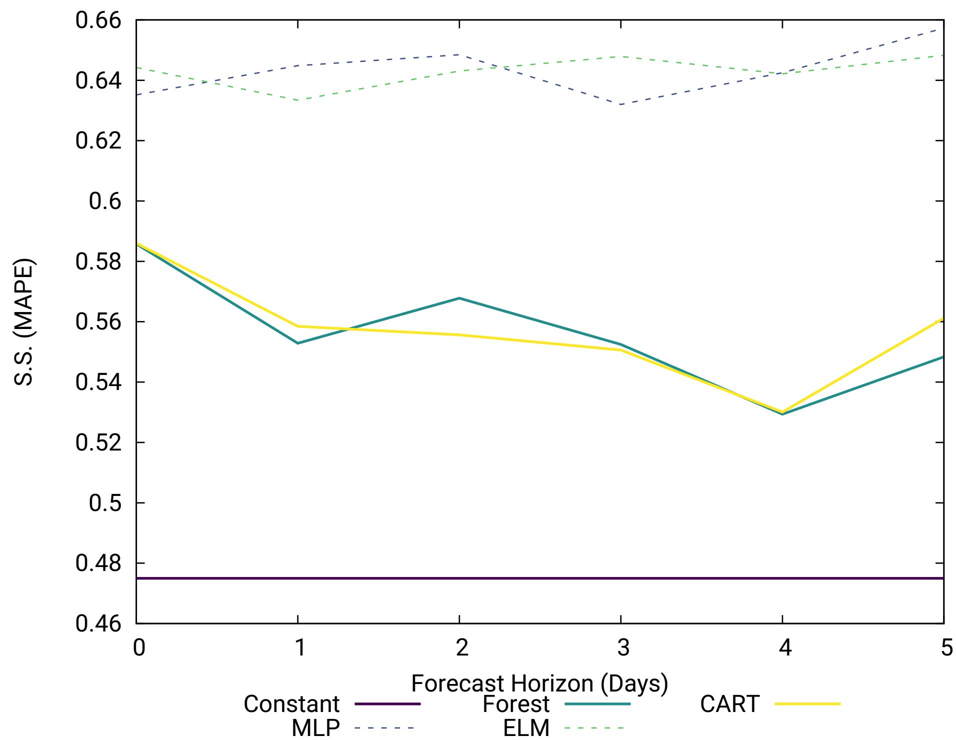


Figure 3.14: *MAPE* based skill scores for WSU test set.

showing up more consistently in the important variables for other stations. Based on the tabulated results, this does not appear to be the case. Measurements made at Moxee and Prosser dominate for the bulk of predictions. However, the relative lack of single pressure measurements in the most important variables, confirms the importance of changes in pressure for these predictions. Additionally, the lack of any of Lind's measurements in the variable rankings, even for predictions made at that station, is an indicator for possible improvements in the distributed prediction scheme.

Table 3.14: Top 5 Moxee station random forest variable importances for distributed regression where the analytical estimate was not provided.

	D_0	D_1	D_2	D_3	D_4	D_5
1	$P_{12} - P_5$ M	$P_{24} - P_{21}$ EG	P_{12} P	$P_{23} - P_{21}$ P	$P_{18} - P_{14}$ P	P_1 P
2	$P_{24} - P_{21}$ EG	$P_{19} - P_{16}$ P	$P_{18} - P_{16}$ P	P_{23} P	$P_{20} - P_{14}$ P	$P_{19} - P_{15}$ M
3	$P_{11} - P_2$ P	$P_{12} - P_9$ M	$P_{19} - P_{16}$ P	$P_{23} - P_{20}$ P	P_{14} P	$P_{23} - P_{20}$ P
4	$P_{13} - P_2$ M	$P_{24} - P_20$ EG	$P_{11} - P_9$ M	P_{19} M	$P_{18} - P_{13}$ P	$P_{18} - P_{14}$ P
5	$P_{11} - P_1$ P	$P_{18} - P_{16}$ P	$P_{12} - P_9$ M	P_{19} P	P_{15} P	$P_{20} - P_{15}$ P

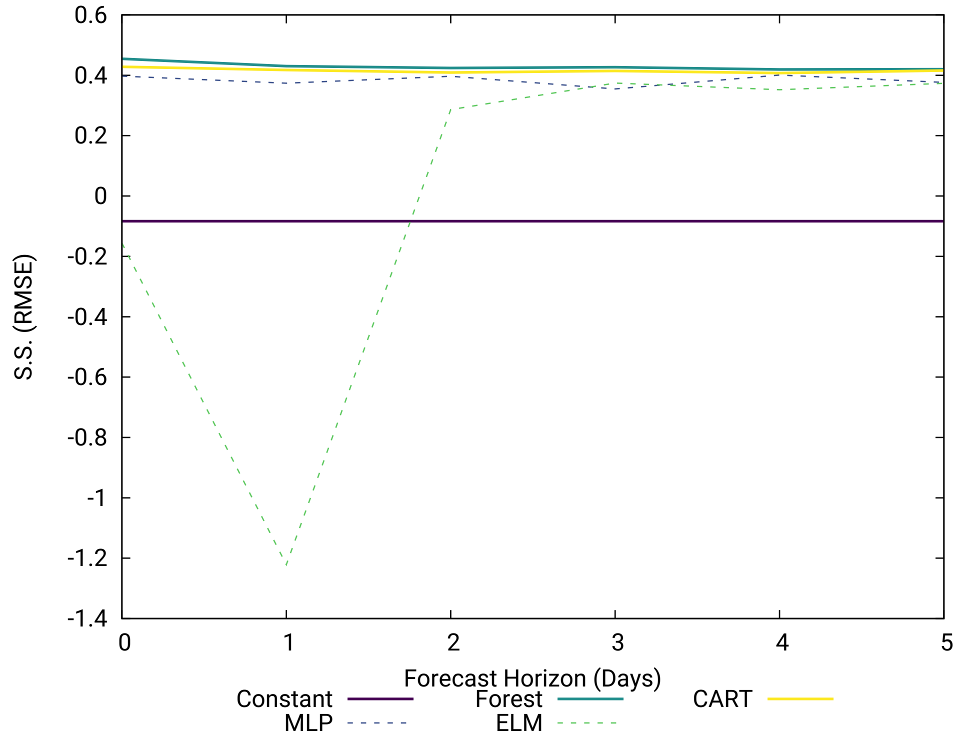


Figure 3.15: *RMSE* based skill scores for WSU test set.

Table 3.15: Top 5 Lind station random forest variable importances for distributed regression where the analytical estimate was not provided.

	D_0	D_1	D_2	D_3	D_4	D_5
1	$P_{12}-P_4$ M	$P_{19}-P_{16}$ P	$P_{18}-P_{16}$ P	P_{12} P	$P_{23}-P_{20}$ P	$P_{19}-P_{15}$ M
2	$P_{12}-P_6$ M	$P_{18}-P_{16}$ P	$P_{19}-P_{16}$ P	$P_{12}-P_9$ M	P_{24} P	$P_{23}-P_{20}$ P
3	$P_{11}-P_1$ P	$P_{19}-P_{15}$ P	$P_{22}-P_{20}$ P	$P_{12}-P_5$ M	$P_{19}-P_{16}$ P	$P_{20}-P_{15}$ P
4	$P_{12}-P_1$ P	$P_{24}-P_{21}$ EG	$P_{19}-P_{16}$ EG	$P_{23}-P_{21}$ P	$P_{20}-P_{16}$ P	$P_{24}-P_{22}$ EG
5	$P_{12}-P_5$ M	$P_{11}-P_7$ P	$P_{24}-P_{21}$ EG	$P_{20}-P_{14}$ P	$P_{23}-P_{21}$ P	$P_{23}-P_{22}$ EG

Use of regression trees with information from individual stations and measurements from the entire network resulted in *MAPE* values shown in Fig. 3.17. For the cases using random forest regression, the *MAPE* values are shown in Fig. 3.18.

Comparing the all-station measurement cases for the regression trees and random forest shows that the random forest case had a significantly lower average error than the regression tree for both the D_0 and D_1 forecasts. With longer forecasting horizons, the average errors were not significantly different. For the rest of the regression tree results, there were no significant improvements between the

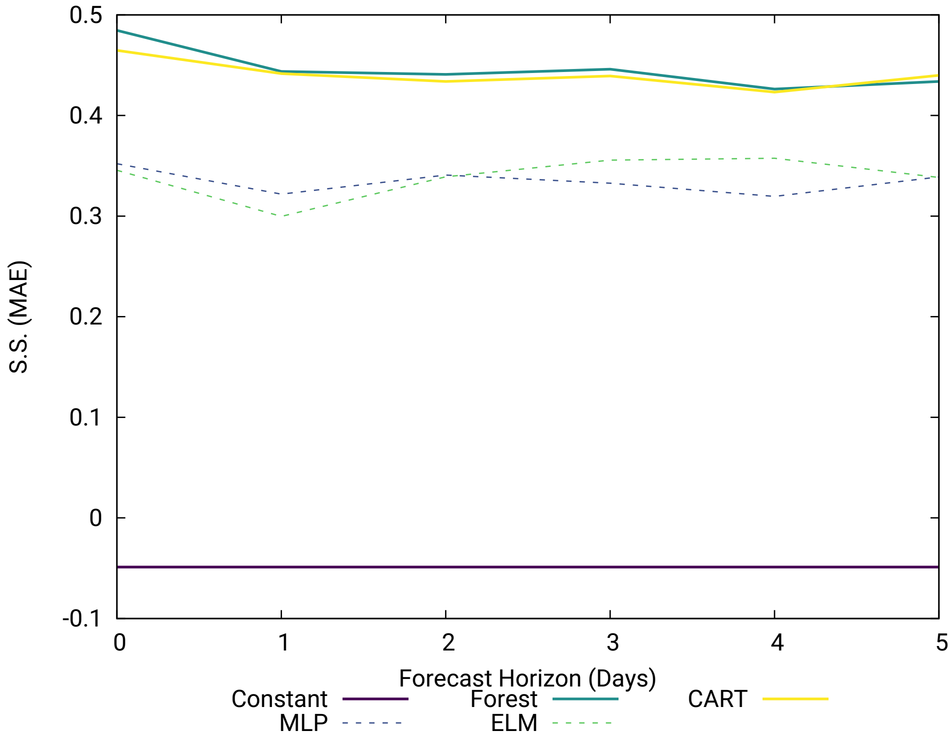


Figure 3.16: *MAE* based skill scores for WSU test set.

Table 3.16: Top 5 East Garfield station random forest variable importances for distributed regression where the analytical estimate was not provided.

	D_0	D_1	D_2	D_3	D_4	D_5
1	$P_{12} - P_4$ M	$P_{12} - P_8$ M	$P_{19} - P_{16}$ P	$P_{20} - P_{16}$ P	$P_{20} - P_{16}$ P	$P_{18} - P_{16}$ P
2	P_1 EG	$P_{12} - P_4$ M	$P_{11} - P_5$ M	$P_{11} - P_5$ M	$P_{19} - P_{15}$ M	P_{14} P
3	$P_{23} - P_{20}$ P	$P_{12} - P_5$ M	$P_{18} - P_{16}$ P	$P_{23} - P_{21}$ P	$P_{23} - P_{20}$ M	P_{12} P
4	$P_{19} - P_{15}$ P	$P_{17} - P_{15}$ P	$P_{20} - P_{16}$ P	$P_{18} - P_{16}$ P	$P_{19} - P_{16}$ P	$P_{19} - P_{15}$ M
5	$P_{13} - P_7$ P	$P_{12} - P_6$ M	$P_{19} - P_{15}$ P	$P_{19} - P_{16}$ P	$P_{18} - P_{16}$ P	$P_{22} - P_{20}$ M

distributed case and the case where only local measurements were used. However, there were a few instances where the average error was actually worse. For the case where all measurements were used in the non-distributed manner, there are no improvements excepting a few forecasting horizons for Prosser and East Garfield stations. For the case where all forecasts were used is compared to the distributed measurement case, East Garfield saw significant improvements, while other nodes saw no significant improvements.

For the random forest method, comparing the distributed to local measurement

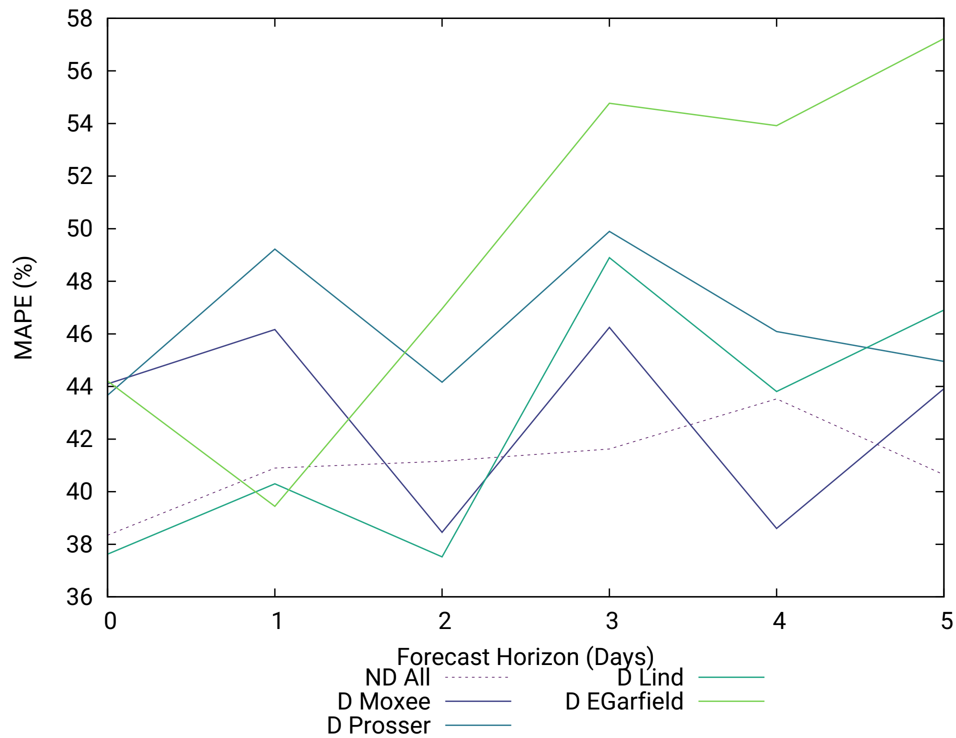


Figure 3.17: *MAPE* values for test set of station predictions using regression trees.

schemes showed a few cases of significant improvement for East Garfield, but nothing significant for any of the others. Comparing the all-measurement case to the local-measurement case, Lind and East Garfield both showed many instances of significant improvement, which was also the case for the comparison of all-measurement case to the distributed measurement case. Comparing just the distributed and local cases, there were only a few cases of improvement for East Garfield.

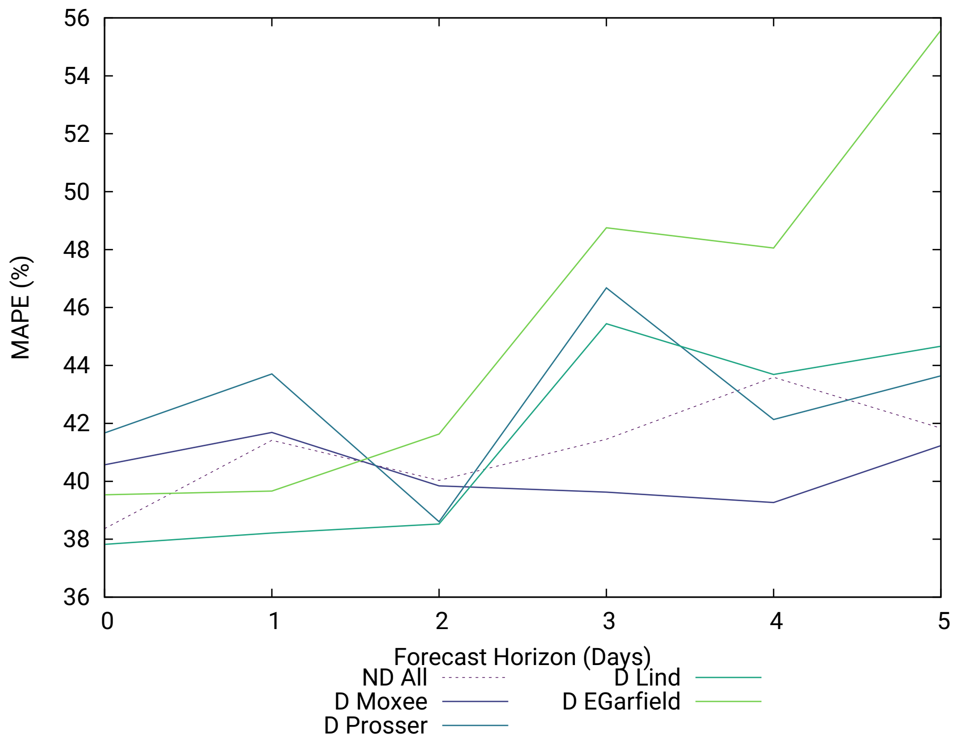


Figure 3.18: *MAPE* values for test set of station predictions using random forests.

3.4 Chapter Summary

In this chapter, the method of creating a forecast of daily incoming solar energies was outlined and involved the calculation of the clear-sky analytical solar energy estimate and the application of the regression techniques introduced in Sec. 2.5.1. A few different configurations of the pressure measurements used for prediction were explored, including a distributed forecast. A regression tree generated in this chapter was used as a representative forecasting model in the simulations of optimized controllers, presented in Chapter 5.

Chapter 4

Simulation of a WSN and Controller Optimization

Simulation of WSN avoids long-term field trials for the development of energy management strategies. While using software simulations may not capture every facet that will be present during actual deployment, they do allow initial development to be performed much faster and cheaply. For cases where sensor nodes make use of energy harvesting associated with weather related phenomenon, either measured data or outputs of numerical weather prediction models can be used to provide realistic weather conditions that the resulting simulation more representative of actual deployment conditions. Additionally, for the purposes of controller optimization using evolutionary methods, it would be impossible perform such an optimization without a repeatable, representative simulation of the node in question.

Using the solar energy forecasts developed in Chapter 3, a number of simulations were performed using the *shawn* simulator. This simulator was extended with additional code to handle solar charging, discharging, the taking of measurements, as well as message processing and transmission. The simulator was also used to determine fitness function values of different fuzzy controllers as part of controller optimization.

Other work for the simulation of a wireless sensor node was presented in [37], where a Simulink simulation was developed to examine the effectiveness of a dynamic fuzzy controller versus static versions. The simulator used three modules to model the solar panel, controller and other hardware. Measured data from the Fairview ACIS site were used. Simulation of energy harvesting monitoring stations in arctic regions is presented in [30]. A number of simulations were also presented in [125], also using Simulink, for arctic, boreal and tropical dry forest locations. In these simulations, an adaptive fuzzy controller with energy buffer and energy forecast inputs was used to change the duty cycle of the sensor node.

Simulation of an arctic monitoring station using Matlab and Simulink was presented in [6]. For this station, multiple energy sources were available for harvest, including solar and wind. Data from meteorological databases were used and the simulation included energy harvesting from both solar and wind sources. Models were present for both of the energy harvesters, power conversion and routing, the battery, and an hourly load profile. The battery model included the effect of temperature, which is an important factor in arctic monitoring, as low temperatures reduce battery capacity. Simple power management was used in the simulation: the monitoring station was completely powered until the battery was depleted, at which point the station shut down.

Simulations of WSN for testing of an algorithm to maximize the networks data throughput were presented in [126]. Fading channels and Gaussian noise were considered for wireless transmission, as well as transmission duration and selected transmission power level. The presented method assumed perfect knowledge of harvested energy and of channel fading levels. Finite battery capacity was also considered.

4.1 Simulation Setup

Measured meteorological data from the WSU meteorological network were used as inputs for the simulated nodes. The layout of these nodes was selected to reflect the relative positions of the actual stations used (Fig. 2.2), which resulted in the layout depicted in Fig. 4.1. Solid lines connecting individual nodes represent communication channels. For the purposes of these simulations, it was assumed that a node in reception range of a transmitting node was available to receive transmissions unless it had already failed. Additionally, the base station was purely a message sink, taking no measurements and sending no transmissions. Communication between nodes was considered to be perfect. The amount of power for a transmission between nodes was constant and independent of the distance between nodes, meaning that fading channel coefficients and transmission durations were neglected. Messages received by a node were moved across the network in a store-and-forward fashion. During a node's scheduled transmission, it transmitted all of the messages it had collected, as well as the ones associated with measurements it took directly [127]. There was no consideration given to the direction a transmission was travelling (i.e., closer or further away from the base station), which allowed for a simpler implementation, as well as for potential usage of the distributed forecasts discussed in Chapter. 3. Using this scheme, it was expected that more connected nodes would experience greater energy usage than those with fewer connections. This is a very simple scheme with room for a great deal of improvement. However, more advanced techniques would be difficult to explore without simulating a larger network with more possible paths to the base station.

With respect to the energy reserve, which was assumed to be a primary battery for this simulation, the effects of temperature were neglected. The inclusion of these effects would likely have had a negative impact on the lifetime of the simulated nodes as both high and low temperatures limit battery capacity and discharge ability. The non-rechargeable nature of the primary battery also made the constraint on the amount of harvested energy more important, as it is lower than would be the case for a rechargeable battery. Self discharge of the energy buffer supercapacitor was also neglected.

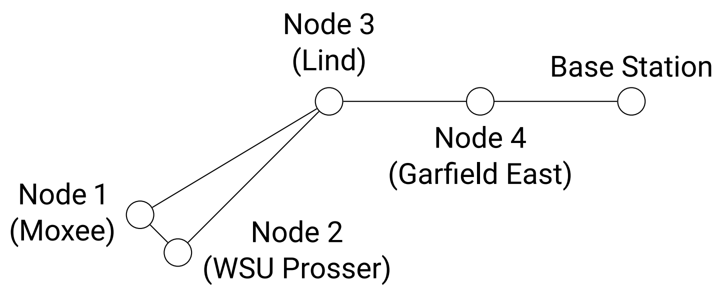


Figure 4.1: Layout of WSU nodes used in simulations: positions reflect relative locations of the actual stations used (see Sec. 2.6.2).

The various energy related parameters of the simulated nodes are shown in Table 4.1. Solar panel size and efficiency values were taken from [37,125]. The battery reserve capacity corresponded to a 75% drop in total battery energy capacity. In this simulation, the nonlinear characteristics of the solar panel were neglected and it was assumed that the solar energy striking the panel was linearly converted to electrical energy with the efficiency shown. Voltage conversion was also modelled linearly with the noted efficiency.

Table 4.1: Energy costs and efficiency parameters used in simulations.

Parameter	Value
Buffer Capacity	75 J
Battery Reserve Capacity	23085 J
Solar Panel Efficiency	22%
Solar Panel Size	648 mm^2
Energy Conversion Efficiency	0.80
Transmission Cost	0.20 J
Receive Cost	0.05 J
Measurement Cost	9.60 mJ
Memory Write Cost	0.03 J
Sleep Cost	56.67 μJ/s

An example value for energy used while making an atmospheric pressure measurement is approximately 2.7 mJ using an appropriate sensor with low energy requirements [116]. This value represents slightly less than one third of the energy used taking measurements in this simulation. The simulations in this thesis assumed that a few low power sensors are attached to each node and that there are no sensors with particularly high energy demands (e.g., nondispersive infrared gas concentration sensors [128]) within the network.

The energy cost associated with the evaluation of an energy forecast model was neglected. However, estimations of the energy used during the evaluation of the evolved fuzzy rule set for the prediction daily solar energy developed in [62] were presented in [129]. There, different combinations of microcontroller, operating frequency, compilers and optimization levels were tested and resulted in different power consumptions and times for forecast creation. Estimates of energy usage ranged between 5.4 μJ and 5817.6 μJ with the bulk of the tested configurations requiring less than 100μJ. With many of these energy estimates below the energy usage during sleep (Table 4.1), it was expected that omission this energy cost would not meaningfully impact the simulation results.

For the purpose of these simulations, the maximum and minimum number of operations were fixed. The time between measurements was allowed to vary between 60 and 3600 seconds (1 minute – 1 hour), while the time between transmissions varied between 120 and 86400 seconds (2 minutes – 1 day). Changes to the frequency of node operations were performed by scaling the number of operations on a per day basis. In these simulations, a node activity level N_A was output from the fuzzy controller and used for both operations, leading to a linear estimate of energy usage per day as opposed to using the node activity N_A to scale the time between node operations, illustrated in Fig. 4.2. The relationship between the number of operations O occurring in a 24 hour period and N_A is expressed using

$$O = N_A(O_{\text{Max}} - O_{\text{Min}}) + O_{\text{Min}}, \tag{4.1}$$

where O may be either measurement or transmission. Selection of O_{min} and O_{max}

could be used as a simple way to ensure that measurements of certain variables are made with enough frequency to maintain data quality. For the simulations here, two updates of node activities were performed, one at the sunrise of the current day and one at sunset. For cases where energy forecasts were used, new forecasts were not made as part of the sunset update; those made during the sunrise update are used, if appropriate.

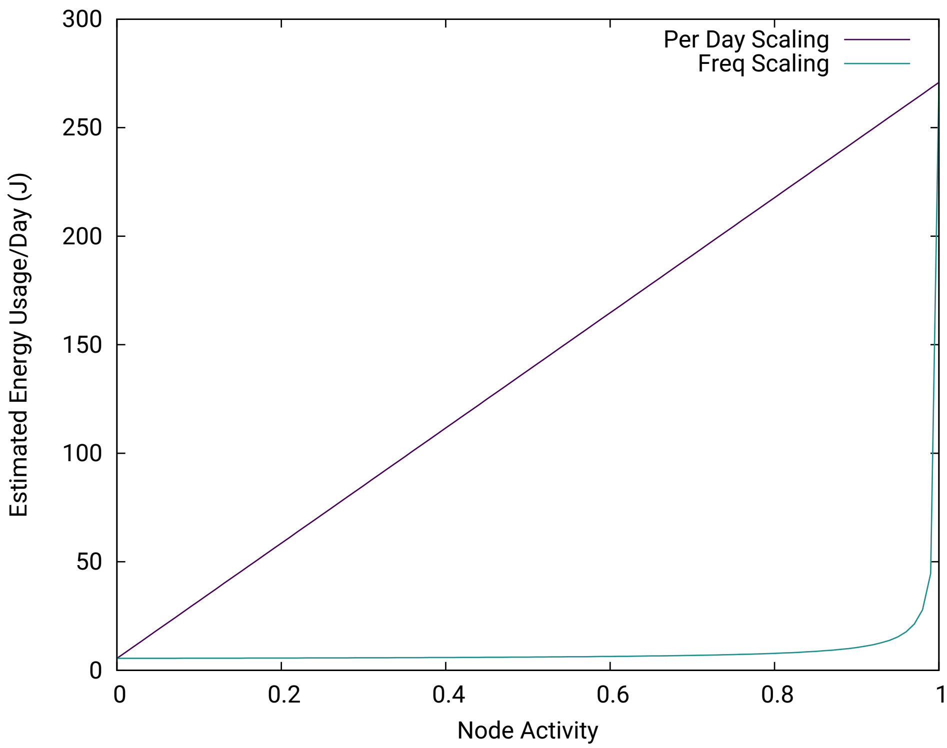


Figure 4.2: Estimated energy usage per day using different scaling methods.

With respect to the failure of a network, there are several potential definitions of a WSN's lifetime, including: 1) the time until the failure of the first node, 2) the time until a certain fraction of nodes fail, and 3) the time until coverage or connectivity constraints can no longer be met [25]. For the simulations presented here, the network was considered to have failed with the failure of any node, as the topology and size of the simulated network made the other definitions less meaningful. Failure of individual nodes due to insufficient energy reserves was considered permanent and nodes were given no chance of recovery. In reality, the energy buffer could potentially become charged again, allowing a node to resume some level of operation. However, as this seemed to be unnecessary using the created controllers, this scenario was not considered. While they are run to completion, for the purposes of optimizations, simulations where nodes fail were heavily penalized through the large amount of energy reserve E_R used, as well as the high amount of lost energy E_L.

4.2 Simulation of Constant Node Activity Levels

Using the described setup, simulations were performed using constant node activity levels in order to examine network performance where no control was used. A number of simulations were performed with constant N_A values ranging from 0 to $\sim$0.65. For higher node activities, reserve energy was used such that node failure would occur prior to the end of the simulated period, thus causing network failure (i.e., node activities of 0.65 and greater cause Node 3 to fail before the end of the simulation). Fig. 4.3 shows the total network measurements taken during the simulation. Fig. 4.4 shows the relationship between energy not collected or used by the nodes and the activity level. Fig. 4.5 shows the total network energy reserve usages.

While the total number of network measurements and lost energy were roughly linear with respect to node activity, the energy reserve usage rose very rapidly for activities above 0.3. Examining the instances when the reserve energy was spent showed that it began during the daylight hours. Overnight, the energy buffer was depleted and the incoming solar energy was initially insufficient to support the activity level. The energy buffer was replenished during the day and the cycle began again. As N_A further increased, reserve energy was used during both day and night hours, as the buffer was more quickly depleted overnight. This behaviour is illustrated in Fig. 4.6.

The minimum amount of energy reserve usage that could be experienced for a

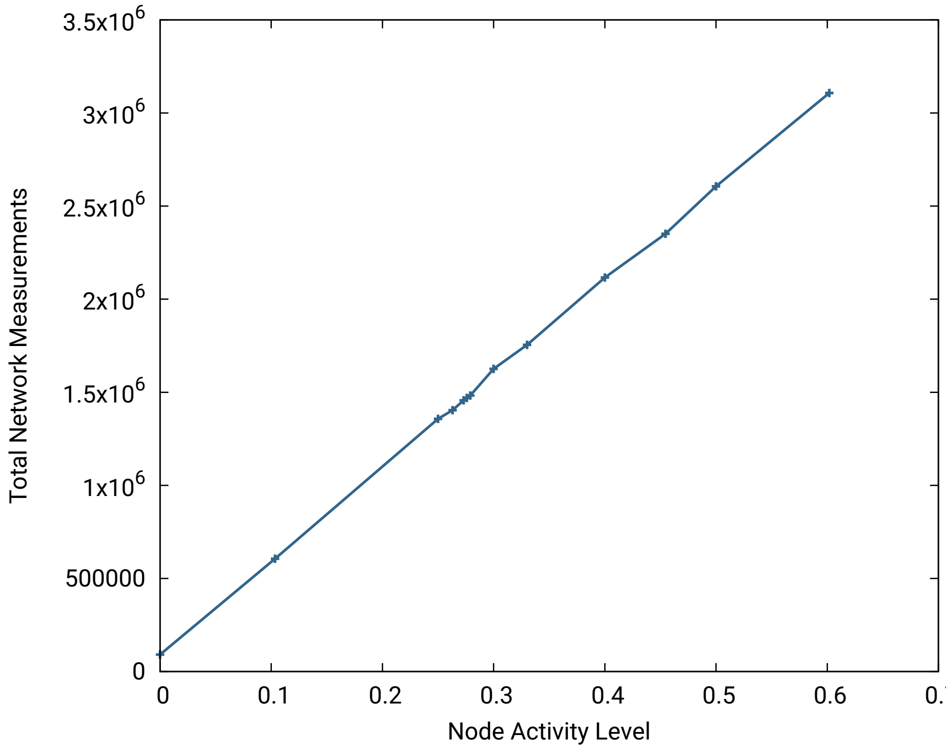


Figure 4.3: Total simulated network measurements for constant node activity levels.

simulation was 11.69 J, the bulk of which was used nearly immediately, as the activity level used by the nodes prior to the first update was too high. The highest node activity that used only this amount reserve energy is roughly 0.276, after which the network began experiencing more energy reserve usage, followed by node failures for activity values greater than $\sim$0.6. For the simulation using the 0.276 activity level, the network took a total of 1,469,391 measurements and experienced 5.11 MJ of lost energy. This activity level was considered the baseline for future simulations, as more effective use of energy would result in less lost energy and a greater number of measurements. Network energy usage for each day, broken down by daylight and nighttime hours, for constant node activities of 0.6019 is shown in Fig. 4.7. No node failed during this simulation.

A linear regression of the plotted total network measurements M_T related to the

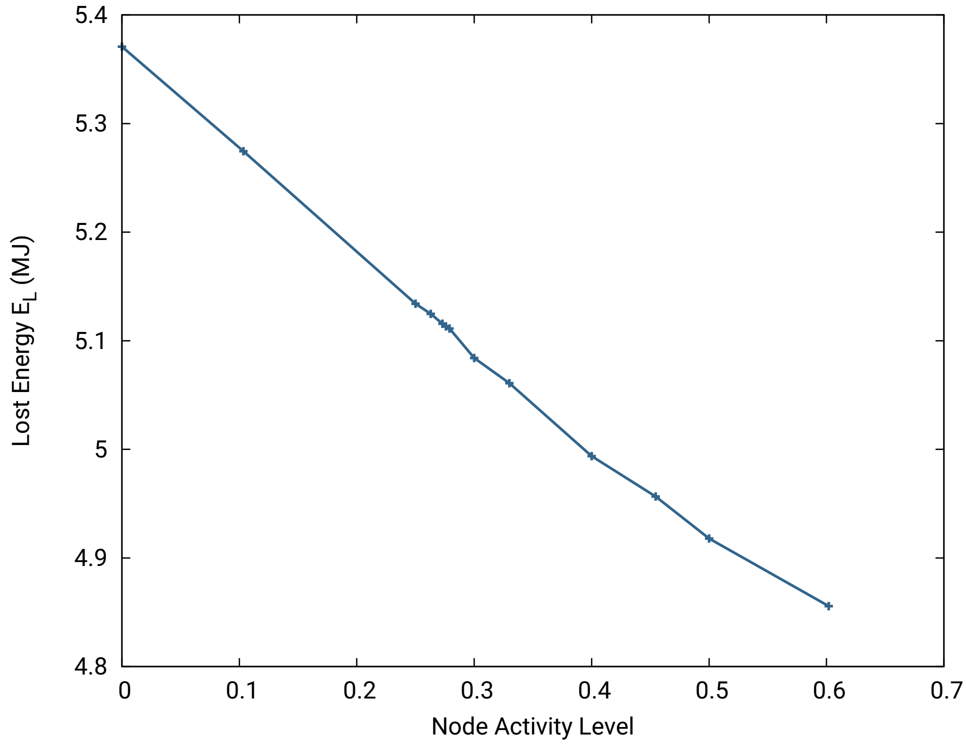


Figure 4.4: Total simulated network lost energy for constant node activity levels.

node activity level resulted in the equation:

$$M_T = 5024012.62529481 N_A + 91182.84914123. \tag{4.2}$$

This equation can be rearranged in order to calculate an estimate N_A using the total number of measurements obtained during a simulation. For simulations where a constant value of N_A was not used, the calculated estimated represented the equivalent value that a network would have to use to obtain the same number of measurements during simulation. For controllers when node activities were changed, comparing the calculated value with values of constant activities provided a method to evaluate controllers with respect to reserve energy usage.

Table 4.2 shows the results for a simulation where the nodes were forced to adopt a constant activity level of zero. In this case, the total number of measurements represented the absolute minimum that each node could take during the sim-

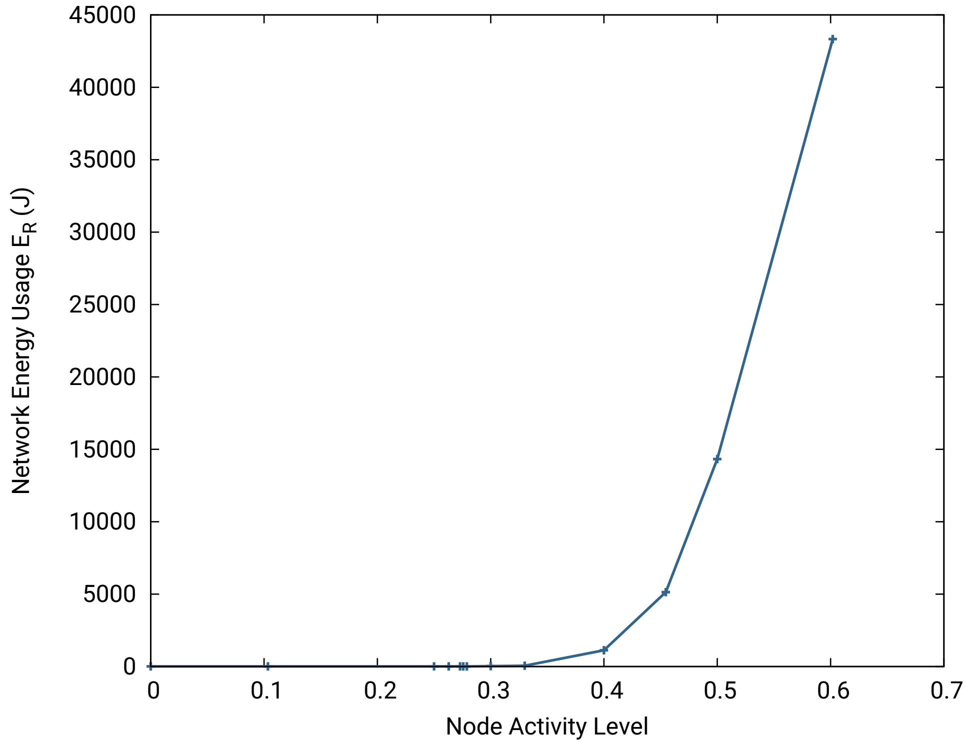


Figure 4.5: Total simulated network energy reserve usage for constant node activity levels.

ulation. Across the network, this equates to 7816 measurements for each joule of reserve energy usage.

Table 4.2: Simulation results with forced zero node activity.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	1.44	22823	25	27	26
2	1.30	22826	25	27	26
3	8.94	22823	25	27	26
4	0	22824	25	27	26
Tot/Comb	11.69	91296	25	903	26

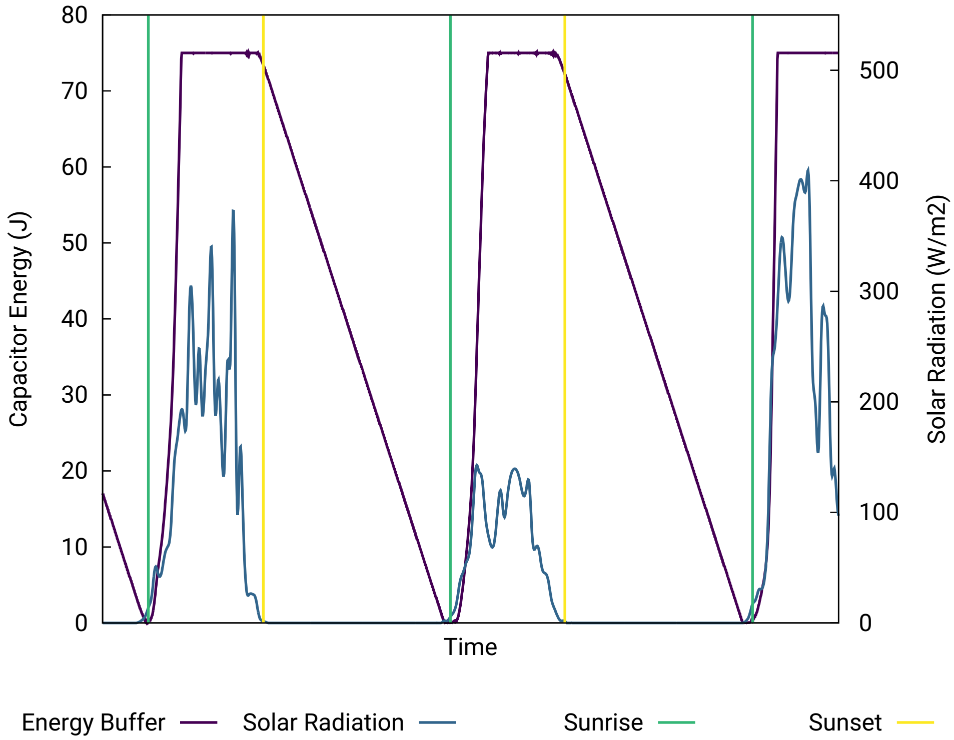


Figure 4.6: Time series of incoming solar radiation and energy buffer levels for Node 3 using a constant node activity of 0.4000 during days of low solar availability.

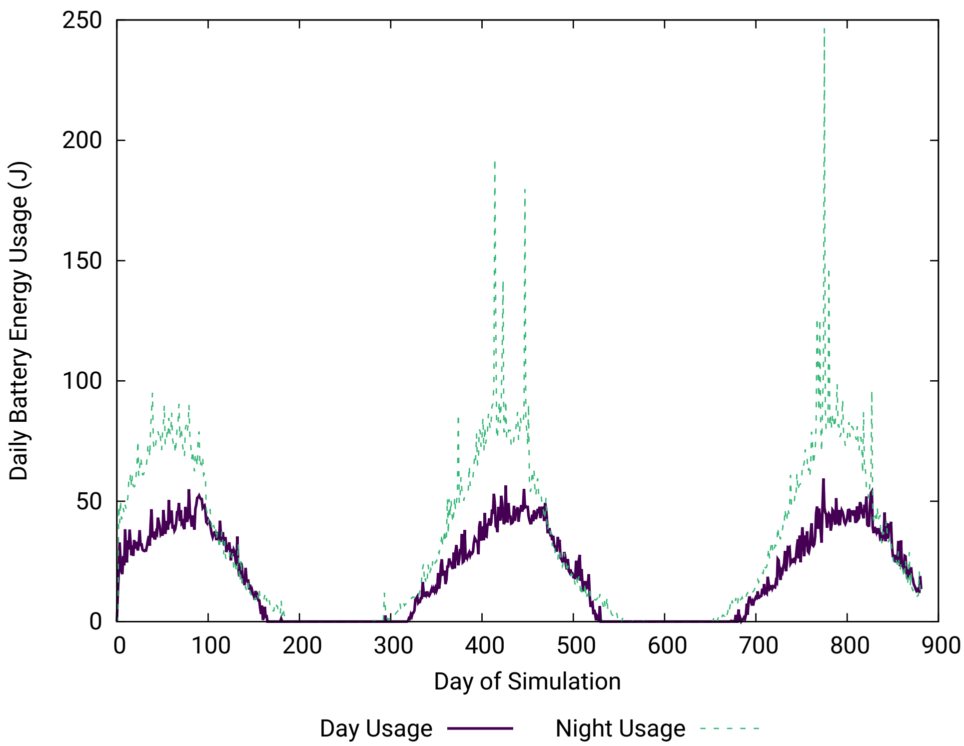


Figure 4.7: Total simulated network energy reserve usage for ∼0.6 constant node activity.

4.3 Reference Controller and Simulation

A human-created controller was used for an initial simulation of the WSN and was based on the procedure in [1]. Five triangular membership functions were used for each input, while 5 singletons were used for the output. The membership functions associated with the energy buffer were all equally sized, whereas the membership functions for the input associated with the energy forecast were sized such that each function covered approximately 20% of the daily energy values for the Fairview location. For each of these inputs, the five membership functions were denoted very low VL, low L, medium M, high H and very high VH. The uneven partitioning of the energy forecast input helps to activate the rules with a more even frequency.

The fuzzy inputs for the energy buffer E_B, energy forecast E_F, and output node activity N_A are shown in Fig. 4.8. The fuzzy rule base used in [1] is shown in Table 4.3. Initially, the simulations performed in [130] used a different rule base, as the one presented here led to node failures during the simulation. Further work revealed that a problem existed in the estimation of sunrise and sunset times due to mishandling of daylight savings time. The rule base used in this case was generally shifted towards medium node activity, with fewer instances of the very low node activity and only one instance of very high node activity. Comparing the corrected and uncorrected perfect forecast performances using the same metrics as in [130], the corrected version had higher total and minimum values for M_D, but lower mean and maximum value for this metric. The corrected version also used less battery reserve energy.

The performance of this controller was examined for the case where no forecast was used, and assumed an incoming energy amount corresponding to individual labels of E_F. The VH ratio amounted to using the value of $\hat{E}_A$ as the forecasted value. Five simulations were run, resulting in the values tabulated in Table A.1. The differences between the amount of energy used and the number of total measurements taken point to static energy ratios resulting in measurements being taken at times

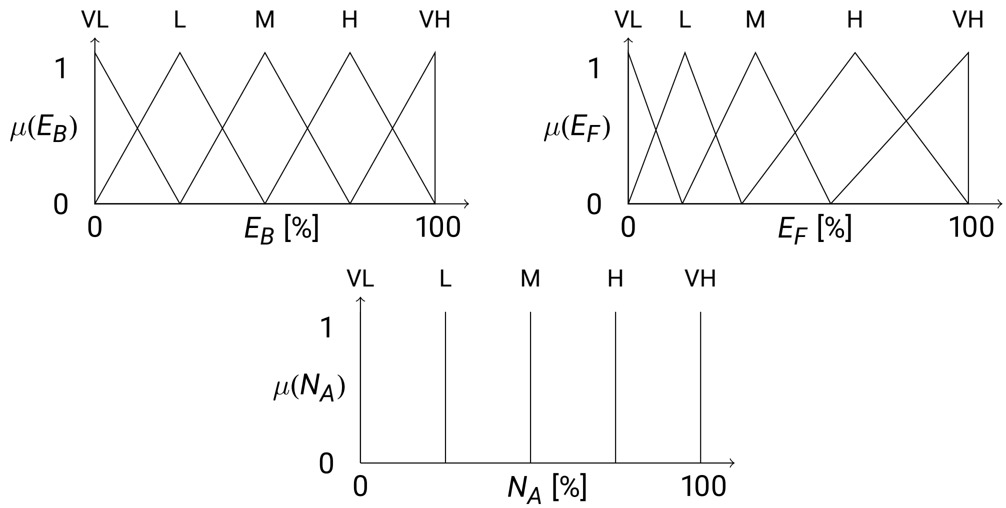


Figure 4.8: Fuzzy input and output partitions for reference controller.

Table 4.3: Fuzzy rule base of the energy management controller from [1].

		E_B				
		VL	L	M	H	VH
	VL	VL	VL	VL	L	M
	L	VL	VL	L	M	H
E_F	M	VL	L	M	H	VH
	H	L	M	H	VH	VH
	VH	M	H	VH	VH	VH

when harvestable energy was not available.

Simulating this controller in the WSN using a perfect energy forecast resulted in the energy buffer level shown in Fig. 4.9, the energy reserve levels shown in Fig. 4.10, and the plot of the measurements per day shown in Fig. 4.11. Tabulated values are presented in Table 4.4. The battery energy plot shows that the central location and simple message transmission scheme of Node 3 caused it to use the most energy, with the bulk usage occurring in the winter. Nodes 1 and 2 experienced similar, but lower energy usage. The energy buffer plot shows the degree to which the energy deficit was present. Compared to the results where the static energy ratios were used, the total number of collected measurements are between the amounts collected used the M and H forecasts, but only used roughly 36% of the battery reserve energies of the M simulation.

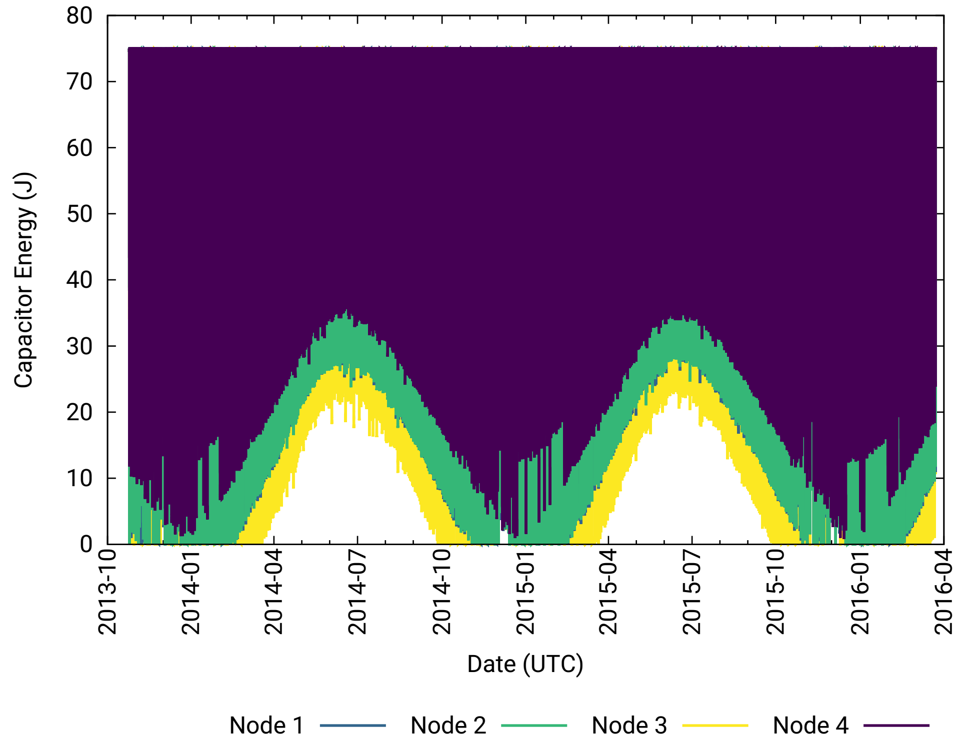


Figure 4.9: Capacitor energies for reference controller during a simulation where a perfect forecast was used (Node 1 values concealed by Node 2 values due to their similarity).

Table 4.4: Reference controller results, perfect forecast.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	770.14	598240	381	1081	680
2	747.13	603435	389	1091	686
3	2129.69	569548	386	1024	647
4	0.00	620424	395	1162	705
Tot/Comb	3646.96	2391647	381	1162	679

The outputs of the fuzzy controller are shown in Figs. 4.12 and 4.13 for the sunrise and sunset updates respectively. The high degree of oscillation during the winter months indicates that the controller selected an activity that was high enough that the amount of energy was lowered to the point where it could not be replenished during daylight hours. The sunset updates had a much lower variance, primarily re-

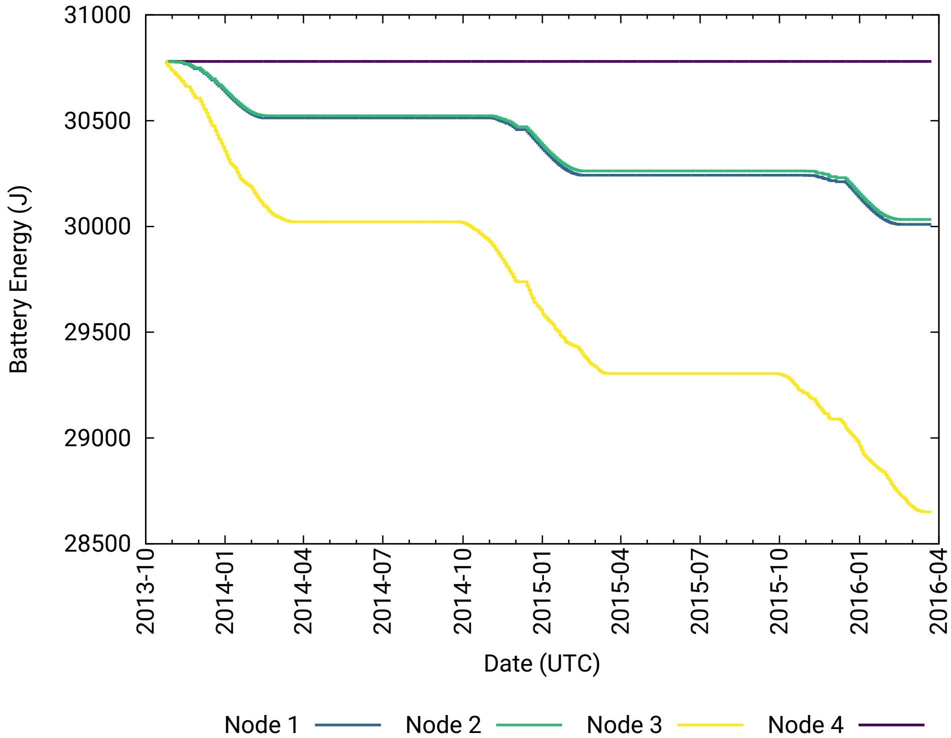


Figure 4.10: Reserve battery energies for reference controller during a simulation where a perfect forecast was used.

maining in the area of 50% node activity.

Using the line fit to the total network measurements of the constant node activity simulations, the simulation of the reference controller using a perfect energy forecast resulted in an effective node activity of approximately 0.4580. Fig. 4.14 shows the effective node activities for the reference controller using static and perfect forecasts with the constant N_A results. In this figure the VL, L, and M static forecasts lie to the left of the line of constant node activities, representing that these simulations could have been outperformed by a constant node activity. As the H and VL static forecasts lie to the right of the line, a higher number of overall measurements was taken using less reserve energy than an equivalent constant value. The M, H, and VH static forecasts all used similar amounts of energy, while having varying node activities, which was due to a very consistent node activity during for sunset updates (when the bulk of the energy reserve was spent) with a more variable node

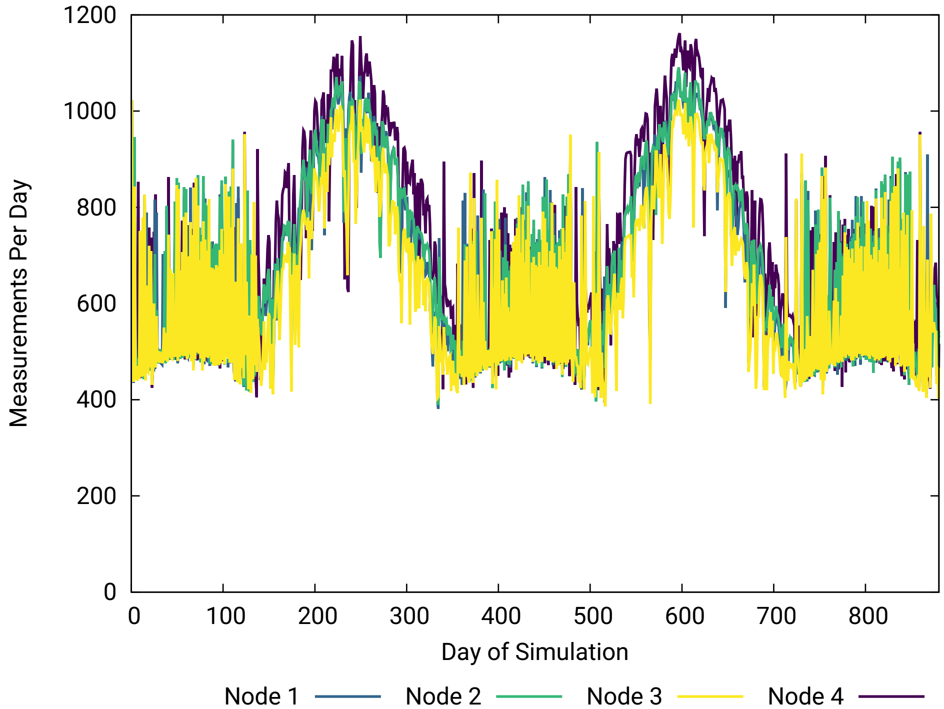


Figure 4.11: Number of measurements per day for reference controller.

activity for sunrise updates. The value for the perfect forecast also lies to the right of the constant node activity values, but had both lower energy usage and effective node activity.

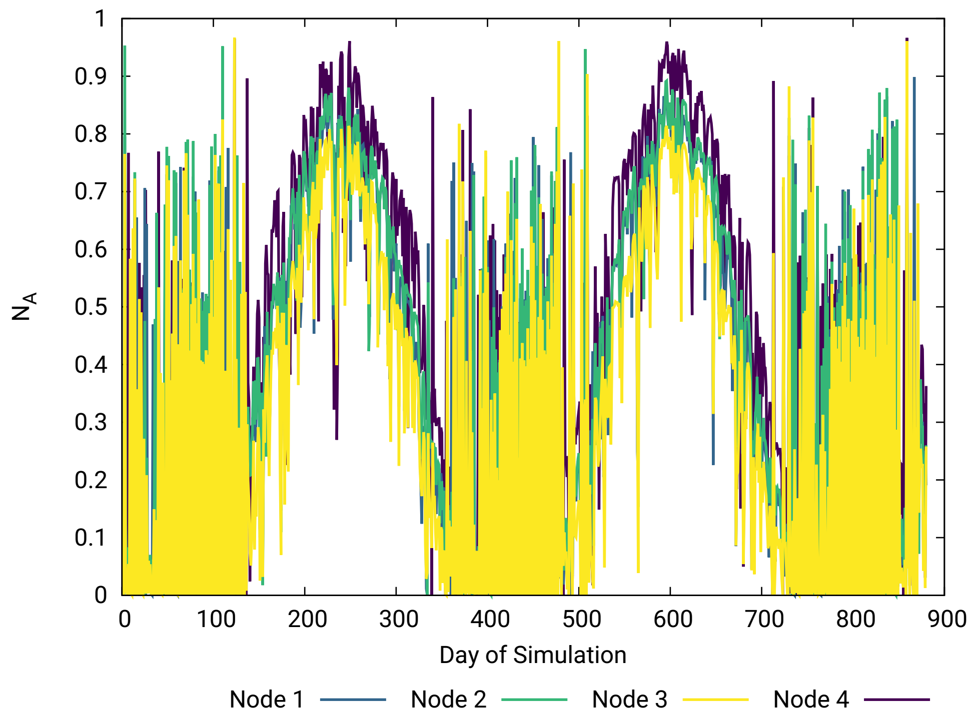


Figure 4.12: Reference controller node activity outputs for sunrise updates.

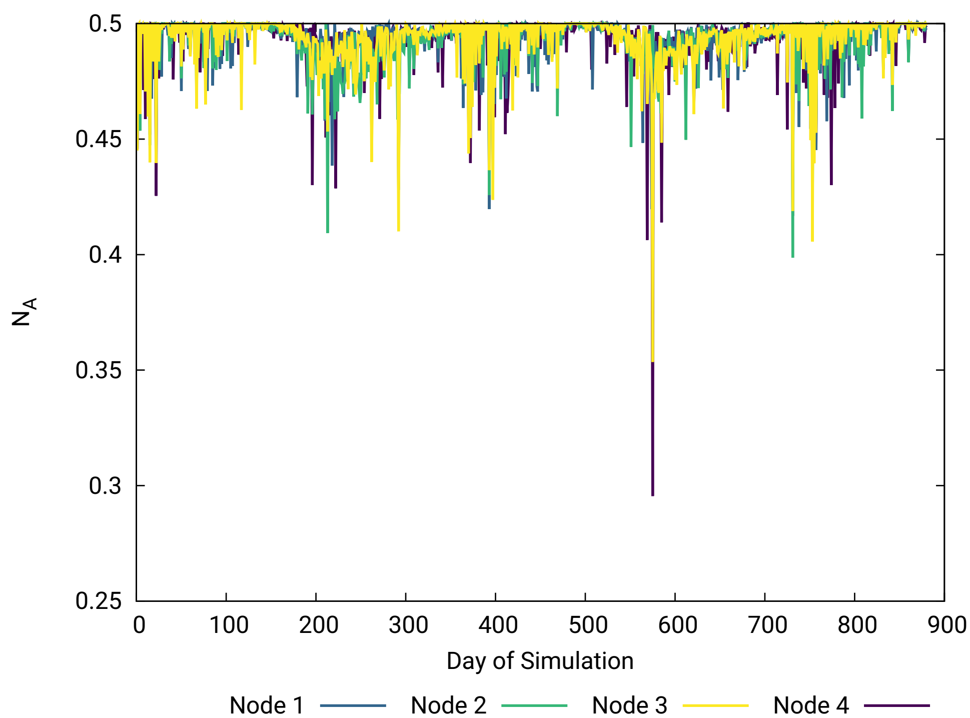


Figure 4.13: Reference controller node activity outputs for sunset updates.

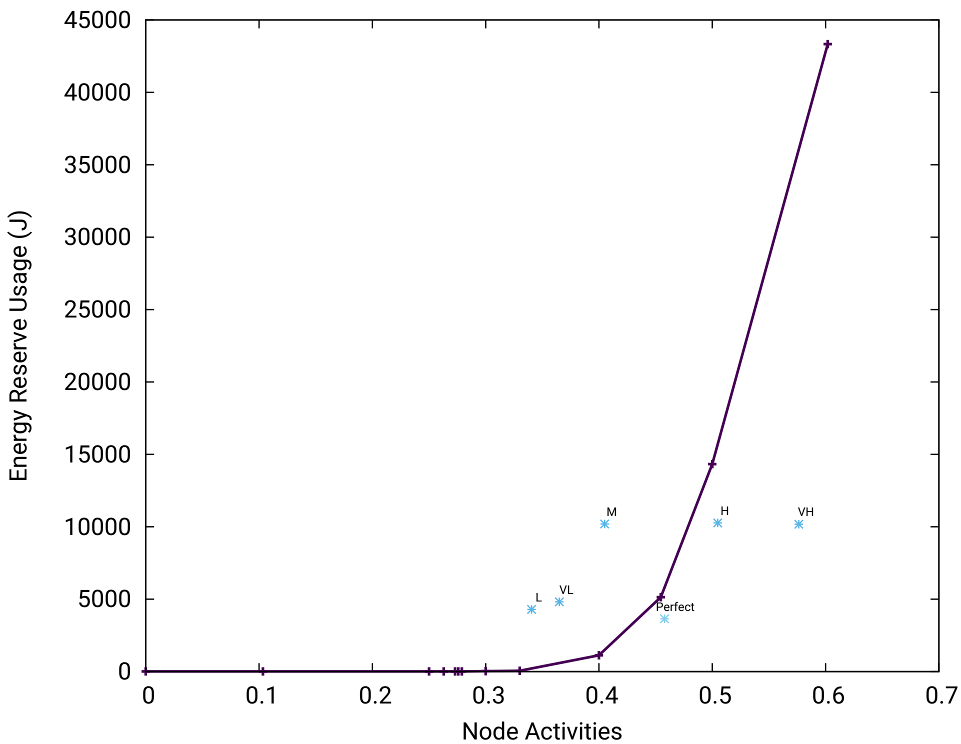


Figure 4.14: Comparison of reference controller effective constant node activities and constant node activities with respect to energy reserve usage.

4.4 Optimization

In order to improve performance of the fuzzy controller, a method of optimizing the shapes of the membership functions was desired. Not only could there be gains in performance realized by tuning the shapes and locations of the membership functions, but as the forecast horizon is extended, the interaction between these predicted values and the energy buffer state become more complicated and difficult to determine.

In order to perform this optimization, first a method of evaluating potential solutions was created. The evaluation metric presented in Sec. 4.4.1 included the amount of battery reserve a node used during the simulated period, the amount of energy available for harvest that was unharvested because of insufficient storage capacity and terms related to the semantic soundness of the fuzzy sets. The inclusion of both energy reserve usage and missed harvesting opportunities was intended to result in a balance between the potential deployment length as indicated by reserve levels and node activity level, which was assumed to increase when the most harvestable energy was used.

The candidate controllers presented in this work are given the shorthand notation of CC#, where # is the unique number given to that controller. The exception is the reference controller, which has just been called Reference. For simulations using different controllers, the results are associated with the identifier CC#/XXXX, where XXXX is the identifier of the forecast model used for the given simulation.

4.4.1 Solution Creation and Evaluation

For the generation of population members, the input and output variables were randomly partitioned with a set number of membership functions. For inputs, trapezoidal membership functions were used. The output was partitioned using fuzzy singletons, with the number of singletons such that every combination of input membership functions had their own. This made the automatic creation of the rules relating the inputs to the outputs simple.

After the optimization was completed, the resulting fuzzy controller could undergo a reduction for cases where output singletons very close together in the output space. These values could be combined and the rule base similarly reduced. The number of input sets could also be combined, but the amount of reduction possible would generally be smaller as the fitness function included terms to prefer membership functions with low to moderate levels of overlap.

The fitness values for each candidate solution was calculated using the function:

$$f = aE_R + bE_L + c(S_p + S_c), \tag{4.3}$$

where E_R represents node energy reserve usage, E_L represents energy present in the environment but not harvested or used by the node, S_p and S_c are values corresponding to possibility and coverage of the candidate fuzzy sets. S_p is calculated by applying Eqn. 2.23 to each pair of membership functions in a fuzzy set, summing the result and dividing by the total number of pairs. Values for Eqn. 2.23 less than 0.25 were not included in the sum in order to allow membership function overlap without penalty. S_c is calculated by finding the total number of points in each fuzzy set with no activated membership function and dividing by the total number of points. Scaling constants a, b, and c have been included in order to weight the relative importance of each of these terms. Lower values of this fitness function represented a better performing and more transparent control scheme.

The inclusion and relative weighting of the E_R term placed a high importance on energy neutral node operation, while the E_L term incentivized the control system to make the best use of the energy present in the environment. The relative importance of E_L and E_R in the fitness function depended on a number of factors, including the length of desired deployment, the expected amount of energy available for harvest, and the value of individual measurements. For this particular case, b was set to 1×10^{-5} since the typical amount of uncaptured and unused energy from previous simulations was on the order of 1 MJ. To allow battery usage to have a high weight, a was set to 1. To encourage meaningful fuzzy sets, c was set to a value of 5.

A downside of this formula for evaluation of solutions is that it relies on the amount of environmental energy used as an analogous value to the number of measurements taken and transmissions made. For cases where the energy reserve needs to be preserved at all costs, the coefficient b would be set to 0 and a would be set to a non-zero value. Here, the E_R value would be the most important, but there would no longer be any incentive for the controller to take any measurements, and constant operation at the lowest permissible activity level would likely be the best performing solution. To address this, terms could be introduced relating to frequency of node operations, but for more complex controllers (e.g., if measurement and transmission rates were no longer linked) the weighting of relative operations would also need to be included, increasing the complexity of relative weighting.

4.4.1.1 Unharvested Energy

The inclusion of the term corresponding to the unharvested or 'lost' energy , E_L, was intended to produce controllers that would operate at the highest activity level that can be supported by the available environmental energy. In order for this value to be lower, energy must either be collected and stored in the energy buffer, or directly used in node operation. For very large buffer sizes, there would be very little lost energy, since the solar energy could be stored. In order to continue keeping the value low, node operations should occur at a frequency such that the buffer is depleted enough to the point where incoming solar energy has a place to be stored. For small buffer sizes, this variable should behave in the same way, but the optimization is much more difficult, as depleting the buffer to the point where all the solar energy may be stored creates the situation where the energy reserve is likely to be needed to keep node operations at the minimum required level during times of reduced harvesting opportunity.

4.4.2 Initial Optimizations

The initial simulations explored the optimization of a fuzzy controller similar to the reference controller, as well as added an additional day of forecasting horizon. These initial optimizations limited the output activity level N_A to be between 0 and 1, simulate with the reference controller.

The initial optimization was presented in [131] and used five trapezoidal membership functions for both the forecasted incoming energy and the status of the energy buffer, resulting in 40 input parameters. Five membership functions were used to mirror the five (VL, L, M, H and VH) membership functions used in the reference controller (Sec. 4.3). Each possible combination of input membership functions was given an output singleton, which resulted in a single output variable with 25 singletons, for a total of 65 total parameters.

Trapezoidal membership functions were selected for use in the fuzzy controller in order to allow for the resulting membership functions to be non-symmetrical. Additionally, these functions could be collapsed down to triangular or singleton membership functions during the optimization.

For this initial test, DE using the *DE/rand/1/bin* variant was used. The population consisted of 250 members. A scaling factor of $F = 0.95$ and a crossover probability of $C = 0.9$ were used. Five optimization trials were run, finishing between 235 and 421 generations. The trial that finished in the smallest number of generations stagnated in a higher local minima than the others. Three other trials finished with similar solutions, while one found a much better solution, pointing to possible improvements in the optimization method.

The results using the best solution, CC1, are presented here. Equivalent membership functions were present in the resulting sets, allowing for a simple and straightforward reduction to be performed. This reduction resulted in four membership functions for the forecast input and three functions for the energy buffer input. The total number of output singletons, as well as the number of rules was reduced from

25 to 11. The resulting input fuzzy sets for the forecast and energy buffer level, and the output singletons are shown in Fig. 4.15. For the forecast and buffer inputs, set3 and set2, respectively, are singletons at 100%. As expected, the resulting output space still contained singletons that were very close together, pointing to the opportunity to further reduce the size of the rule base. A matrix representation of the rule base is given in Table 4.5. While interpretation of the input and output memberships functions is straightforward, the relationships between them presented in the table are harder to understand. One possible reason is that the simulations used for the optimization were performed using sequential weather data, allowing for the possibility of any recurring trends to be exploited. Another possible explanation is that the low frequency of occurrence for the extremes of the forecasting range, lead to few training examples for such cases.

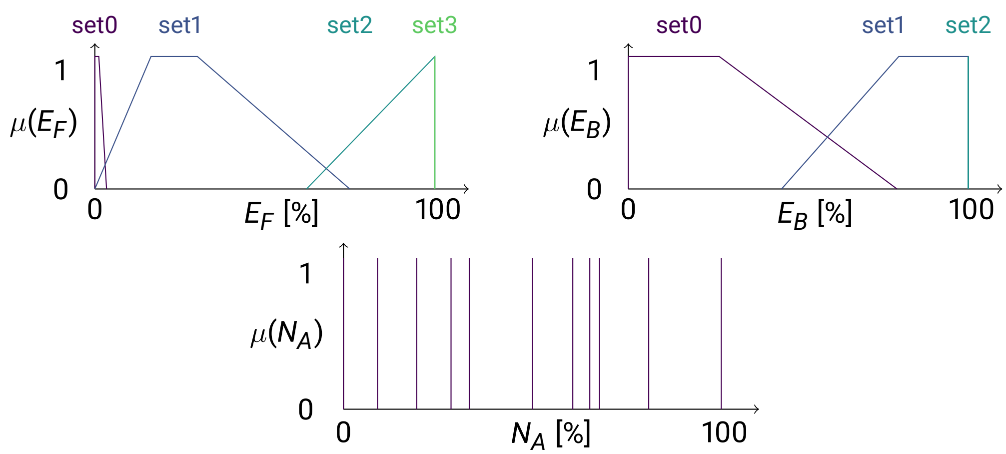


Figure 4.15: Fuzzy input (energy forecast, top left; energy buffer, top right) and output (node activity, bottom) partitions for CC1.

Table 4.5: Matrix representation of CC1.

		E_B		
		set0	set1	set2
E_F	set0	0.090	0.285	0.333
	set1	1.000	0.000	0.652
	set2	0.678	1.000	0.607
	set3	0.808	0.500	0.194

Using a perfect forecast, this evolved controller (CC1) collected 121% of the total measurements taken by the reference controller while using 2.26% of the reserve energy, across the small network.

As it received messages from only one node, Node 4's energy usage had the smallest dependence on overall network activity. Conversely, because of its central location and reduced harvesting opportunities relative to the other nodes during the winter months, Node 3 had the highest dependence on overall network activity. Fig. 4.16 and Fig. 4.17 show the battery usage and the number of daily measurements (M_D) for this controller. A summary of the results for all nodes is shown in Table 4.6. Compared to the reference using the same forecast, not only did the evolved controller use less energy overall, but the reduction in energy usage was very large for Nodes 1 and 2.

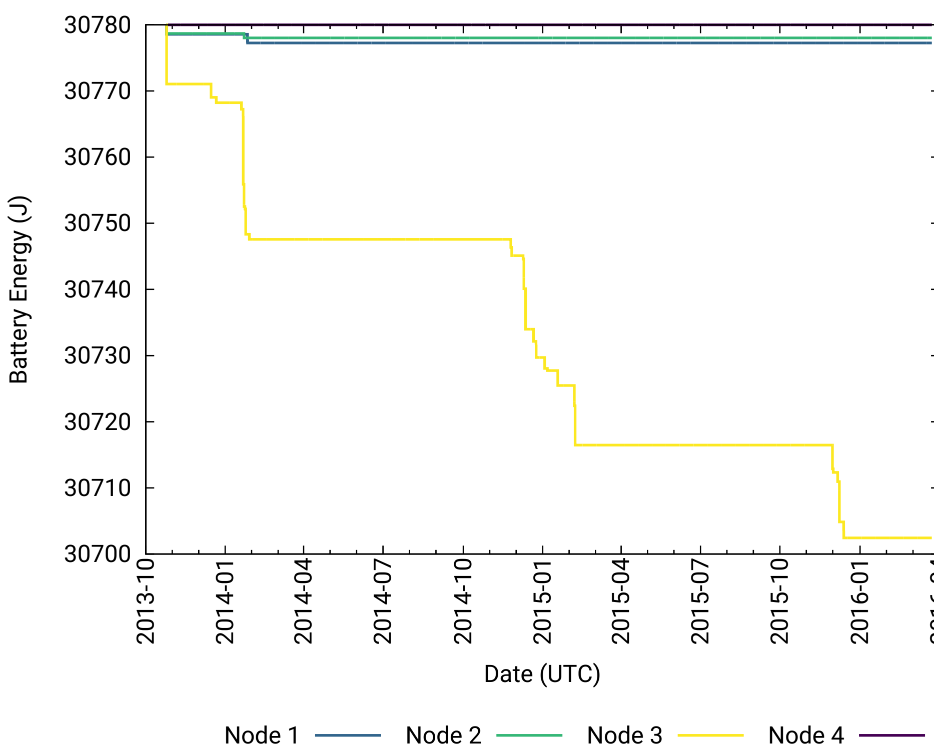


Figure 4.16: Battery backup energies for CC1 controller using a perfect forecast.

The overlap present in the five fuzzy sets motivated a reduction in the number of

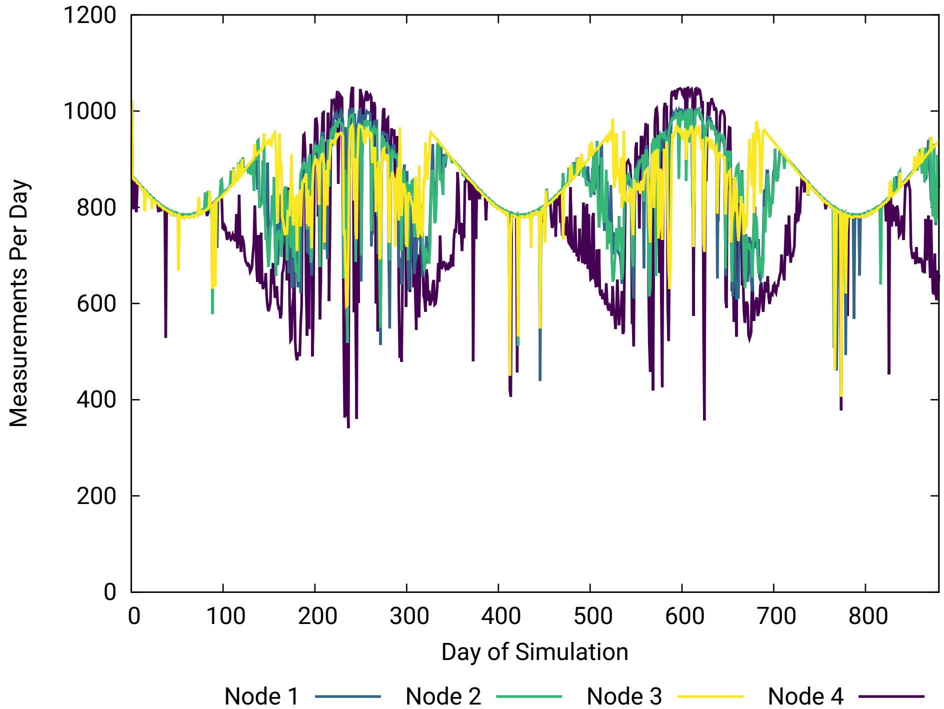


Figure 4.17: Number of measurements per day for CC1 controller using a perfect forecast.

sets used for the input to to 3 membership functions per input set. Also investigated at this point was the increased forecasting horizon [132]. Using an adaptive method of applying DE, WDE (Sec. 2.5.4), two controllers were optimized, one using only the current day (D_0) solar energy (CC2) and one using current and next-day (D_0/D_1) forecasts (CC3). Three trapezoidal membership functions were used for each input and each combination of input memberships were provided with an output singleton.

The resulting fuzzy sets for the single forecast case are shown in Fig. 4.18. Results of the optimization where the controller is provided with current and next-day energy forecast are shown in Fig. 4.19. In both cases, the close proximity of singletons in the output partition may have allowed for reduction in the number of rules. The controller using both forecasts (CC3) optimized to a lower fitness value compared to the controller using only one forecast (CC2) (11.65 vs. 12.29).

A summary of the results for the perfect energy forecasts for current-day fore-

Table 4.6: Evolved controller results, perfect forecast CC1

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	2.75	732579	439	1005	832
2	1.98	737761	508	1008	838
3	77.57	747532	406	984	849
4	0.00	676342	341	1051	769
Tot/Comb	82.30	2894214	341	1051	822

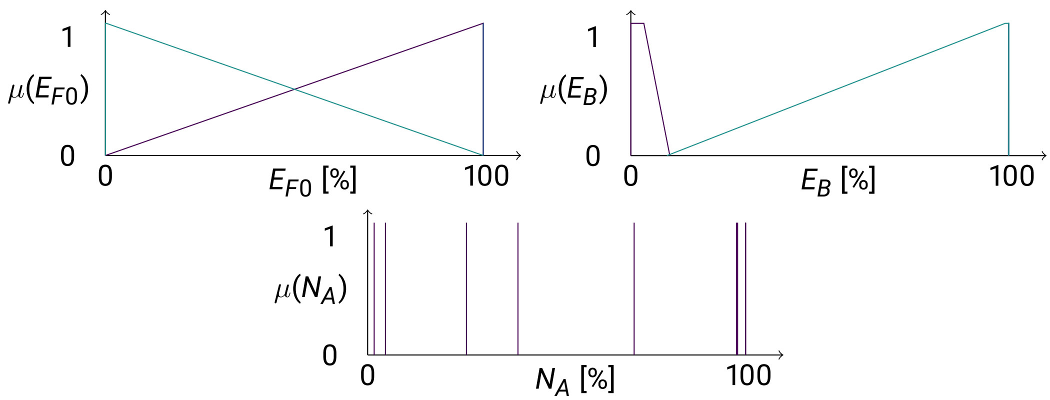


Figure 4.18: Fuzzy input (energy forecast, top right; energy buffer, top left) and output (node activity, bottom) partitions for the CC2 D_0 controller.

cast is shown in Table 4.7, while results for the current and next-day forecast are tabulated in Table 4.8. Using both energy forecasts, the CC3 controller used 161.1% of the reserve energy of the D_0 forecast while taking 125% of the total network measurements. Fig. 4.23 shows the number of measurements per day for Nodes 3 and 4 for the perfect forecast simulation. These nodes were chosen because Node 4 has the most stable energy usage (only receiving messages from one other node) and Node 3 had the most variable and highest energy usage (as it receives messages from three other nodes). For these simulations, the CC3 D_0/D_1 energy controller provided a smoother transition of measurements per day (partially attributed to the higher number of output singletons), while using more energy during certain times of year, especially for Node 3. This increase in Node 3 energy usage was not unexpected, since as the network became more active, it placed a greater burden on this node while there may not have been sufficient harvest opportunity to support the increased traffic. Additionally, the increase in energy usage was expected to be higher

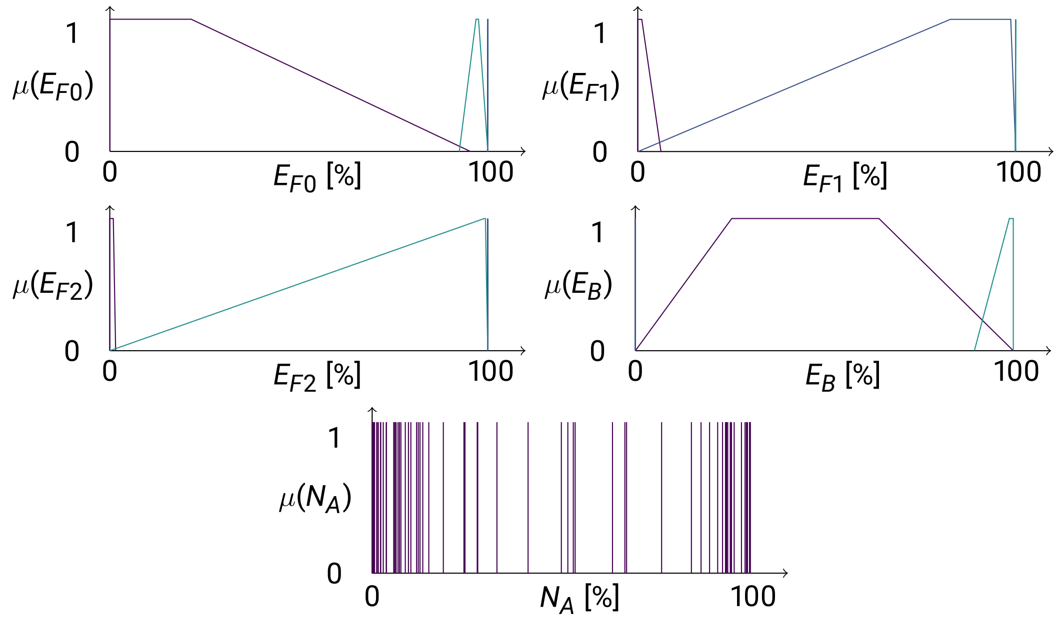


Figure 4.19: Fuzzy input (energy forecasts, top and middle left; energy buffer, middle right) and output (node activity, bottom) partitions for the CC3 D_0/D_1 controller.

than the increase in measurements taken since the higher activity associated with a greater number of measurements also increases the number of transmissions. As nodes in this simulation must receive messages sent to them and pay the associated energy cost, a modest increase in the number of transmissions can greatly raise the total energy usage.

Using only the D_0 forecast, the controller CC2 takes 115% of the measurements of the reference controller, while using 11.00% of the total energy reserve across the network. Comparing the D_0/D_1 controller CC3 to the reference controller, 17.75% of the reserve energy was used while 144% of the total network measurements were taken. Plotting the differences between the reference controller and CC2 results in Fig. 4.20 while a similar plot comparing the reference controller and CC3 is shown in Fig. 4.21. For these plots, values greater than zero represent less energy usage than the reference controller, while values less than zero represent more energy usage compared to the reference controller. Both controllers show improvement for Nodes 1, 2 and 3, with slightly more energy used for Node 4.

Plotting the difference in battery usages of CC3 with respect to the CC2 con-

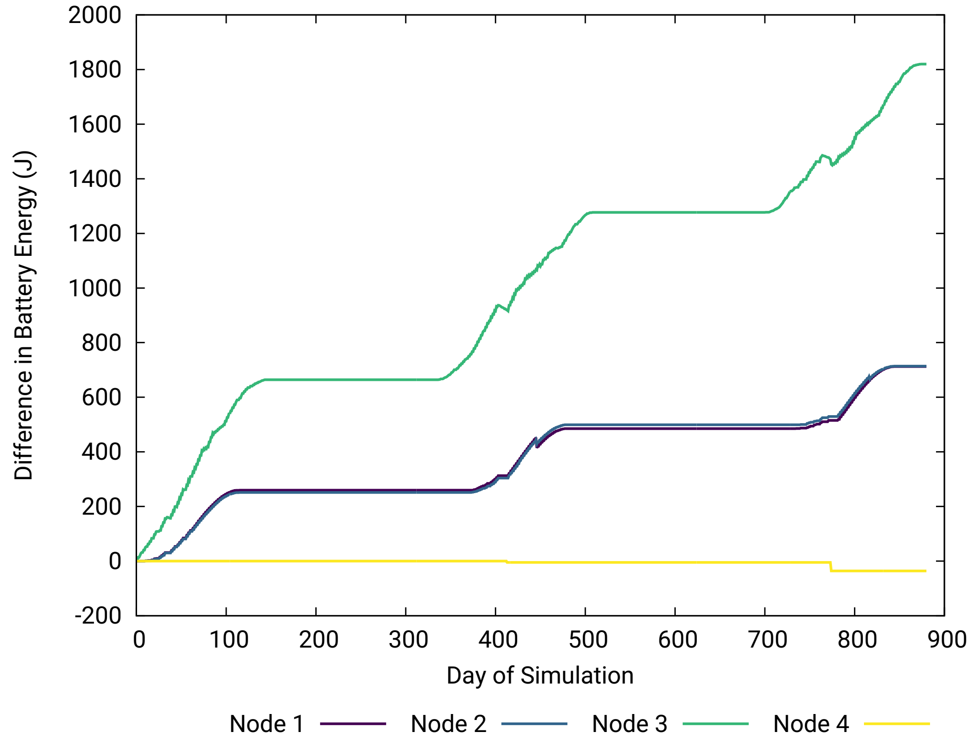


Figure 4.20: Difference between battery backup energies for reference and CC2 controllers where a perfect forecast was used. Values greater than zero represent less energy usage than the reference controller.

troller perfect forecast simulations results in Fig. 4.22. Here we see that the primary contribution of CC3's higher energy usage compared to CC2 is due to the usage of Node 3 during winter. However, the CC3 controller also took more measurements during the simulation, meaning that this energy was not spent uselessly.

The outputs of the fuzzy controller for the single day case are shown in Figs. 4.24 and 4.24 for sunrise and sunset updates, respectively. The two day case node activ-

Table 4.7: CC2 D_0 evolved controller results, perfect forecast.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	40.06	704370	424	1139	800
2	19.53	707452	396	1137	804
3	310.11	631803	396	1145	718
4	31.98	716118	415	1137	814
Tot/Comb	401.68	2759743	396	1145	784

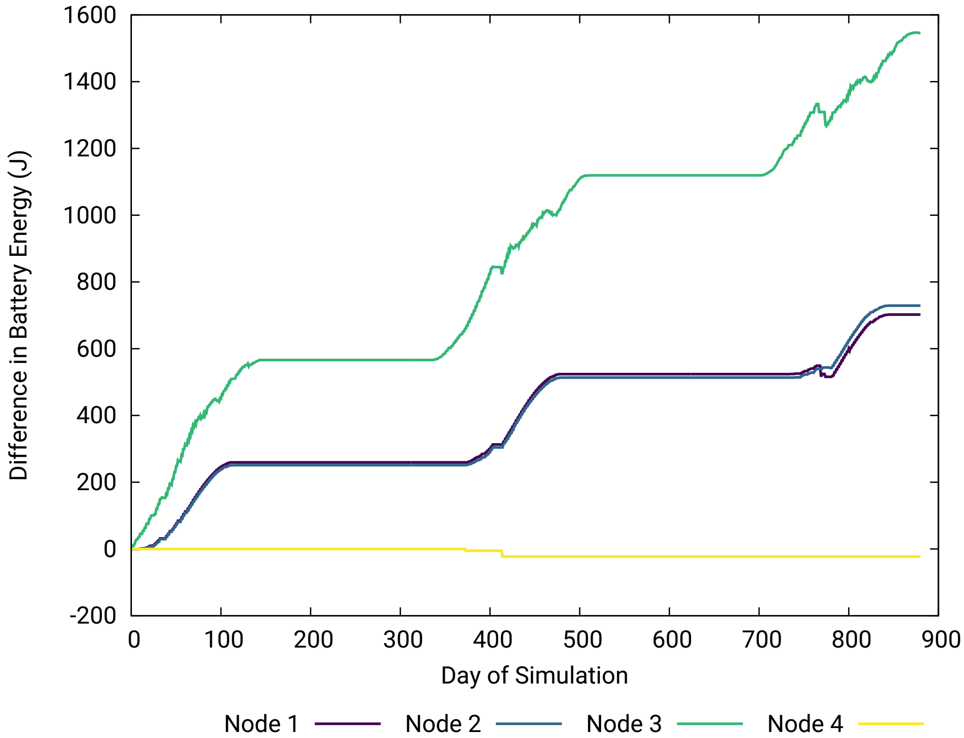


Figure 4.21: Difference between battery backup energies for reference and CC3 controllers where a perfect forecast was used. Values greater than zero represent less energy usage than the reference controller.

ities are shown in Figs 4.26 and 4.27. This illustrates the difference when the additional of forecasting horizon was provided. In the case where one day was provided, the controller favoured a relatively constant activity level during the sunset update while having a much greater variety of different activities being selected during the sunrise update. With the 2-day controller, the sunrise activity was consistently very high, while the sunset update had more variability.

Table 4.8: CC3 D_0/D_1 evolved controller results, perfect forecast.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	34.60	862099	504	1152	980
2	7.71	860482	358	1150	978
3	585.51	855833	511	1154	973
4	19.44	858081	194	1154	975
Tot/Comb	647.26	3436495	194	1154	976

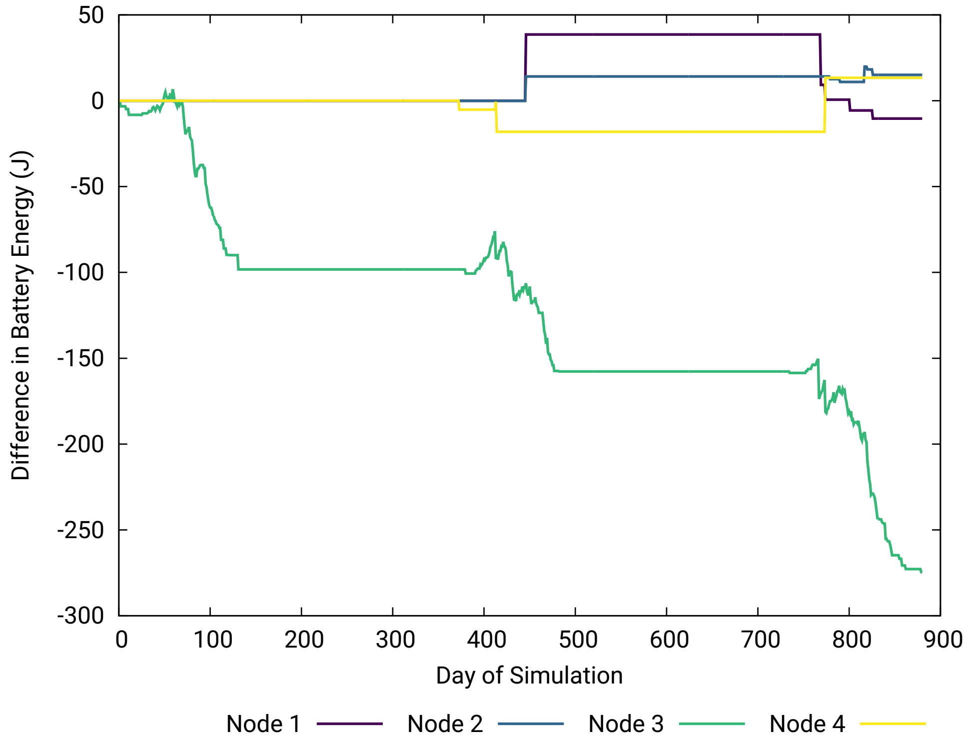


Figure 4.22: Difference between battery backup energies for CC2 and CC3 controllers where a perfect forecast was used. Values greater than zero represent less energy usage than the CC2 controller.

4.4.3 Improvements and Longer Forecasting Horizons

The optimization of the fuzzy controllers was revisited. Here, a larger initial population of 500 members was used, and the stopping condition was changed to no improvement in 50 generations. An additional day of forecasted energy was provided to the new controller, as well as a number of cases where no forecast was used, and only an input based on the state of the energy buffer was available. Different numbers of membership functions were used for the input fuzzy set to allow for a larger number of control outputs. The D_0 and D_0/D_1 cases discussed in Sec. 4.4.2 were redone to be consistent with these new optimizations.

Overall during optimization, the relatively high penalties on the usage of the energy reserve and semantic soundness resulted in candidate solutions where these factors contributed very little, if any, to the fitness value. With this being the case

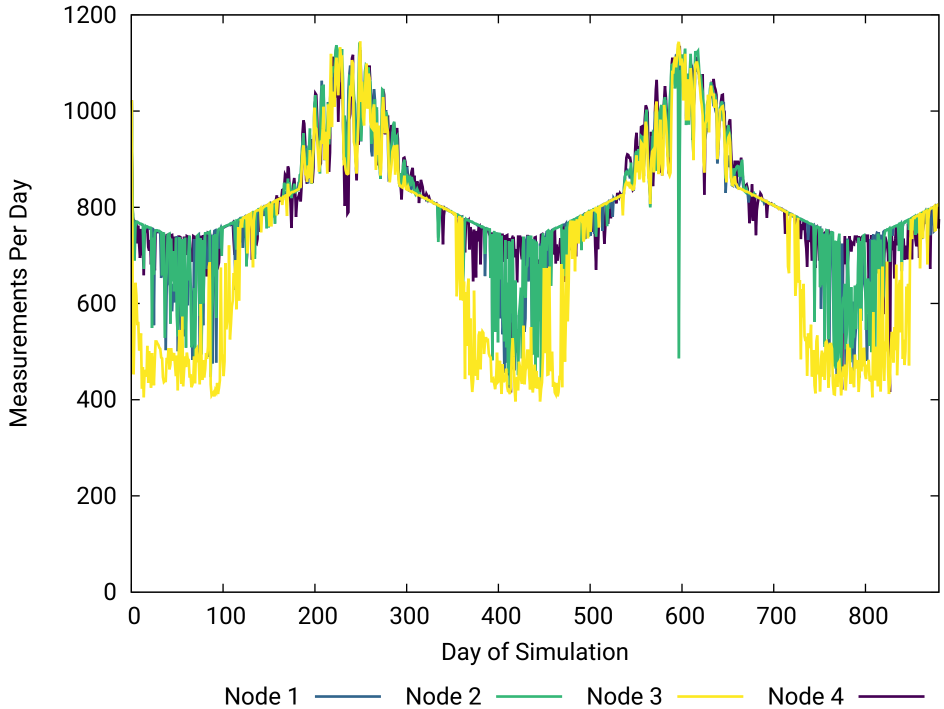


Figure 4.23: Number of measurements per day using a perfect energy forecast, CC2.

and because of the much lower weighting of uncollected energy, the difference between feasible solution's fitness values may have been very small.

The optimization was performed using a single location (Node 1/Moxee) and the first 670 days of the data set. Subsequent simulations of the controller to examine performance used all the WSU sites and an expanded time series of 880 days.

Depending on the types of membership functions used and the number of inputs included in the controller being optimized, there was a differing number of parameters available for the optimization. The number of parameters available for different controller setups are tabulated in Table 4.9. It can be seen that increased the number of forecasted days available to the controller greatly increases the number of parameters available. This large increase occurred because every combination of input membership function was provided with its own output membership function. The resulting controllers will likely have the possibility for simplification, allowing for

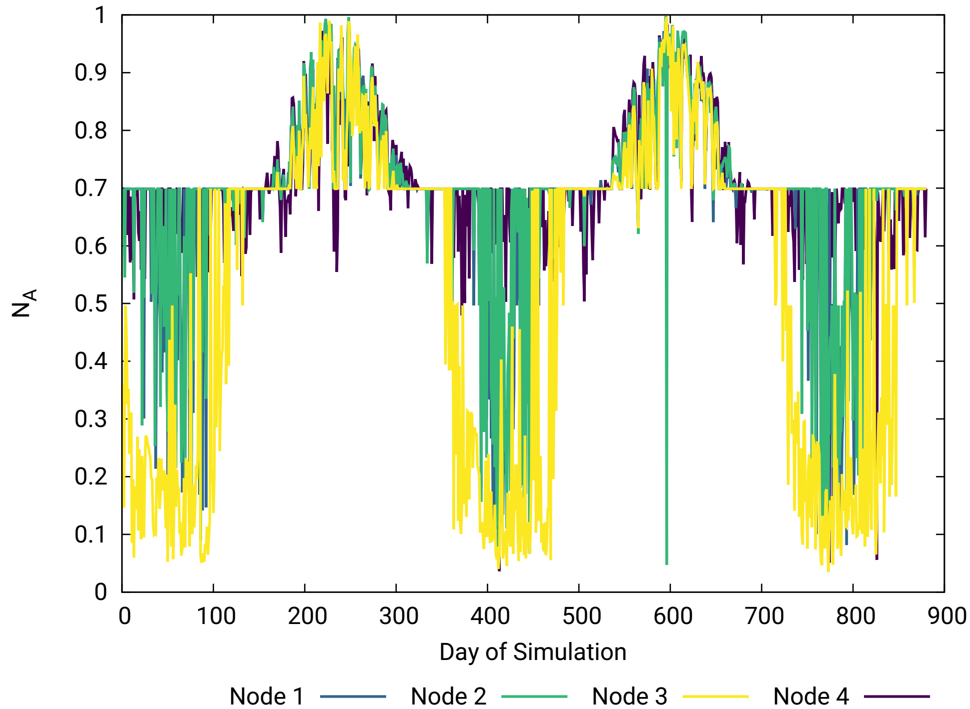


Figure 4.24: CC2/Perfect node activity (N_A) for sunrise update.

the full number of outputs allowed for the most flexibility in the outcome.

4.4.3.1 No Forecast Optimization

In order to determine if a forecast was necessary for good performance, controllers were optimized for the case where no forecast was provided, i.e., the fuzzy controller's only input is the state of the energy buffer. Three different cases where varying numbers of membership functions were used for the input set, for 3 (CC4), 5 (CC5), and 9 (CC6) input membership functions (MF). The increased number of membership functions may have allowed for more dynamic control, but also resulted in larger numbers of parameters to optimize.

For the case where 3 trapezoidal MF were used for each input, the resulting fitness was 11.59, while the case using 5 MF per input was 11.69 and the case with 9 MF was 11.77. The resulting membership functions for the input and outputs are

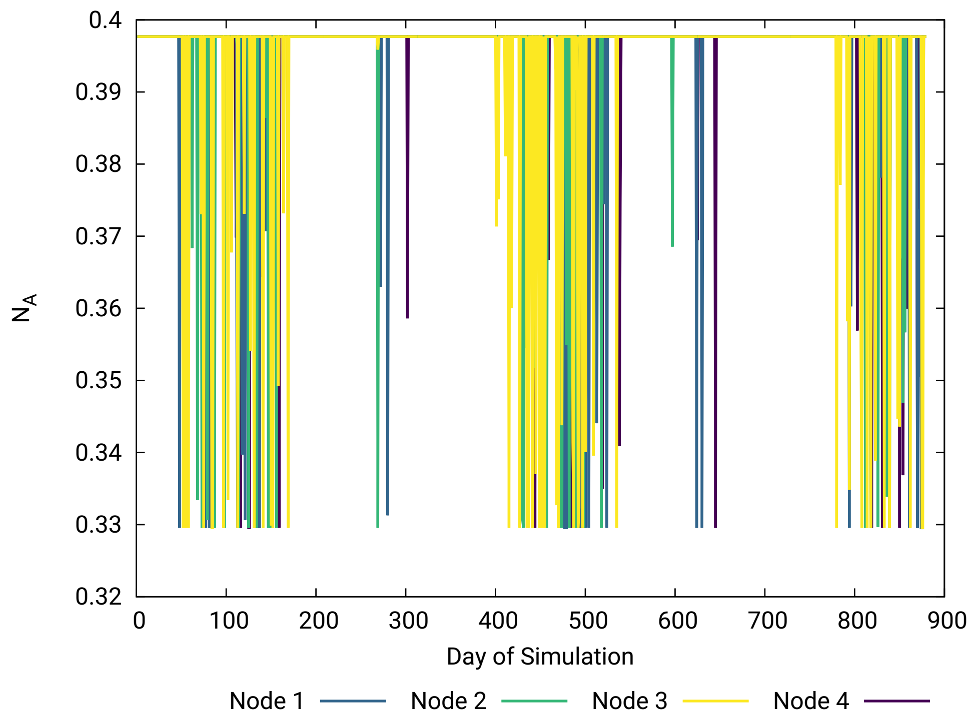


Figure 4.25: CC2/Perfect node activity (N_A) for sunset update.

shown in Fig. 4.28 (3 MF, CC4), Fig. 4.29 (5 MF, CC5), and Fig. 4.30 (9 MF, CC6). While the controllers with more MF were expected to provide finer control and improved performance, the 2 update/day scheme prevented this control from being exercised. Additionally, the case where 9 MF were used had an instance of high overlap between functions, which contributed negatively to its fitness value.

Fig. 4.31 shows the battery usage for the nodes at different sites during the simulated time period. As in other cases, due to its central location and the transmission scheme used in the simple simulation, Node 3 saw the greatest amount of energy usage during the simulation.

Fig. 4.32 shows the levels of the capacitor used as an energy buffer as reported at the sunrise and sunset updates. The portions of the curve that are truncated by zero indicate instances where the buffer was completely drained and the node required energy from the reserve to support operation. The controller using 9 MF had

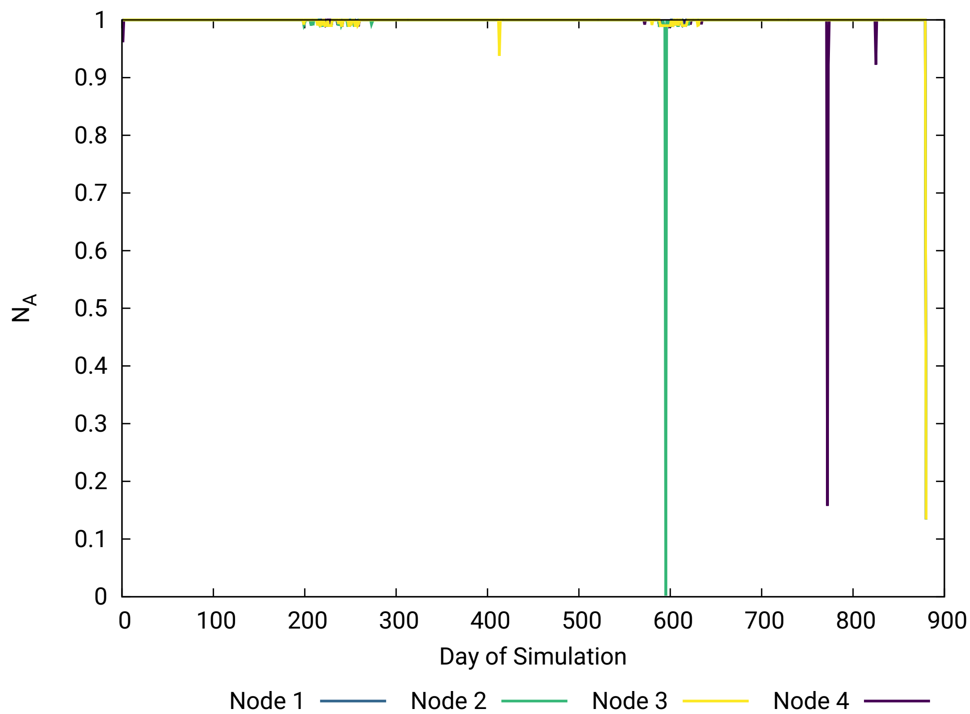


Figure 4.26: CC3/Perfect node activity (N_A) for sunrise update.

the simulated energy buffer levels shown in Fig. 4.33, which shows a different issue. In this case, the capacitor energy levels remained higher throughout the simulation and while this is preferable to the case in Fig. 4.32, it suggested that slightly higher node activities could have been supported with the rate of incoming energy.

The sunrise and sunset fuzzy outputs for the CC4 case are shown in Figs. 4.34 and 4.35. In this case, the sunset update was a constant value for all nodes, while the sunrise update was more dynamic, changing between the same value used at sunset and a much lower value during the winter months. This points to the sunset update selecting a value that was too high in some instances. The energy buffer was depleted during the night, causing the update at sunrise to select a lower value. Then, the sunset update again selected its constant activity level and used the excess energy that was collected during the day. This repeated until the incoming energy was able to support the activity level during both day and night time.

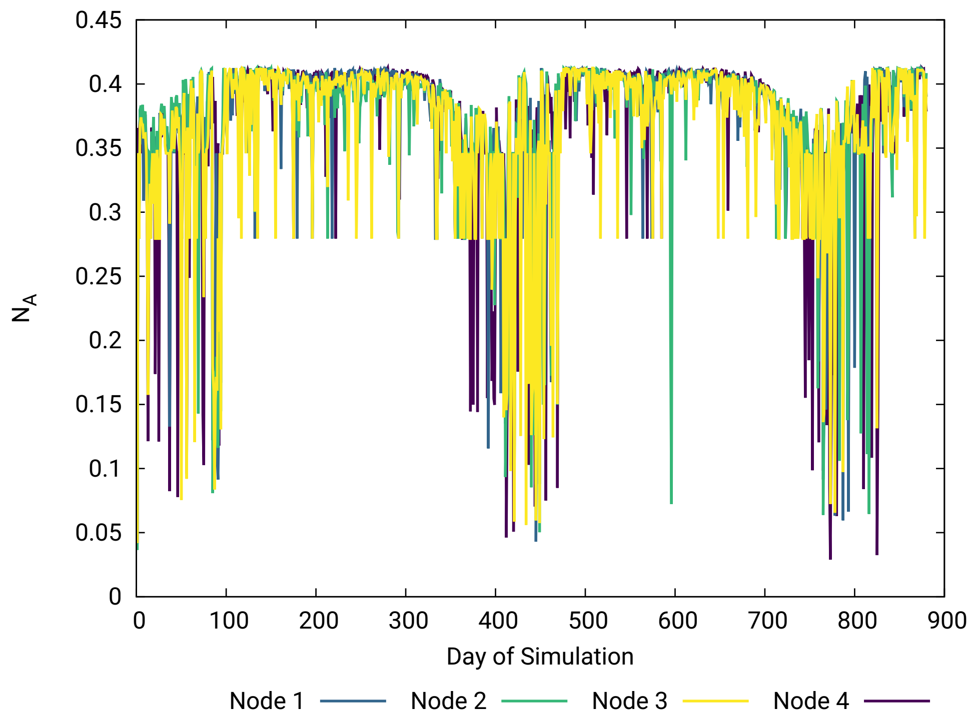


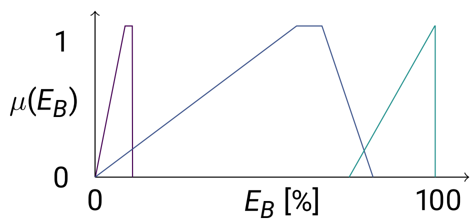
Figure 4.27: CC3/Perfect node activity (N_A) for sunset update.

Tabulated results for simulation of these 3 controllers are shown in Tables 4.10, 4.11, and 4.12, for 3, 5, and 9 MF, respectively. Overall, the increasing number of MF decreased the amount of reserve energy that was used across the entire network. Increasing the size of the controller from 3 MF decreased the total number of measurements taken across the network, as well as the average number of measurements taken per day. The controllers with higher numbers of MF had both higher overall minimum and maximum number of daily measurements for the network, indicating that the finer control allowed them to make better use of the available energy.

Tabulated results for the performance of the controller using only the state of the energy buffer are shown in Table 4.10. Compared to the reference controller using a perfect forecast, this controller used 44.3% of the energy reserve to take 112% of the measurements across the entire network. Comparing this controller to

Table 4.9: Number of parameters for the different optimizations.

Descriptions	Params optimized
No forecast with 3 trapezoidal MF and output singletons	15
No forecast with 5 trapezoidal MF and output singletons	25
No forecast with 9 trapezoidal MF and output singletons	45
D_0 with static reference controller input MF and output singletons	25
D_0 with 3 trapezoidal MF and output singletons	33
D_0 with 5 trapezoidal MF and output singletons	65
D_0/D_1 with 3 trapezoidal MF and output singletons	129
$D_0/D_1/D_2$ with 3 trapezoidal MF and output singletons	801
D_0 with 3 Gaussian MF and output singletons	21
D_0 with 5 Gaussian MF and output singletons	45



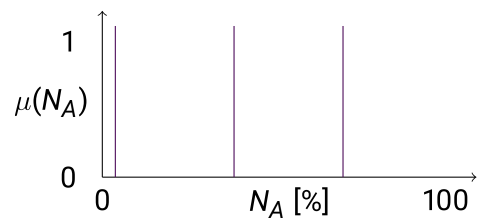


Figure 4.28: Fuzzy input (energy buffer, left) and output partitions (node activity, right) for CC4 (3MF).

the initial 5 MF optimized controller (CC1), 1965% of the reserve energy was used to take 92.7% of the total network measurements.

Table 4.10: CC4 simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	245.46	678358	442	1032	771
2	141.47	681126	446	1030	774
3	908.90	622515	416	1032	707
4	321.88	700005	244	1032	795
Tot/Comb	1617.25	2682004	244	1032	762

Comparing performance of the 3 (CC4) and 9 MF (CC6) cases, the 9 MF case used 23.6% of the reserve energy while taking 95.3% of the total network measurements.

4.4.3.2 Optimizations using Forecasts

Static Input Membership Functions Performing a similar optimization to that done in [1], a controller was created using the input fuzzy sets from the reference con-

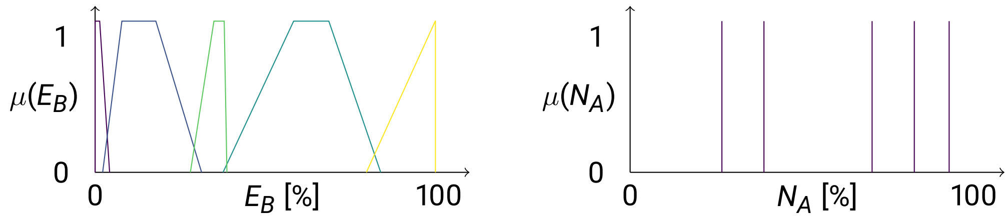


Figure 4.29: Fuzzy input (energy buffer, left) and output (node activity, left) partitions for CC5 (5MF).

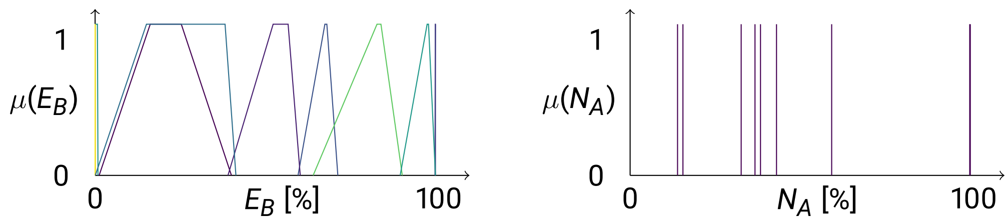


Figure 4.30: Fuzzy input (energy buffer, left) and output (node activity, right) partitions for CC6 (9MF).

troller and allowing optimization to occur on the output singletons (CC15). The use of static input membership functions removed some of the flexibility, but also reduced the number of parameters that were available for the optimization method to operate on. For the controller in [1] there were different outputs for measurement and transmission, while here they were linked to the single output of node activity. Additionally, the update scheme was different with this controller performing just two node activity updates during the day, one at sunrise and one at sunset.

One trial finished with a fitness value of 12.41 after running for 372 generations. For the simulation of this controller applied to the complete network, a total of 3,122,369 measurements were taken, using 591.83 J of reserve energy. As shown in Fig. 4.36, much of the energy reserve usage occurred during the day.

Using Eqn. 4.2, this translated to an effective N_A value of 0.6033, which represented an improvement over some controllers. Compared to the reference controller with a perfect forecast, this controller took 131% of the total network measurements while using 16.2% of the reserve energy across the entire network. While this has a higher percentage of total measurements versus the initial 5 MF controller (CC1) at 121%, it also used more of the available reserve energy (compared to 2.2%).

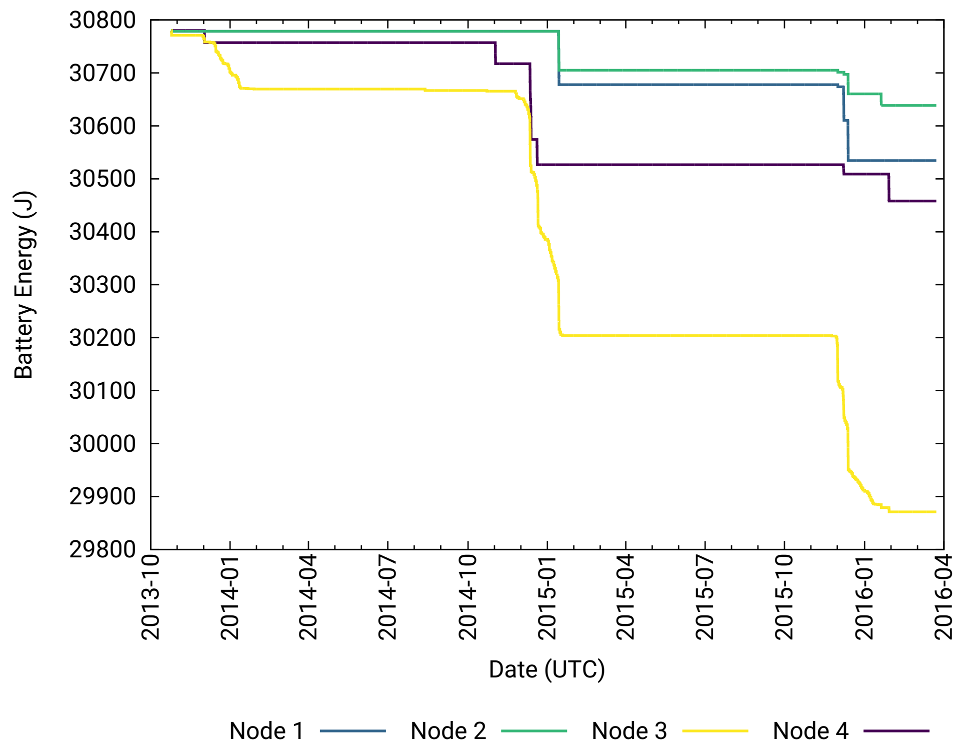


Figure 4.31: Battery reserve energies for CC4 (3MF).

Table 4.11: CC5 simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	101.38	609289	470	1148	692
2	47.83	610599	469	1102	694
3	429.59	594007	469	1272	675
4	374.40	632502	470	1374	719
Tot/Comb	953.2	2446397	469	1374	695

Different Forecast Horizons Three forecasting horizons were supplied to fuzzy controllers in order to examine the effect foreknowledge of harvestable energy has on a node's ability to make the best use of the energy present in the environment while leaving as much energy in reserve storage as possible. The 3 cases were: the forecast for the upcoming day was available at sunrise, D_0, that day and the next day's forecast were available D_0/D_1, and both those days with the next day, $D_0/D_1/D_2$.

The controller with the lowest fitness function created using the D_0 setup, CC7,

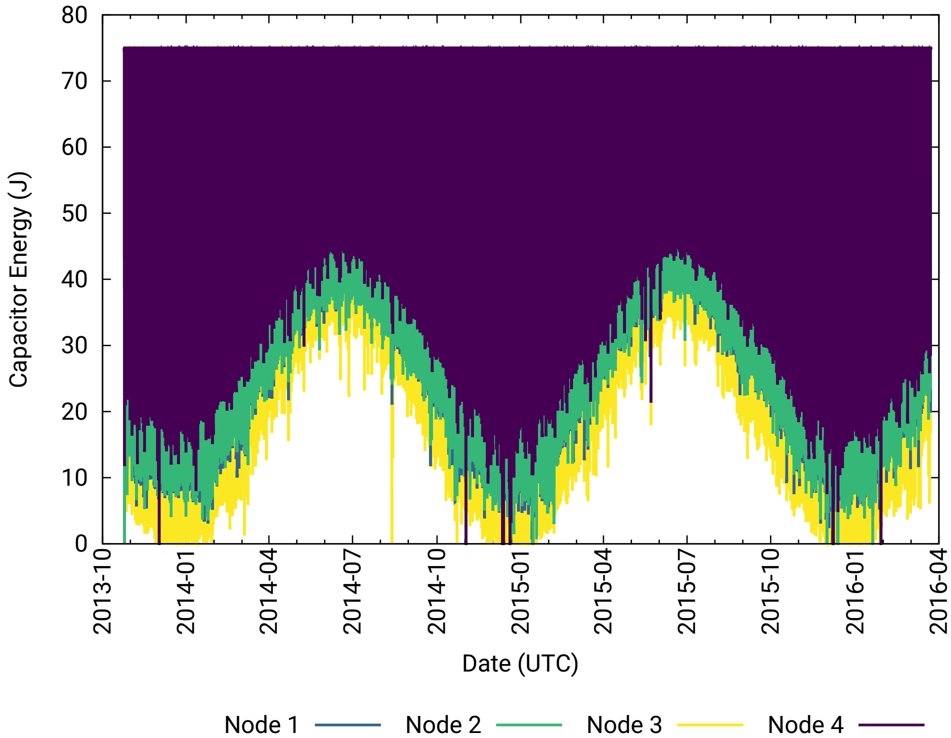


Figure 4.32: Capacitor energies for CC4 simulation (Node 1 values concealed by Node 2 values due to their similarity).

had a fitness value of 11.44. The input and output partitions are shown in Fig. 4.37.

Simulating this controller with a perfect forecast results in the energy reserve and buffer levels are shown in Figs. 4.38 and 4.39, respectively.

The D_0/D_1 case, CC8, performed very well with a fitness value of 11.35. The energy reserve levels for this controller using a perfect forecast are shown in Fig. 4.40 with very little energy usage. The energy buffer levels for this controller is shown in Fig. 4.41 and, as expected, indicates very few instances of complete depletion. The number of measurements per day taken by the nodes in the simulation are shown in Fig. 4.42. This curve tracks the expected incoming solar energy well with the bulk of the dips occurring during winter months.

Results of the simulation of these controllers using perfect forecasts are tabulated in Tables 4.13 and 4.14 for the D_0 (CC7) and D_0/D_1 (CC8) horizon cases, respectively.

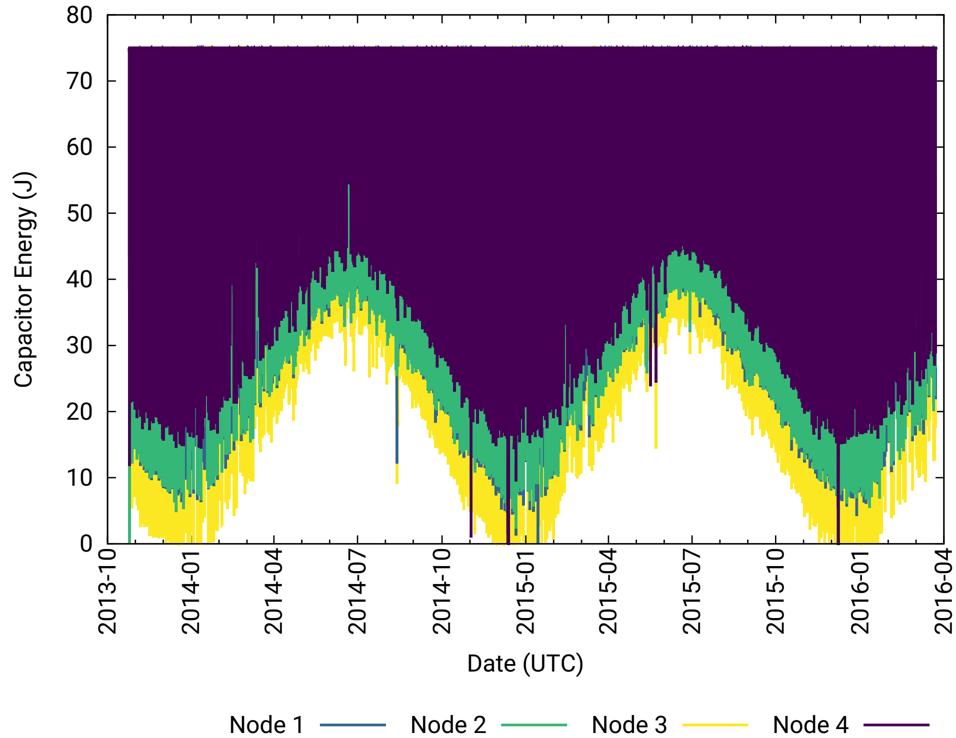


Figure 4.33: Capacitor energies for CC6 simulation (Node 1 values concealed by Node 2 values due to their similarity).

Comparing these controllers, the controller using D_0 CC7 and D_0/D_1 CC8 forecasted energy values took more total measurements and uses less reserve energy than the controller using only the D_0 forecast, CC7.

The controller using $D_0/D_1/D_2$ forecasts, C11, ended optimization with a higher fitness value than the other two controllers at 12.54. After running the simulation of this controller with the perfect forecast, results very similar to the case of a zero activity level (Table 4.2) were obtained. Possible reasons for the poor solution obtained by this optimization include the large number of parameters involved in optimization (Table 4.9) requiring a larger population than was provided, and the term in the fitness function related to coverage. The additional day of forecast solar energy may not have been useful for the control of the sensor nodes due to the size of the energy buffer, but the penalty incurred by having poor coverage of a fuzzy set prevented the membership functions from being collapsed so that they had no

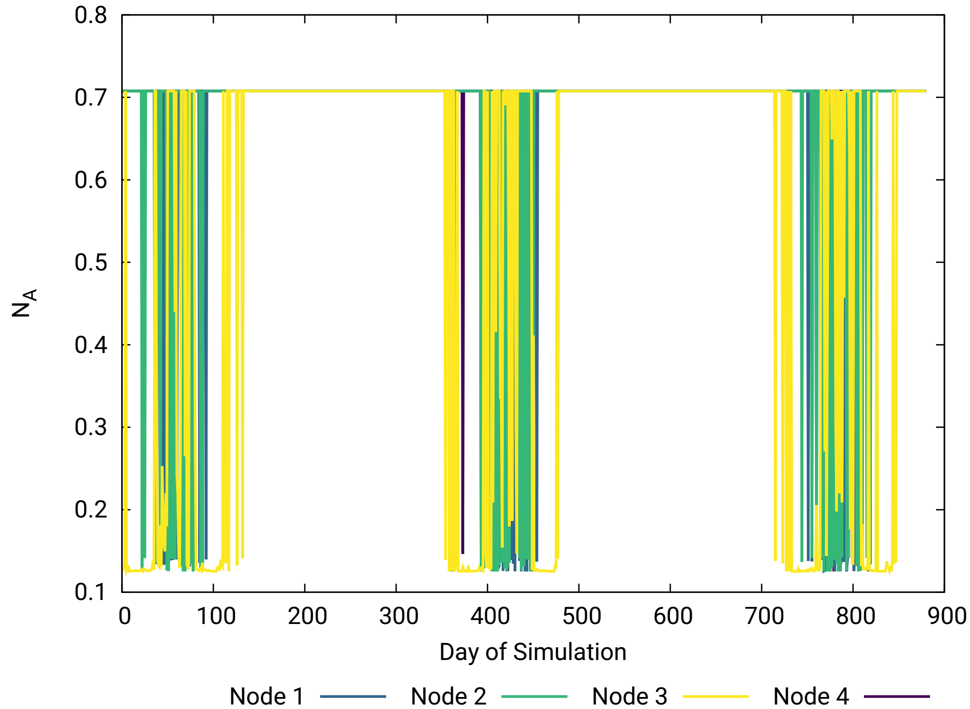


Figure 4.34: Node activity (N_A) for sunrise update for the CC4 simulation.

Table 4.12: CC6 simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	33.81	648713	351	1139	737
2	1.30	647084	549	1137	735
3	217.73	596797	424	1141	678
4	128.90	663693	335	1148	754
Tot/Comb	381.74	2556287	335	1148	726

effect on the output.

During the simulation where the perfect forecast was provided the controller CC7, used 146% of the reserve energy used in the CC8/Perfect simulation while taking fewer measurements over the simulated time period (93.7%). The controller using the 2-day forecast (CC8) had higher minimum and maximum values of M_D, showing that it was better adapted to the variability in harvesting opportunities during the simulated period. Comparing the behaviour of the controllers, the sunrise and sunset activity updates had similar behaviours, both selecting lower values at

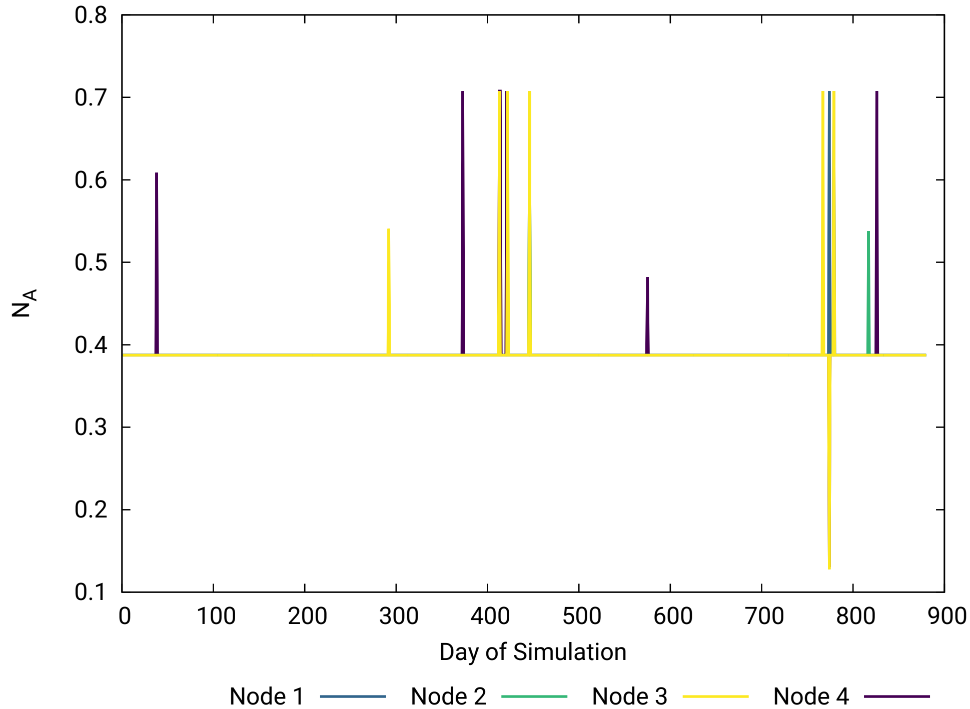


Figure 4.35: Node activity (N_A) for sunset update for CC4 simulation.

Table 4.13: CC7/Perfect simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	4.95	791809	232	1117	900
2	9.13	785221	380	1117	892
3	326.64	779242	301	1129	886
4	0.00	798658	304	1134	908
Tot/Comb	340.72	3154930	232	1134	896

night and very high values in the morning. However, the CC8 controller selected higher values at sunset, due to the foreknowledge of the next day's harvesting opportunities.

The case using the D_0 energy forecast with 5 trapezoidal MF (CC1) was also retried using the adaptive DE algorithm and resulted in CC16. Optimization resulted in a fitness function of 11.72. The results for this controller simulated using a perfect forecast are shown in Table 4.15. These values represent an improvement over the original attempt using 5 MF (CC1) by taking 93.3% of the total number of mea-

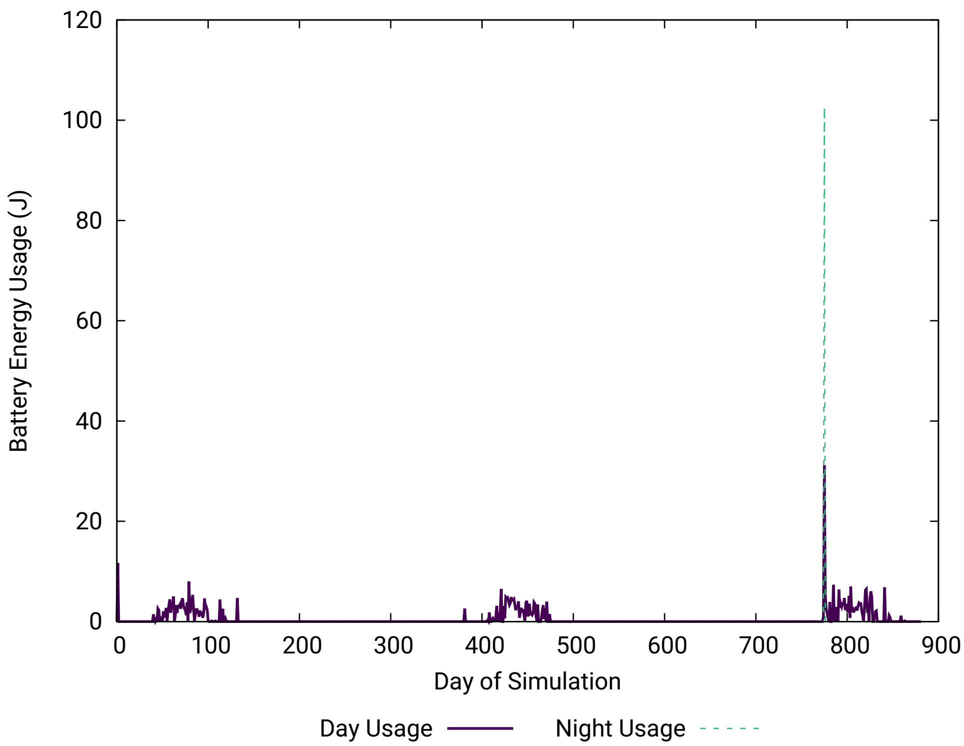


Figure 4.36: Day and night energy reserve usages for the entire network broken down by day and night for the CC15/Perfect simulation.

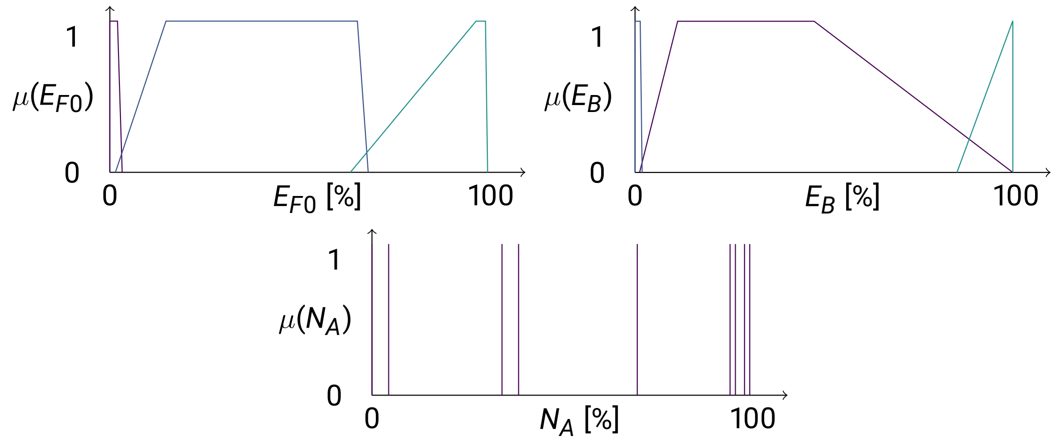


Figure 4.37: Fuzzy input (energy forecast, right; energy buffer, left) and output (node activity, lower) partitions for CC7.

Table 4.14: CC8/Perfect simulation results.

Node	E_R (J)	M_T	Min M_D	Max M_D	Mean M_D
1	1.44	846382	399	1261	962
2	1.30	850069	287	1257	966
3	179.37	826683	396	1263	939
4	51.97	843765	322	1264	959
Tot/Comb	234.08	3366899	287	1264	957

surements of the CC1 controller, while using only 14.21% of the total reserve energy across the network.

Table 4.15: CC16/Perfect simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	1.44	690110	181	1043	784
2	1.30	673098	162	1042	765
3	8.94	630956	171	1046	717
4	0.00	705375	214	1046	802
Tot/Comb	11.68	2699539	162	1046	767

Gaussian Membership Functions Gaussian membership functions are defined by only two parameters, mean and standard deviation. The lower number of parameters, when compared to trapezoidal membership functions, translates to fewer dimensions for optimization. These membership functions are symmetrical. However, an asymmetrical effect may be obtained by having membership functions

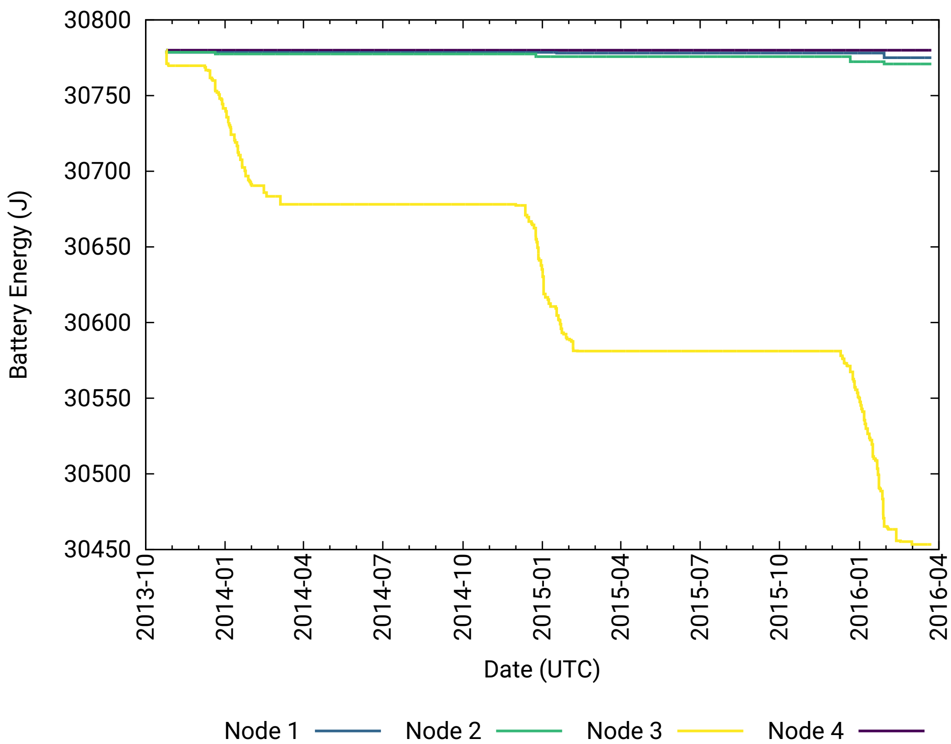


Figure 4.38: Battery reserve energies for CC7/Perfect simulation.

placed such that they extend beyond the range of the input value. Naive calculation of these membership functions would be more intensive, but approximations or look-up tables could be used to mitigate this on limited hardware.

Input and output partitions for cases where 5 MF (CC9) were used are shown in Fig. 4.46. As was the case with the trapezoidal membership functions, a number of the input membership functions were collapsed to singletons with some having a high degree of overlap, again leading to possible simplifications of the rule set. Performing the same optimization only using 3 membership functions per input (CC10) resulted in the partitions shown in Fig. 4.45.

Because of the constraints placed on the creation of trapezoidal membership functions, i.e. the membership levels of the different points are fixed, these membership functions represent shapes that could not be created with the setup as was used in the trapezoidal cases. Fitness values for these two optimizations were

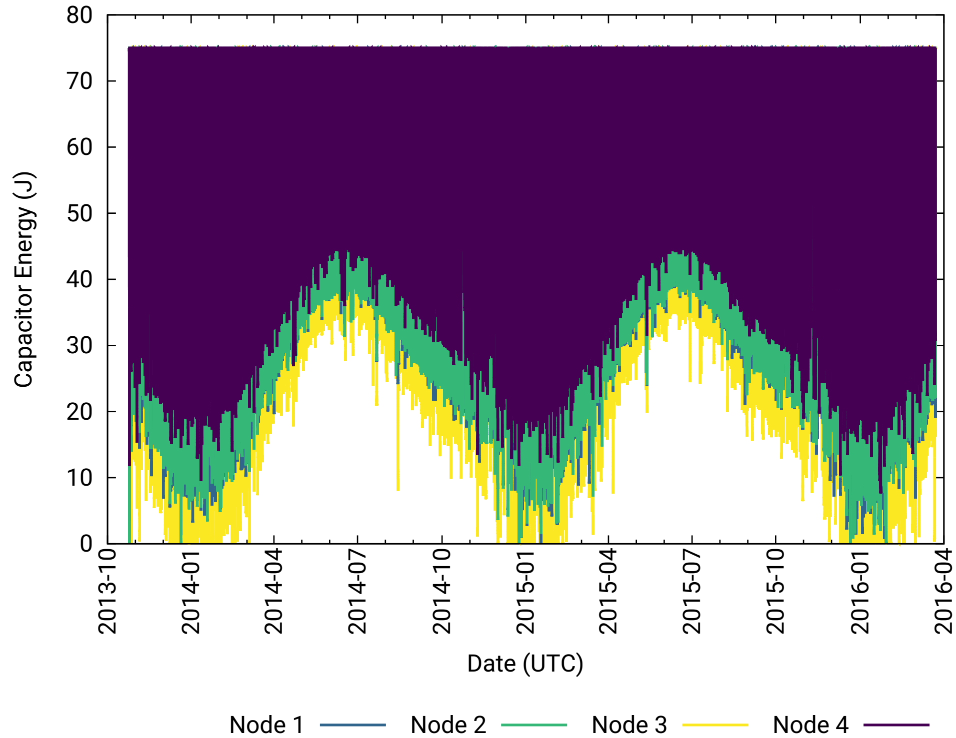


Figure 4.39: Capacitor energies for optimization of CC7/Perfect simulation.

11.681 and 11.682 for the 3 and 5 membership function cases, respectively. While the fitness values were similar when comparing the Gaussian (11.68) and trapezoidal (11.72) 5 MF cases, in the 3 MF case, the Gaussian (11.68) was worse than the trapezoidal case (11.44).

In the resulting fuzzy sets, each one had single membership function that spanned the entire input space, with each of the others having been reduced to singletons. This suggests that for these controllers, only one output singleton was activated for the majority of the time and that an extremely simple controller could be used.

Results obtained by simulating these two controllers are shown in Tables 4.16 and 4.17 for the 3 and 5 membership function cases, respectively. Both controllers caused very similar usages of energy reserve and similar numbers of measurements to be taken. Comparing the perfect forecast performance of two controllers employing 3 MF (CC7 and CC10), the controller using Gaussian functions used 17.7%

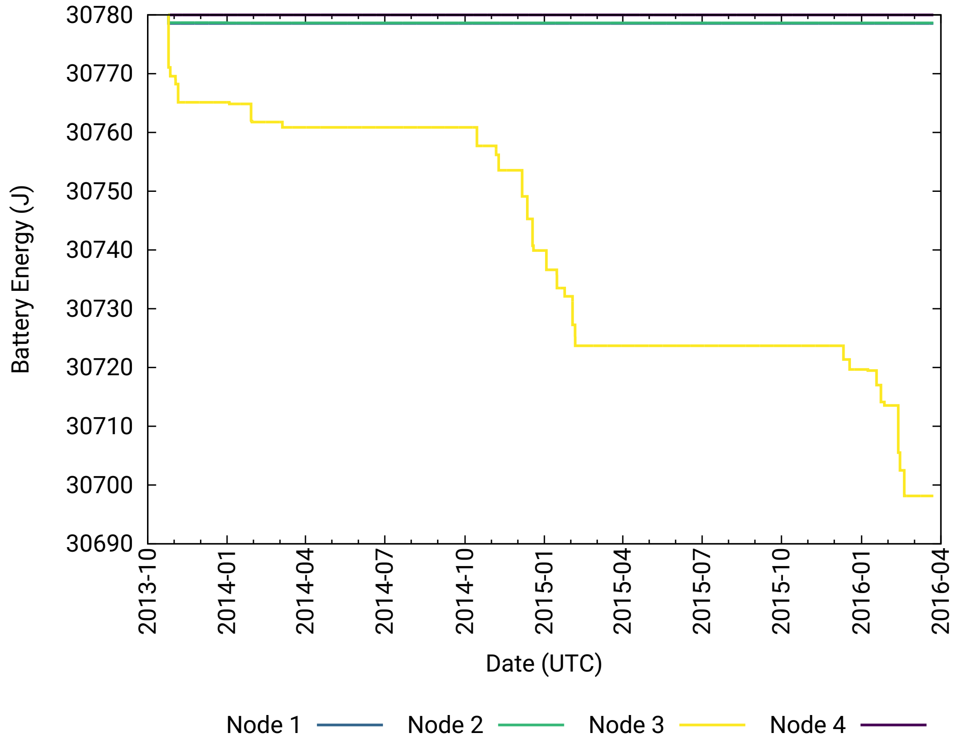


Figure 4.40: Battery reserve energies for optimization of CC8/Perfect simulation.

of the reserve energy to take 77.9% of the measurements across the network.

Table 4.16: CC10/Perfect simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	3.46	615138	572	872	699
2	1.30	615013	588	808	699
3	45.47	614421	553	871	698
4	10.07	613254	558	868	698
Tot/Comb	60.30	2457726	553	872	698

4.4.3.3 Summary

The fitness values for selected resulting controllers are shown in Table 4.18. Summaries of the controllers as simulated using a perfect forecast are shown in Table 4.19. For the case where no forecasts were used, the increasing number of membership functions led to higher fitness function values. The higher value of the no-forecast 9 membership function case is partially explained by the high de-

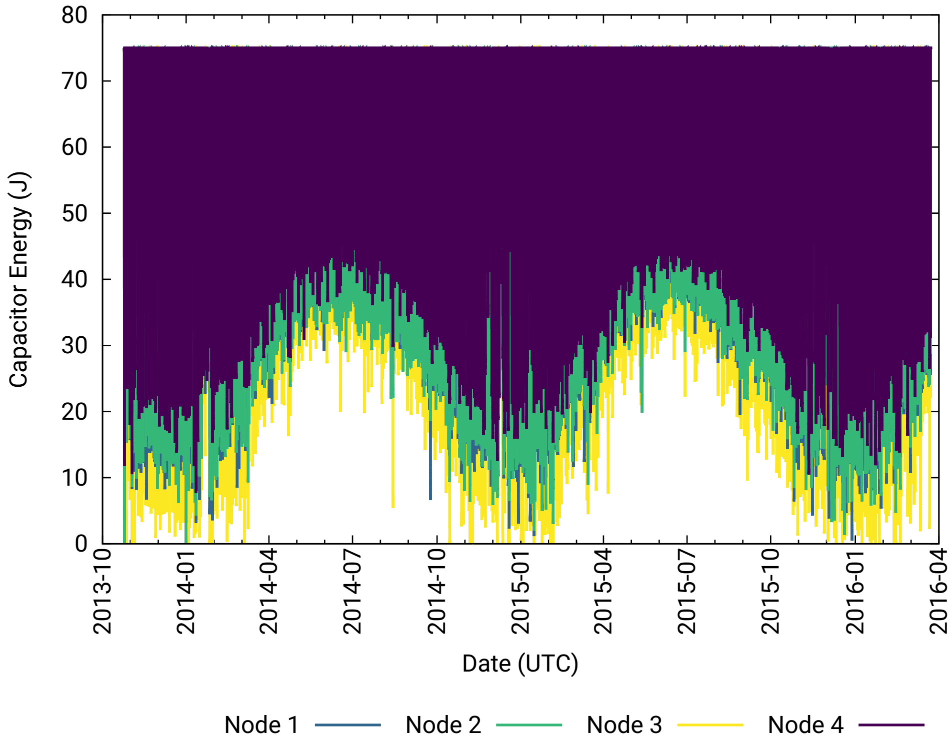


Figure 4.41: Capacitor energies for optimization of CC8/Perfect simulation (Node 1 values concealed by Node 2 values due to their similarity).

gree of overlap between 2 of the membership functions, but more generally the poorer performance may be attributed to population sizes that were too small for the number of parameters available for optimization in these larger cases. Where forecasts were included, the D_0/D_1 case (CC8) had better fitness values than other optimized cases. This case took 107% of the measurements of the next highest controller (CC7), while using 68.70% of the total reserve energy when considering the entire network. Moving to a longer forecast horizon resulted in a much higher fitness function value. Two possible reasons are that the increase in number of parameters realized through the addition of the extra day required a much larger population than was used, or the size of the energy buffer relative to the amount of energy that can be used in a day may mean that the 3rd day of forecasted energy is extraneous information. In this case, the high penalty on the coverage and distinguishability of the associated fuzzy set forced the membership functions to occupy

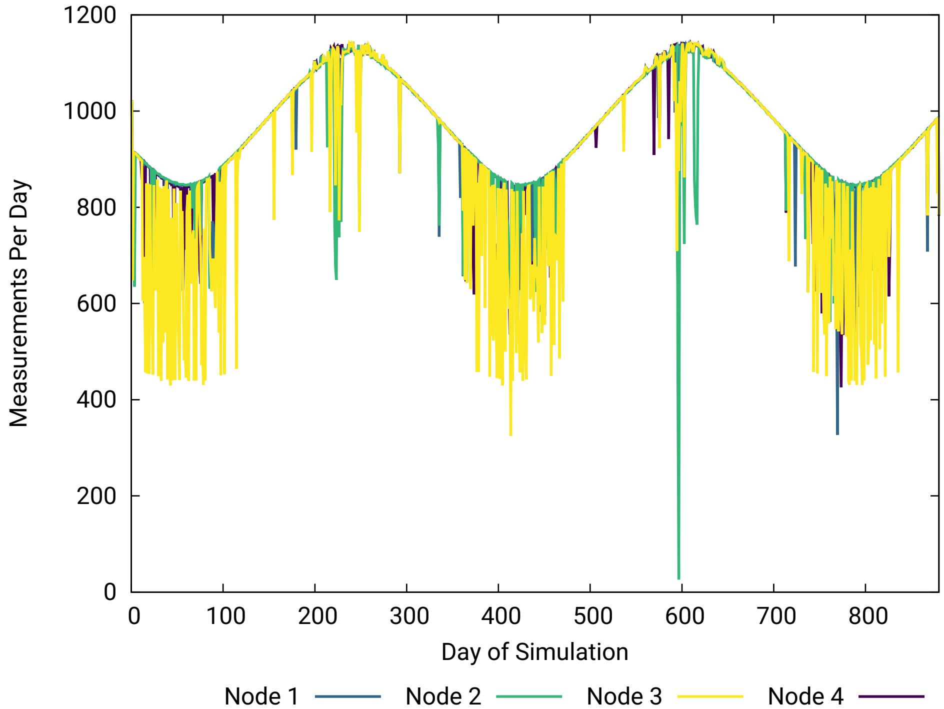


Figure 4.42: Number of measurements per day for CC8/Perfect simulation.

Table 4.17: CC9/Perfect simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	2.57	616616	572	901	701
2	1.30	618415	586	847	703
3	48.35	615176	553	903	699
4	9.56	613584	558	903	697
Tot/Comb	61.78	2463791	553	903	700

the whole input space.

For these controllers, comparisons with the reference controller can be made:

- The no-forecast 3MF (CC4) controller took 112% of the total network measurements, while using 44.35% of the total reserve energy across the network. Increasing the number of membership functions in the input and output to 5 (CC5), resulted in a controller that took 102.29% of the measurements while using 26.14% of the reserve energy and further increasing the number of membership functions to 9 (CC6) resulted in a controller that 106.88% of the mea-

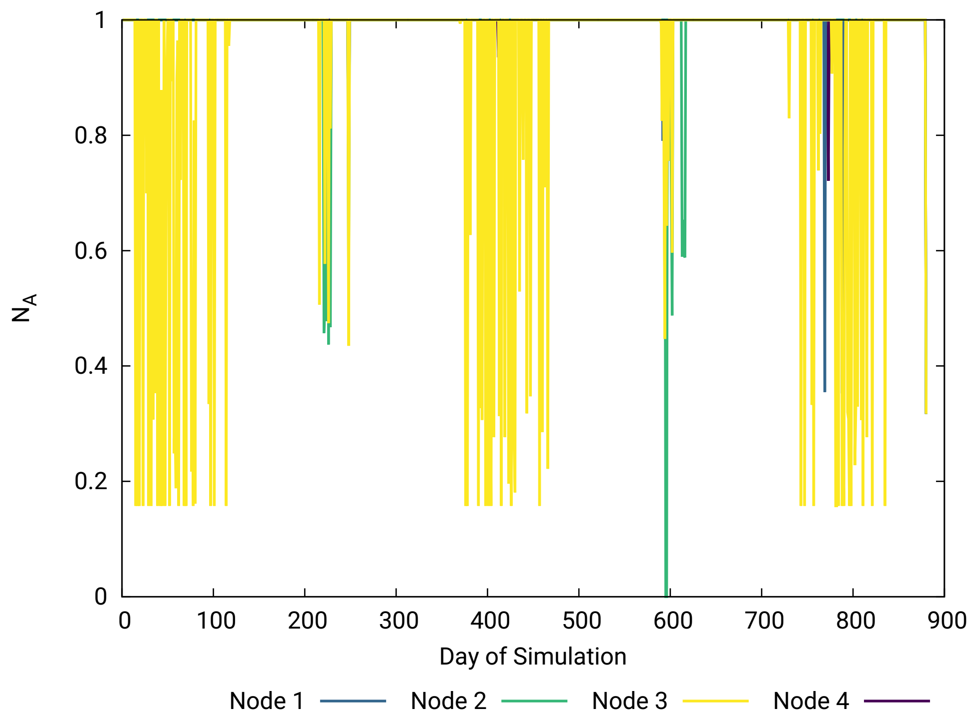


Figure 4.43: Sunrise node activity (N_A) for CC8/Perfect simulation.

surements and used 10.47% of the energy reserve.

- Reducing the number of membership functions to 3 and providing forecasts with different horizons as inputs generally resulted in increased performance when compared to the cases where only the state of the energy buffer was provided as an input. For these optimizations, the controller provided with the D_0 forecast (CC7) took 131.91% of the measurements while using just 9.34% of the total energy reserve energy. The controller that used both the D_0 and D_1 forecasts (CC8) took 140.78% while using only 6.42% of the energy reserve.

- An optimization undertaken using 5 Gaussian MF for both inputs of the D_0 controller scheme was performed (CC9). The resulting solution had a fitness of 11.70, which was similar to the value of 11.72 obtained using 5 trapezoidal functions. The 3 MF controller using Gaussian membership functions (CC10) took 102.76% of the total network measurements of the reference controller

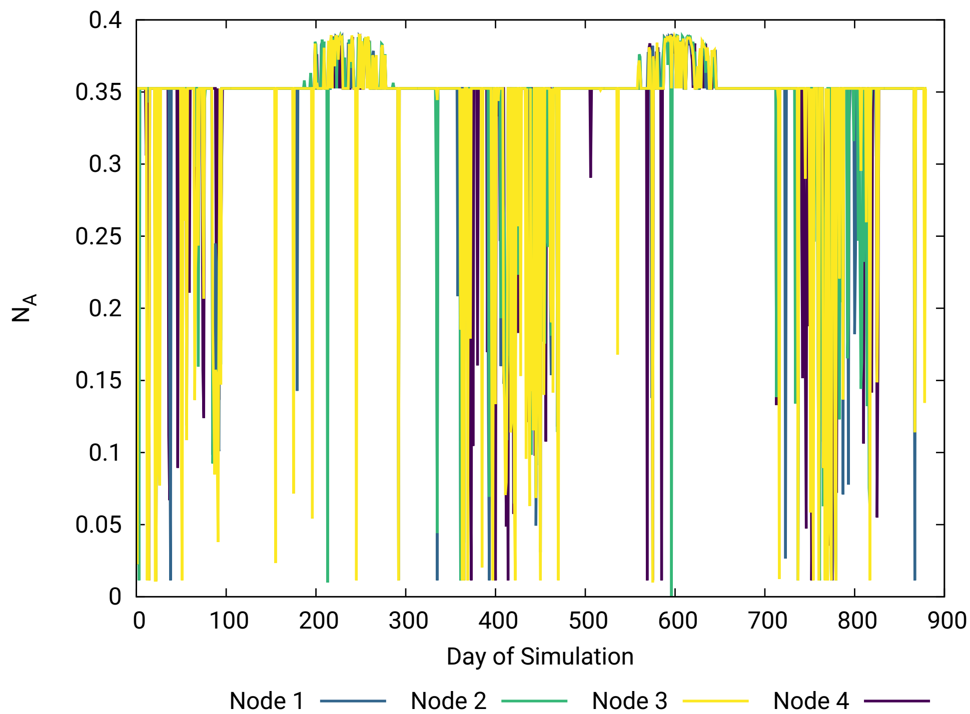


Figure 4.44: Sunset node activity (N_A) for CC8/Perfect simulation.

while using 1.65% of the reserve energy. The 5 MF case took 103.02% of the total network measurements while using 1.69% of network energy reserve.

4.4.4 Unconstrained Node Activity Optimization

For these controllers the upper limit constraint of $0 \leq N_A \leq 1$ was removed, allowing the measurements and transmissions to occur with a greater frequency than in the reference controller and ideally allowing for better usage of harvestable energy when it is available. The fitness values of the resulting optimized controllers are shown in Table 4.20. In the case where no forecast was provided (CC12), the constrained and unconstrained cases using 3 MF resulted in similar fitness values. However, in this case, the perfect forecast simulation showed that this controller used only 44% of the reserve energy while taking 99% of the total measurements. Examining the output partition (Fig. 4.47) showed that there are no singletons cor-

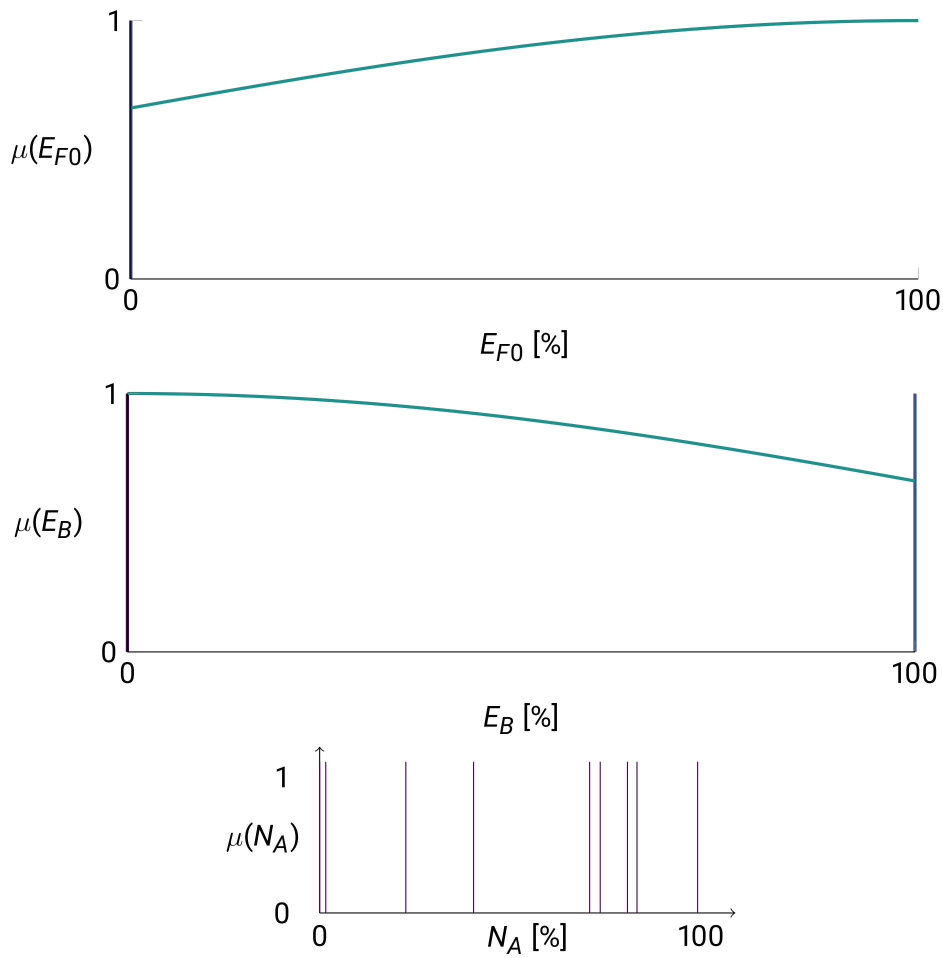


Figure 4.45: CC10 fuzzy input (energy forecast, top; energy buffer, middle) and output (node activity, bottom) partitions.

responding to a node activity greater than one, meaning that this solution was entirely possible for the case where N_A was constrained. Additionally, the fitness value for this optimization was nearly identical to the fitness obtained where constrained node activities were used. This suggests that without foreknowledge of the incoming energy, there is nothing to be gained by allowing a node to use more energy.

For the case where N_A did not have an upper limit, the optimization matched the node activity to the amount of available energy to a very high degree. For example, the fitness value for the controller using a D_0 energy forecast (CC13) was 10.92 and included node activities far in excess of 100%. The fuzzy sets for the inputs and outputs are shown in Fig. 4.48. Unfortunately, the increased size of the search

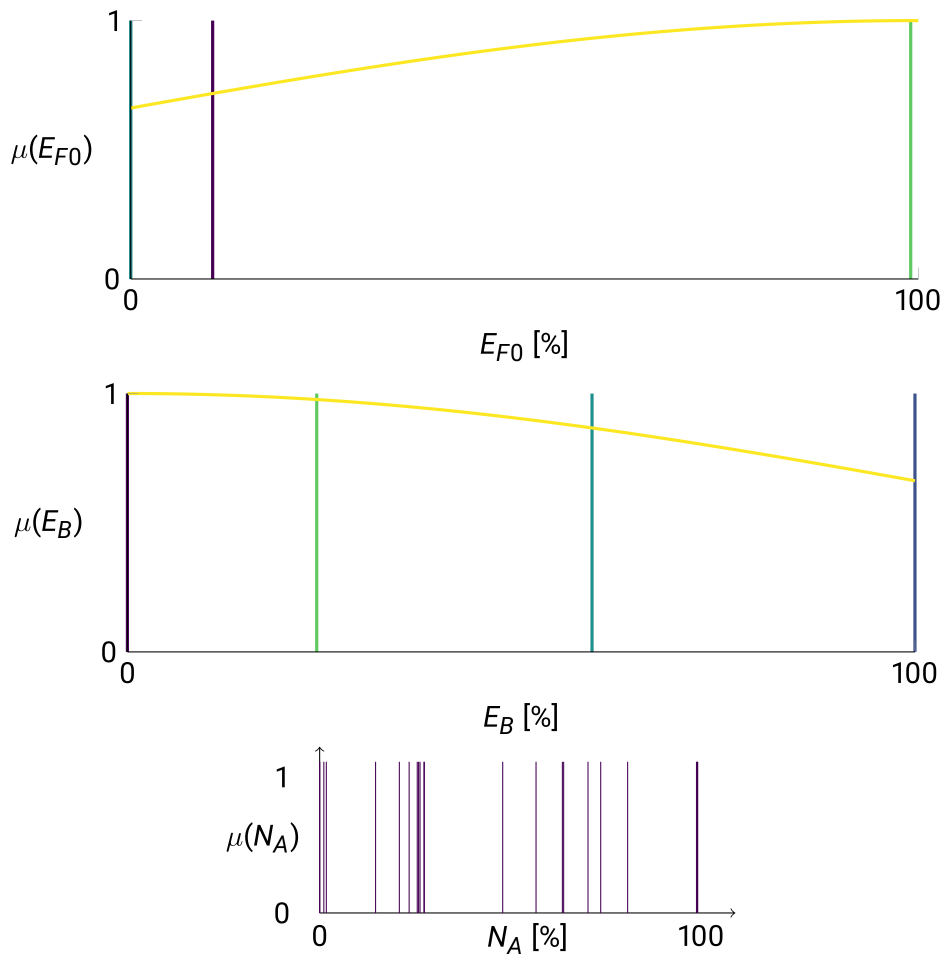


Figure 4.46: CC9 fuzzy input (energy forecast, top and energy buffer, middle) and output (node activity, bottom) partitions.

space realized through removal of the upper bound on N_A increased the difficulty in reaching a good solution.

However, when this optimized controller was applied to a simulation of the entire network, the differences between optimization and simulation proved to be great. Even using a perfect forecast, the addition of nodes not present during optimization failed to perform well. In the case of the controller using only the current day forecast, Node 3 used a large amount of reserve energy, shown in Fig. 4.49. Tabulated results for the CC13/Perfect simulation are shown in Table 4.21.

The node activity values as output by the controller for this case are shown in Figs. 4.50 and 4.51 for the sunrise and sunset updates, respectively. These plots

Table 4.18: Best fitness values for different controllers where node activity is constrained.

Controller	Fitness Value
D_0/D_1 3 MF (CC8)	11.35
D_0 3MF (CC7)	11.44
No-Forecast 3MF (CC4)	11.59
No-Forecast 5MF (CC5)	11.69
D_0 5MF (CC16)	11.75
No-Forecast 9MF (CC6)	11.77
Static input D_0 (CC15)	12.41
$D_0/D_1/D_2$ 3 MF (CC11)	12.54

Table 4.19: Network summary for perfect forecast simulation of different controllers where node activity is constrained.

Controller	E_R (J)	M_T	Avg. M_D	Min. M_D	Max. M_D
No-Forecast 3MF (CC4)	1617.25	2682004	762	244	1032
No-Forecast 5MF (CC5)	953.20	2446397	695	496	1374
No-Forecast 9MF (CC6)	381.75	2556287	726	335	1148
D_0 3MF (CC7)	340.72	3154930	896	232	1134
D_0 5MF (CC16)	11.69	2699539	767	162	1046
D_0/D_1 3MF (CC8)	234.08	3366899	957	287	1264
$D_0/D_1/D_2$ 3MF (CC11)	11.68	91680	102	25	102
Gaussian 3MF (CC10)	60.30	2457726	698	553	872
Gaussian 5MF (CC9)	61.78	2463791	700	553	903
Static input D_0 (CC15)	592.13	3122369	887	168	1135

show that only Node 3 was fully activating the rule associated with the very high node activity singleton seen in Fig. 4.48, and that this rule was only fully activated during the winter months during sunrise updates. This rule was fully activated by E_F and E_B both having fully activated set1.

For the unconstrained controller using two energy forecasts, the D_0/D_1 scheme, node activity values as high as 26 were used. This corresponded to nodes taking a measurement approximately every 2 seconds and making a transmission roughly every 5 seconds. Simulation of the controller revealed that the minimum node activity was consistently selected for overnight periods, while very high values were selected for daytimes. The high node activities required very long simulation pro-

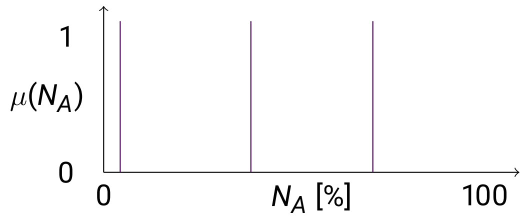


Figure 4.47: Fuzzy partition of output node activity N_A for no-forecast unbound optimization.

Table 4.20: Best fitness values for different controllers where node activity is unconstrained.

Controller	Fitness Value
D_0 3 MF (CC13)	10.92
No Forecast 3 MF (CC12)	11.60
D_0/D_1 3 MF (CC14)	12.15

cessing times, and for many applications this frequency of measurement and transmission is likely not required. Additionally, this controller was especially vulnerable to failing with non-perfect forecasts as selection of a very high node activity without the incoming energy to support it caused more energy reserve usage in a single day than the entire simulated period with a different controller.

In the case where the node activity is unconstrained, the candidate solutions were able to perform better because they used more of the energy available for harvest than the controller with its activity constrained. However, as the simulation where the optimization was performed only used one location, the removal of the limit allowed for creation of an over fit solution. In that simulation the node used very little reserve energy, while consuming a great amount of energy harvested from the environment. However, when this controller was applied to the entire network, the different weather patterns and node layouts caused high energy use in one node for the unconstrained D_0/D_1 controller.

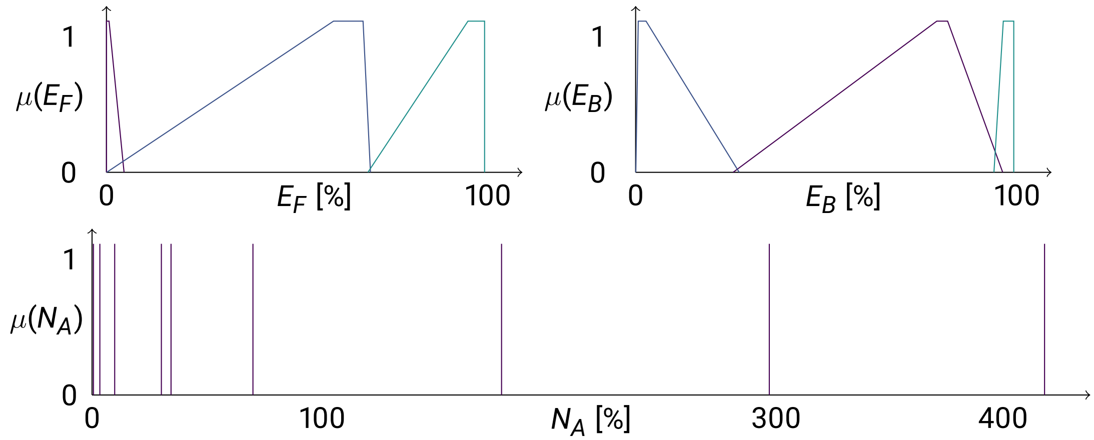


Figure 4.48: Fuzzy input (energy forecast, top left; energy buffer, top right) and output (node activity, bottom) partitions for D_0 energy forecast unbound optimization, CC13.

Table 4.21: CC13/Perfect simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	2.38	1023166	204	2873	1163
2	1.30	1074873	205	2867	1226
3	4123.22	1282003	94	2885	1457
4	0	966118	175	2886	1098
Tot/Comb	4126.90	4349760	94	2886	1236

4.4.5 Relationship between optimization fitness and simulation fitness

The comparison of controller performances when the entire network is used is difficult because slight changes in any node activity can greatly affect the energy usage of other nodes in the network. This also has implications when a forecast is used, and there may be cases where more energy is forecast than actually happens, and in this case the increased node activity may increase the energy usage of other nodes disproportionately

Because of the relationship between node activity and the energy reserve usage plotted in Fig. 4.5, increasing activity levels resulted in reserve usage growing much faster than the corresponding drop in the amount of unused solar energy (Fig. 4.4). This, combined with the weighting used in the calculation of solution fitness, means that calculating the fitness for the simulations in the same way that the fitness is

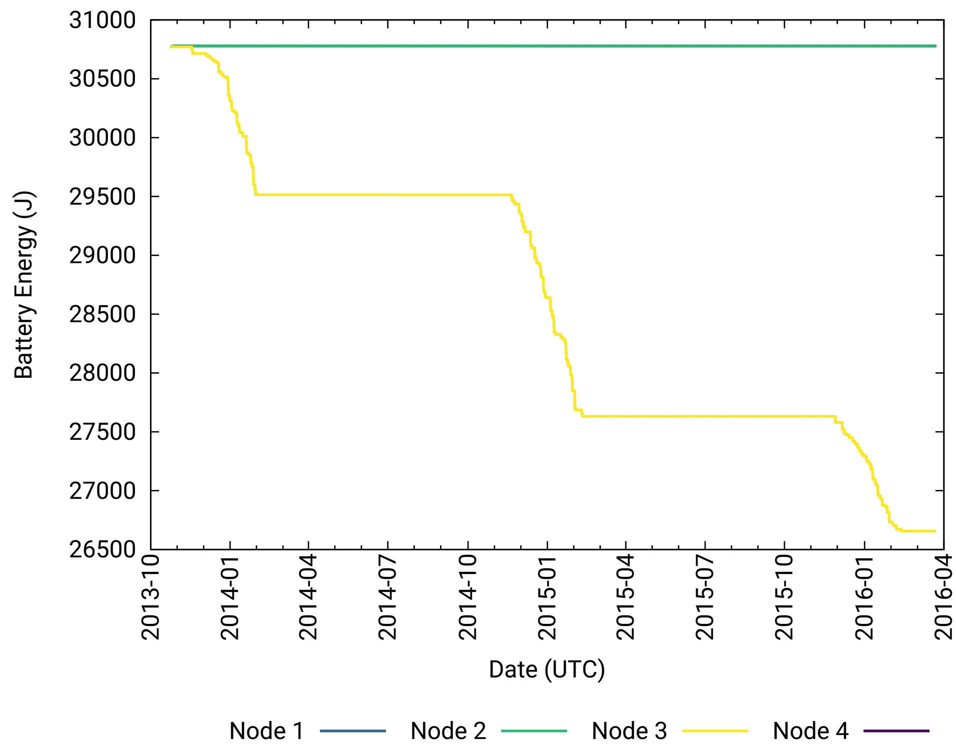


Figure 4.49: Reserve battery energies for unbound optimization, CC13/Perfect.

calculated is dominated by the energy reserve usage.

In order to keep a constant activity, and therefore keep the measurement and transmission frequency constant, the nodes must operate during times of darkness and completely rely on energy stored in the buffer. Where the node is permitted to change its activity level, some cases where a node would be forced to dip into its energy reserve are:

- an activity level is selected for the night period that cannot be supported by the amount of energy in the buffer,
- an activity level is selected for the day that is so high that the amount and rate of incoming solar energy cannot support it and,
- an appropriate activity level is selected for the given period, but other nodes have selected higher levels, and the constraints on receiving messages forces energy usage of the given node.

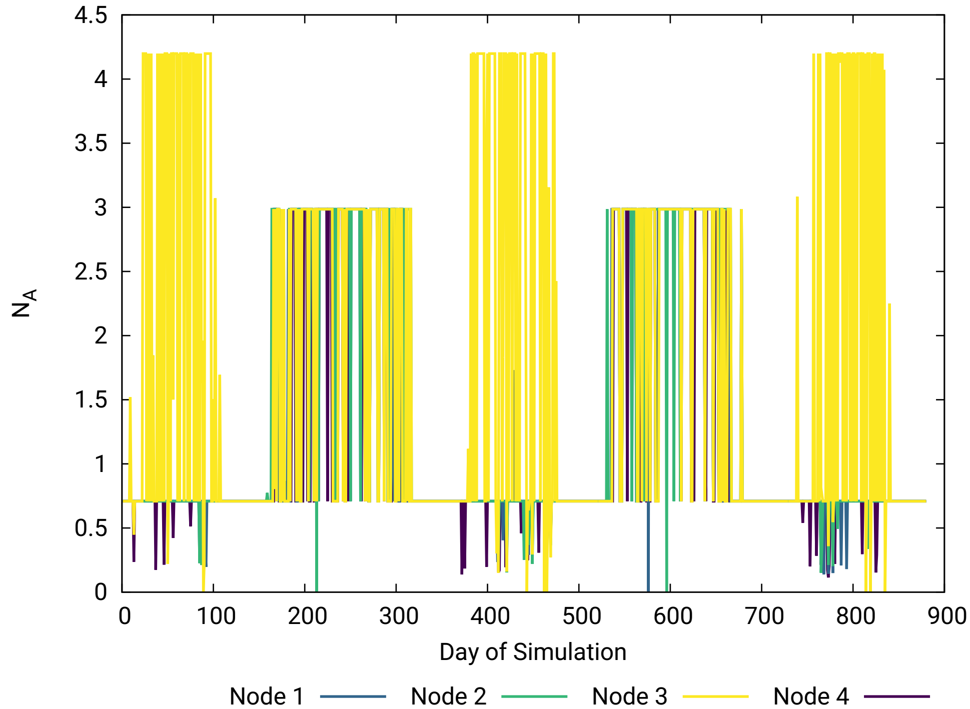


Figure 4.50: Sunrise node activity (N_A) for unbound CC13/Perfect.

Looking at the energy reserve usage broken down by day and night intervals illustrates which case is contributing most to the energy usage for a given simulation. For simulations that used the lowest amount of reserve energy, these plots show that all of this energy usage occurs during the first night of the simulation, because of the initial network activity level prior to the first node activity update point.

For higher node activity levels, the amount of energy usage during the night continued to increase. For example, Fig. 4.7 shows the day and night-time energy reserve usages for a constant node activity of approximately 0.6. This node activity was also the first time where energy reserve usage during the day is experienced, although it was very small in relation to the amount used at night. For even greater node activities, nodes began to fail before the end of the simulation.

A plot of effective N_A values versus total network energy reserve usages for simulated controllers using perfect energy forecasts is shown in Fig. 4.52. In this

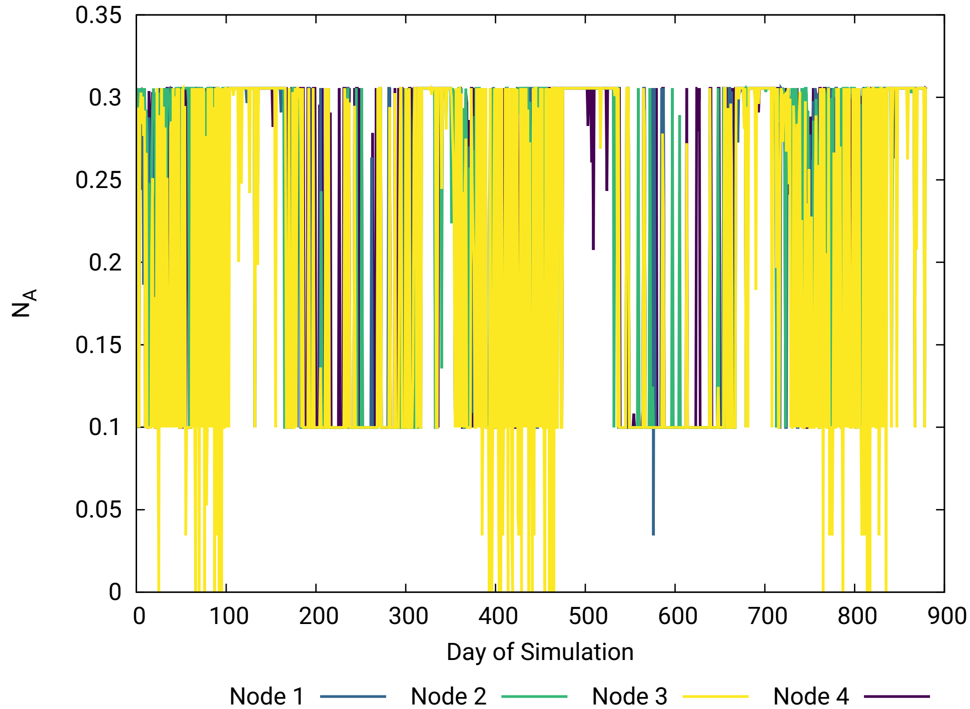


Figure 4.51: Sunset node activity (N_A) for unbound CC13/Perfect.

figure, there is no controller above the solid line that represents the constant N_A simulations, meaning that all of the controllers represented an improvement over the constant case as they did not use more reserve energy, but took a greater amount of total measurements.

Unfortunately, a simulated controller with one of the lowest fitness functions, the unconstrained CC13 case, had both a large effective node activity and energy usage when simulated using a perfect forecast. Since so much emphasis was placed on keeping the use of battery power low during optimization, the fact that this controller used so much suggests that the relationship between fitness of the controller during optimization and the performance of the resulting controller when simulated in a small network is not as correlated as was initially hoped, and that the difference between the optimized node and the simulated network was too great for this scheme to be used effectively.

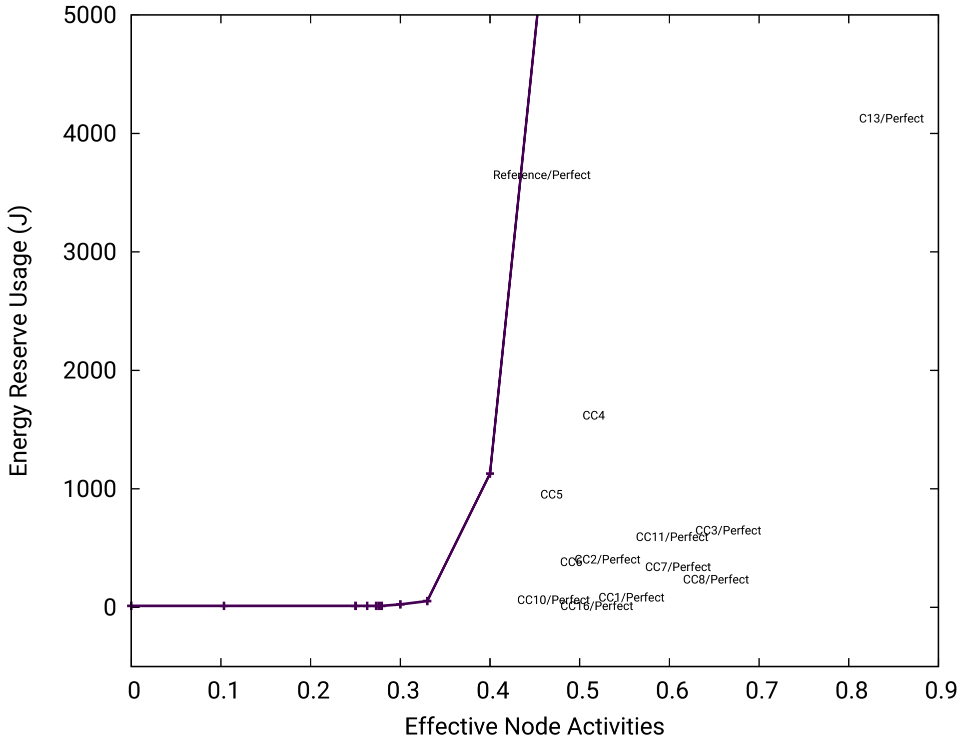


Figure 4.52: Effective activities versus energy reserve usages for select controllers using perfect energy forecasts.

4.5 Chapter Summary

In this chapter, parameters and simplifying assumptions for the simulated wireless sensor nodes were introduced. The defined network was simulated using constant node duty cycles and a human-created reference controller in order to establish baseline performance. A number of fuzzy logic controllers were optimized using differential evolution (Sec. 2.5.4) using different types and numbers of membership functions. These controllers were simulated again in Chapter 5, this time using a forecasting model developed in Chapter 3 in order to examine their performances when perfect foreknowledge of incoming energy is not available.

Chapter 5

Simulation of Controllers using Non-Perfect Forecasts

In Chapter 3, forecasts of daily incoming solar energy were created with measurements of atmospheric pressure used as the predictor variables. Chapter 4 outlined the optimization of number of potential fuzzy controllers for energy management in wireless sensor nodes. For the cases where forecasts were required, perfect forecasts were provided for both the optimization and the full network simulation. In this chapter, the previously developed forecasts based on pressure measurements and the created controllers are combined to examine the effect of the error associated with the data-based forecast on the performance of the optimized controllers. Results from controllers that do not make use of a forecast remain unchanged.

The simplest forecast assumed a static value for the transmissivity factor and allowed the varying value of expected incoming solar energy ($\hat{E}_A$) to dictate the forecast energy value. This static value could be assigned the value of 1, meaning that the entire amount of predicted energy is forecasted as being available for node harvest. This case has the highest potential for causing the usage of reserve energy as there should never be more energy available for harvest than the analytical estimate. The available energy should always be less. Another possible selection for a static transmissivity value is a value that reduces the error between the actual and estimated curve. The resulting performance of the controllers using perfect forecasts have been shown previously, as these forecasts were used in the optimization process. Likely the worst energy forecast that could be used is the $\hat{E}_A$ estimate.

Ideally, the developed forecasts from Chapter 3 would result in performances ranging between this forecast and the perfect forecast used during controller optimization. For ease of implementation, an appropriate CART forecast was used as a representative of the forecasts developed in Chapter 3.

5.0.1 Reference Controllers

The performance for the controller using the static $\hat{E}_A$ forecast is tabulated in Table 5.1 while the values for the simulation where the CART forecast was used are shown in Table 5.2.

Table 5.1: Reference/$\hat{E}_A$ simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	2082.35	741840	392	1114	843
2	1990.01	741076	478	1106	842
3	6095.89	703961	301	1048	800
4	10.85	799927	303	1197	909
Tot/Comb	10179.1	2986804	301	1197	849

Table 5.2: Reference/CART simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	859.07	562120	406	962	639
2	820.89	554138	375	974	630
3	2309.43	536790	390	938	610
4	0.00	595678	203	1075	677
Tot/Comb	3989.39	2248726	203	1075	639

As expected, comparing the energy usage of the reference controller when using the analytical energy estimate $\hat{E}_A$ to the energy usage when a perfect forecast (Table 4.4) is used shows worse performance. The controller resulted in 279% of the total network energy reserve being used, while taking 125% of the total network

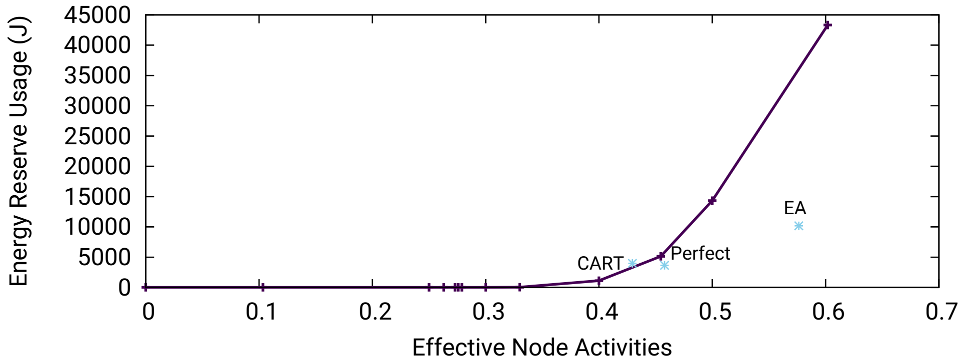


Figure 5.1: Effective activity values for simulations using different forecasts plotted against reserve energy usage for the reference controller. The solid line represents simulations performed using constant values of N_A. Simulations below this line represent an improvement while those represent lower performance than the constant activity case.

measurements. Comparing the performances of one of the pressure-based forecasts to the perfect forecast, we see that this pressure-based forecast results in the usage of 109% of the total network reserve energy, while taking 94% of the total network measurements.

For this controller, the static, optimistic scenario forecast resulted in both more energy usage as well as more measurements, across the entire network. The pressure based forecast resulted in more energy usage with fewer measurements being taken. This is illustrated in Fig. 5.1 where calculated effective activity values are plotted against the reserve energy usage for simulations where the different forecasts were used. This shows that the pressure-based CART forecast was worse than the constant N_A case for this controller. While the $\hat{E}_A$ based forecast was still greatly outperformed by the equivalent constant node activity, the difference between it and the perfect forecast with respect to reserve energy usage suggest that the controller could be more effective with the provided information. However, even using this controller there is already a trade off made when using different controller/forecast pairs.

5.0.2 Optimized Controllers

The results of applying different forecasts to the initial optimizations presented in Sec. 4.4.2 are shown here. When the pressure-based forecasts were used, the error associated with the output had only a very small effect on the overall energy usage for the reference controller while it collected a similar amount of measurements. For the optimized controllers, the use of a pressure-based forecast consistently led to a greater use in total network reserve energy when compared to the simulation of the same controller using a perfect forecast. Unfortunately, this increased energy usage did not translate to a corresponding increase in total network measurements in most cases. However, when compared to the reference controller, the optimized controllers maintained better or comparable performance with respect to at least one of the two metrics (i.e., network energy usage and total network measurements).

5.0.2.1 Initial Optimizations

For the initial D_0 optimization using 5 trapezoidal MF per input (CC1), the tabulated results where $\hat{E}_A$ is used as the forecast is shown in Table 5.3, while the CART pressure-based results are shown in Table 5.4. Compared to the use of the perfect forecast for this controller (Table 4.6), the $\hat{E}_A$ forecast used 77.5% of the energy while taking 90.3% of the total network measurements. Using the CART pressure-based forecast, 384% of the energy is used when compared to the perfect forecast, while taking 93.3% of the total network measurements.

Table 5.3: CC1/$\hat{E}_A$ simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	1.44	654262	499	836	743
2	1.30	655732	597	827	745
3	54.12	659479	487	851	749
4	6.93	645255	487	831	733
Tot/Comb	63.79	2614728	487	851	743

Table 5.4: CC1/CART simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	53.86	681285	476	943	774
2	11.46	671621	472	953	763
3	198.15	717645	551	980	816
4	52.97	630728	338	1032	717
Tot/Comb	316.44	2701279	388	1032	767

Fig. 5.4 shows the battery energy levels for the CC1/CART simulation. Fig. 5.5 shows the number of measurements per day for this simulation.

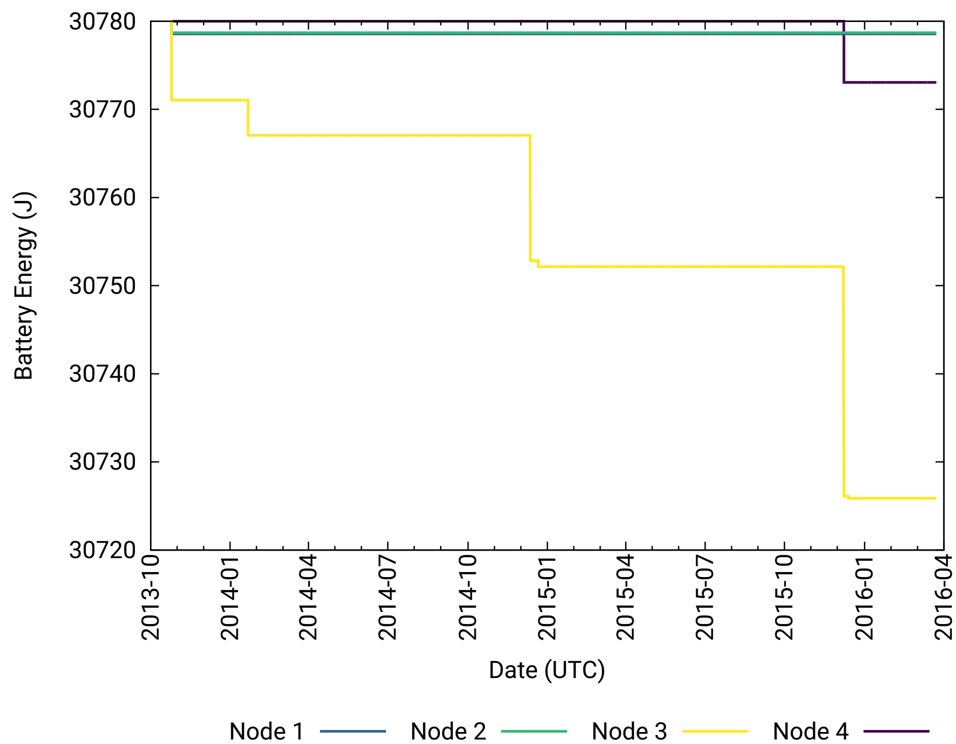


Figure 5.2: Battery reserve energies for the CC1/$\hat{E}_A$ simulation.

The initial optimizations with 3 trapezoidal MF using only the D_0 forecast (CC2) are shown in Table 5.5 using $\hat{E}_A$ as a forecast and Table 5.7 where the CART pressure-based forecast was used. Where both D_0 and D_1 forecasts are used (CC3), the results are shown in Table 5.6 where $\hat{E}_A$ is used as the forecast and in Table 5.8 for the case where the CART pressure-based forecast is used.

Compared to the simulations where perfect forecasts were used, the CC2/$\hat{E}_A$

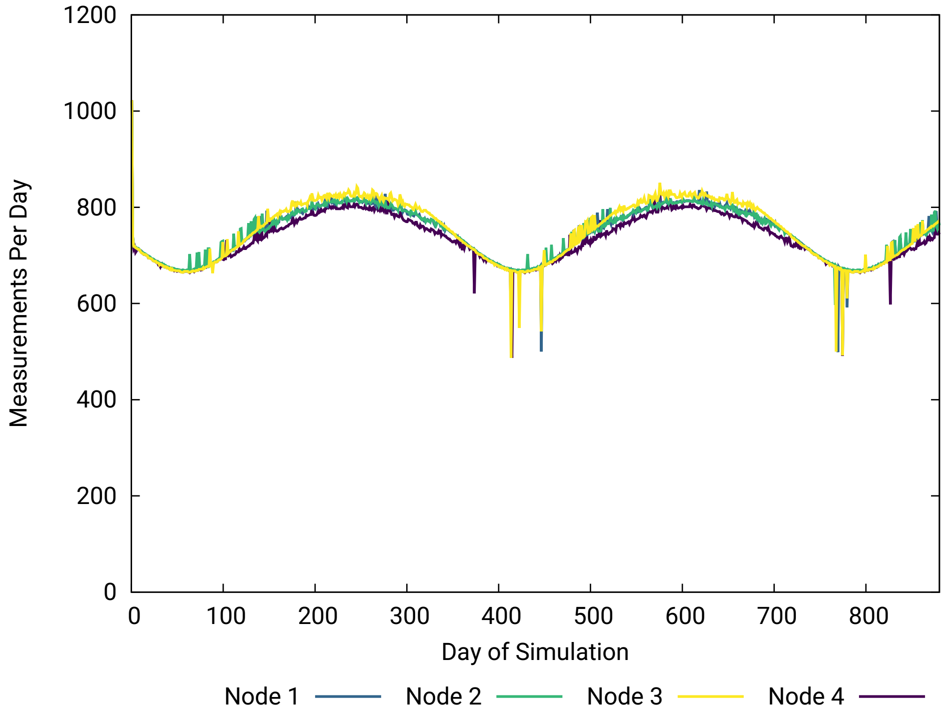


Figure 5.3: Number of measurements per day for the CC1/$\hat{E}_A$ simulation.

simulation resulted in 477% of the total energy usage and 52.4% of the total collected measurements. Using the pressure-based forecast in the CC2/CART simulation resulted in 134% of the total energy usage and 97.5% of the total collected measurements. Using the D_0/D_1 scheme with the CC3 controller, the $\hat{E}_A$ forecast resulted in 96.3% of the total energy usage while collecting 19.3% of the total collected measurements when compared to the perfect forecast simulation. Using the pressure-based forecast, the 101% of the total network measurements were collected while using 187% of the total energy.

Plots of the energy usage for CC2 using the $\hat{E}_A$ and pressure-based forecasts during simulation are shown in Figs. 5.6 and 5.8, respectively. The number of measurements taken per day for these simulations are similarly shown in Figs. 5.7 and 5.9.

Figs. 5.10 and 5.12 show the plots of energy usage for the CC3 controller using the $\hat{E}_A$ and pressure-based forecasts, respectively. The number of measurements

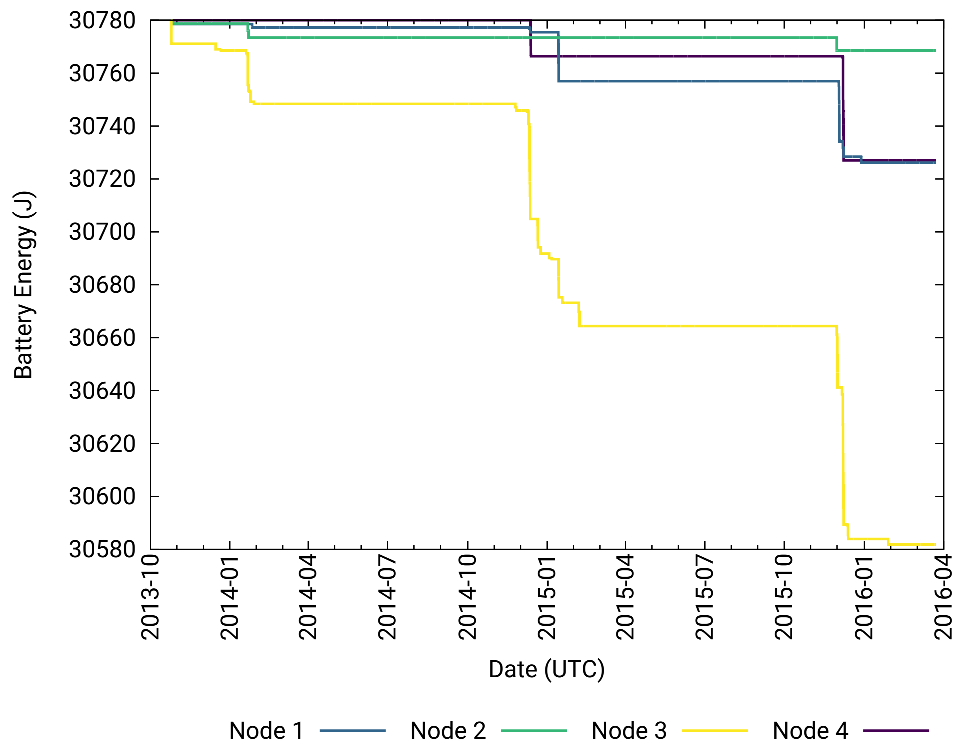


Figure 5.4: Battery reserve energies for the CC1/CART simulation.

Table 5.5: CC2/$\hat{E}_A$ simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	323.15	352165	232	900	400
2	163.74	330444	233	898	376
3	1407.17	462631	230	953	526
4	20.03	301055	230	889	342
Tot/Comb	1914.09	1446295	230	953	411

taken per day during these simulations are similarly shown in Figs. 5.11 and 5.13.

Comparing the battery reserve energies of these two controllers using perfect and pressure-based forecasts results in Figs. 5.14 and 5.15 for CC2 and CC3, respectively. In these plots, negative values represent the deficit of energy of the pressure-based forecast compared to the controller using perfect foreknowledge of the incoming solar energy. In both cases, Node 3 is the most affected by the use of a non-perfect forecast, with the bulk of the energy usage taking place during the winter months.

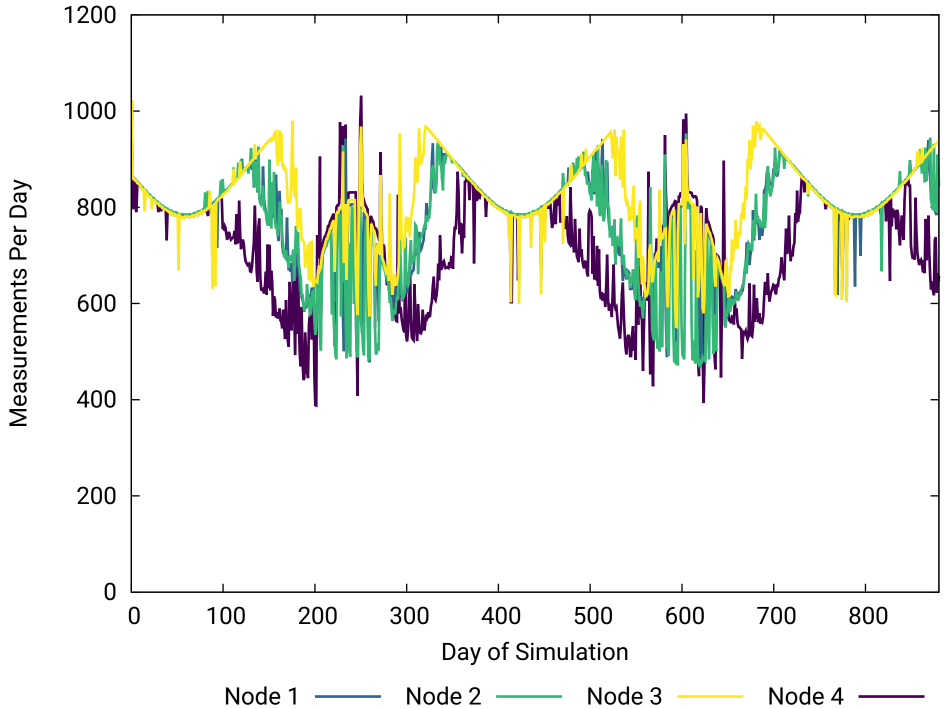


Figure 5.5: Number of measurements per day for the CC1/CART simulation.

Table 5.6: CC3/$\hat{E}_A$ simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	1.44	151601	79	566	172
2	1.30	151583	80	570	172
3	8.94	119029	79	508	135
4	0.00	241272	88	738	274
Tot/Comb	11.68	663485	79	738	188

Effective activities are plotted against network energy reserve usages for the CC1, CC2 and CC3 controllers where different forecasts are used in Fig. 5.16. In these cases, we see how poor forecasts can affect the performance of a controller. For the case of the CC1 controller, the $\hat{E}_A$ forecast reduced the effective activity of the simulated nodes and also slightly reduced the energy reserve usage when compared to the perfect forecast, suggesting that the forecasted information was not being effectively utilized. For the CC2 and CC3 controllers, both are very negatively impacted by the poor forecast, resulting in energy usages higher than that of the

Table 5.7: CC2/CART simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	66.39	684348	435	981	778
2	33.97	684451	435	980	778
3	383.88	614081	416	980	698
4	52.26	706578	116	1023	803
Tot/Comb	536.5	2689458	116	1023	764

Table 5.8: CC3/CART simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	409.16	873819	664	1152	993
2	322.14	873190	661	1148	992
3	1979.28	867107	662	1152	985
4	241.39	874000	645	1154	993
Tot/Comb	2951.97	3488116	645	1154	991

constant N_A case. Not unexpectedly, when the CART forecasts were used, the simulated nodes experience larger energy usages when compared to the perfect case, with the one-day case have a lower N_A value and the two-day case having a larger N_A.

5.0.2.2 Other Optimizations

Results for CC7/$\hat{E}_A$ are tabulated in Table 5.9 while results using the pressure-based CART forecast are shown in Table 5.10. Compared to the performance when the perfect forecast was used, the $\hat{E}_A$ energy estimate used 2037% of the reserve energy while taking 34.6% of the measurements across the whole network. The pressure-based CART energy forecast resulted in 220% of the energy usage while taking 104% of the total network measurements. A plot comparing the energy usage of CC7 when simulated with perfect and pressure-based forecasts is shown in Fig. 5.17. Here, Nodes 3 and 4 were the most affected by the use of the non-perfect forecast, while the other nodes used amounts similar to the perfect forecast case.

Similarly, the simulation results for CC8/$\hat{E}_A$ and CC8/CART are tabulated in Tables 5.11 and 5.12, respectively. Compared to the perfect forecast results, the CC8/$\hat{E}_A$

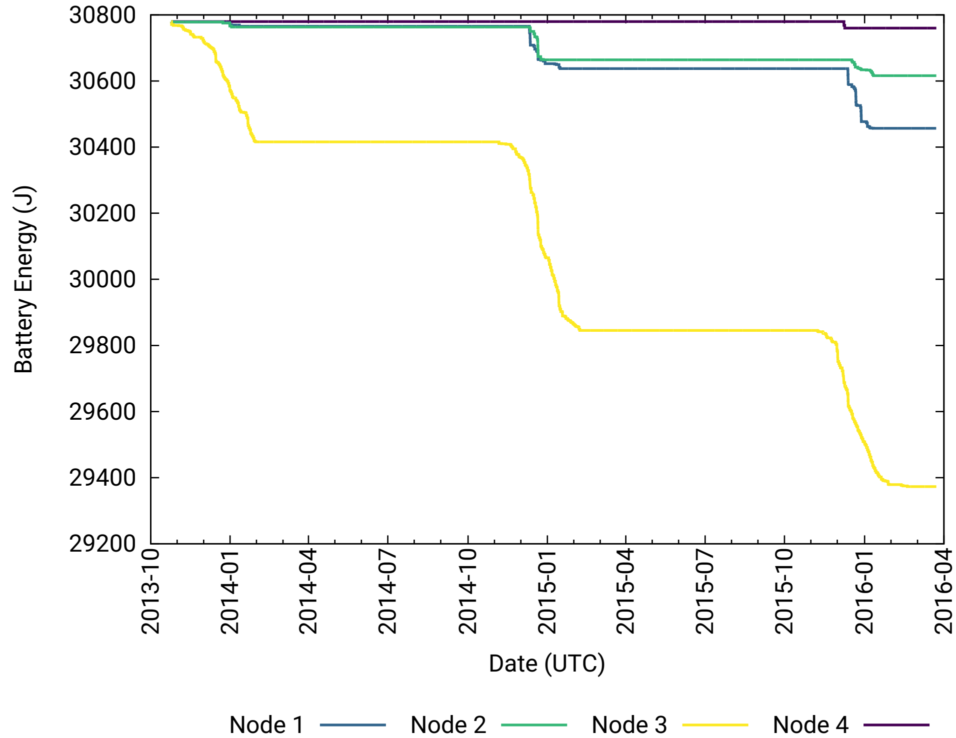


Figure 5.6: Battery reserve energies for the CC2/$\hat{E}_A$ simulation.

simulation resulted in the usage of 4.99% of the total network energy while only taking 2.75% of the total measurements. Where the pressure-based forecast was used, the controller resulted in 136% of the total energy usage and 99.4% of the total measurements of the perfect forecast case. The energy usage of the nodes during simulations using perfect and pressure-based forecasts is shown in Fig. 5.18. In this case, Node 3 is again the most affected by the use of the pressure-based forecast. However, the remaining nodes have energy usages similar to the perfect case.

The controller using the D_0 forecast with no bounds on its node activity, CC13

Table 5.9: CC7/$\hat{E}_A$ simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	1.44	273155	217	386	310
2	1.30	273134	219	385	310
3	8.94	273314	217	387	311
4	0.00	273310	217	387	311
Tot/Comb	11.68	1092913	217	387	311

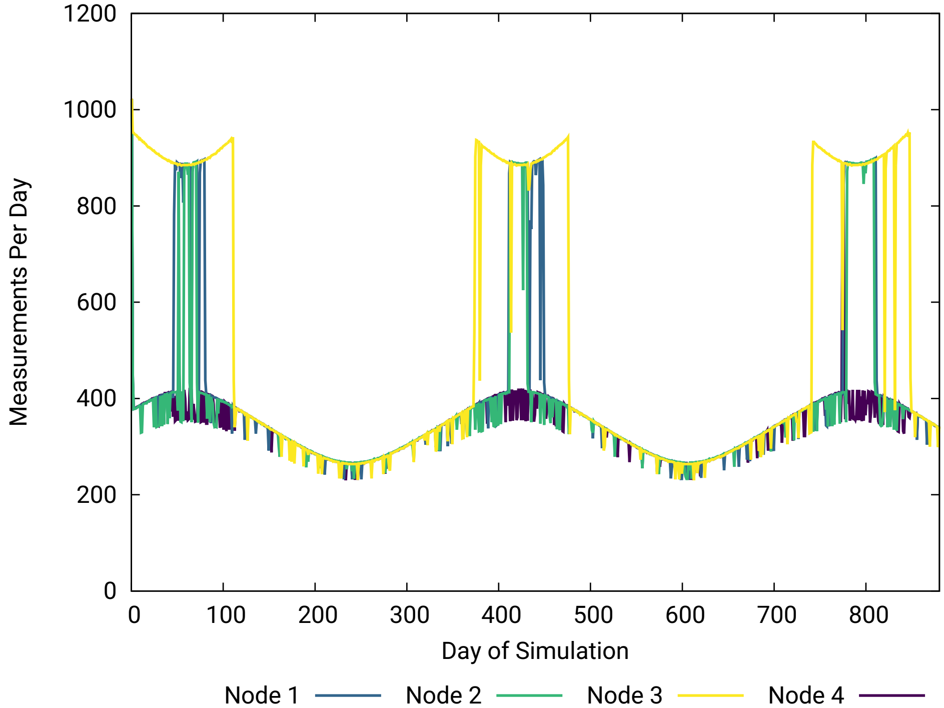


Figure 5.7: Number of measurements per day for the CC2/$\hat{E}_A$ simulation.

Table 5.10: CC7/CART simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	154.25	835042	553	1426	949
2	23.11	838949	563	1136	953
3	528.87	820394	541	1426	932
4	40.21	825500	545	1138	938
Tot/Comb	746.44	3319885	541	1426	943

were simulated using the $\hat{E}_A$ and pressure based forecasts, resulting in the values shown in Tables 5.13 and 5.14, respectively. Compared to this controller simulated using a perfect forecast, the $\hat{E}_A$ forecast resulted in 20.2% of the total measurements, using 0.28% of the total reserve energy. Comparing the simulation using the pressure-based forecast, 76.5% of the total measurements are taken using 136% of the total network energy.

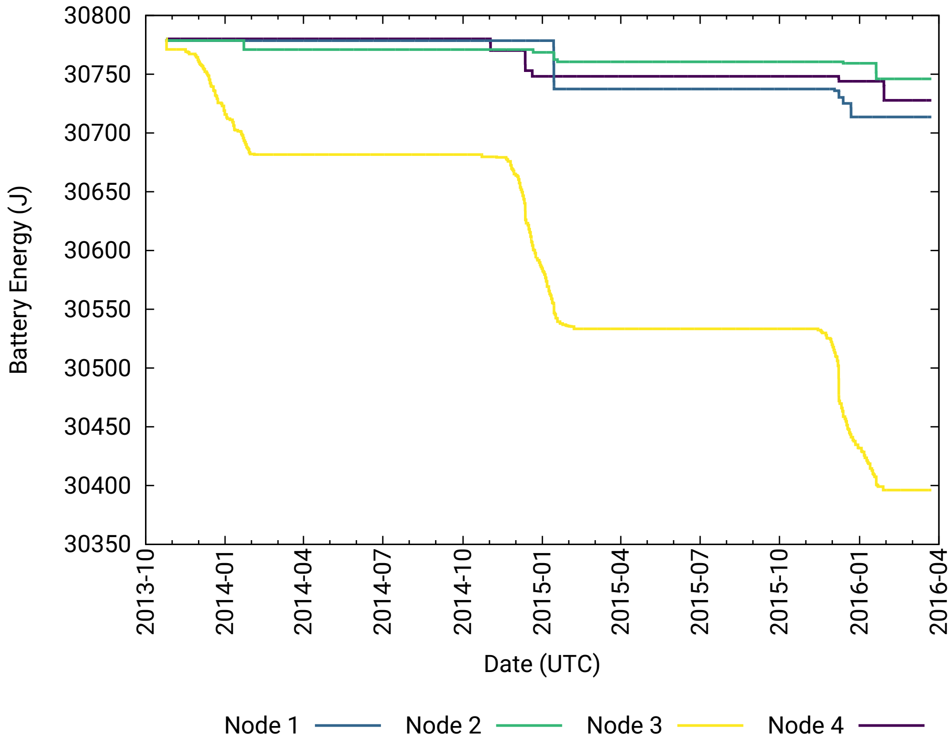


Figure 5.8: Battery reserve energies for the CC2/CART simulation.

Table 5.11: CC8/$\hat{E}_A$ simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	1.44	23149	25	27	26
2	1.30	23153	25	27	26
3	8.94	23142	25	27	26
4	0.00	23136	25	27	26
Tot/Comb	11.68	92580	25	27	26

Table 5.12: CC8/CART simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	5.74	840754	525	1180	955
2	9.50	841406	529	1190	956
3	253.32	818965	412	1268	930
4	48.95	845198	411	1264	960
Tot/Comb	317.51	3346323	411	1268	951

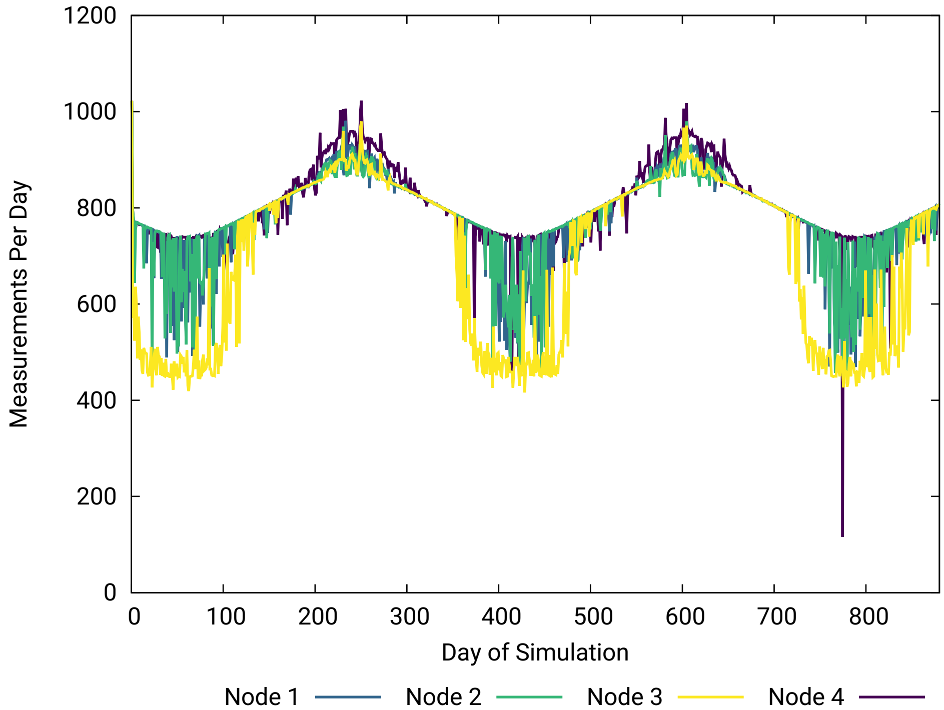


Figure 5.9: Number of measurements per day for the CC2/CART simulation.

Table 5.13: CC13/$\hat{E}_A$ simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	1.44	219227	176	309	249
2	1.30	219170	177	308	249
3	8.94	219232	175	310	249
4	0.00	219224	169	310	249
Tot/Comb	11.69	876853	169	310	249

Table 5.14: CC13/CART simulation results.

Node	E_R (J)	M_T	Min. M_D	Max. M_D	Mean M_D
1	176.43	784023	159	2873	891
2	208.97	671996	163	2865	764
3	5196.35	1044689	100	2885	1187
4	16.63	826504	466	2886	939
Tot/Comb	5598.11	3327212	100	2886	945

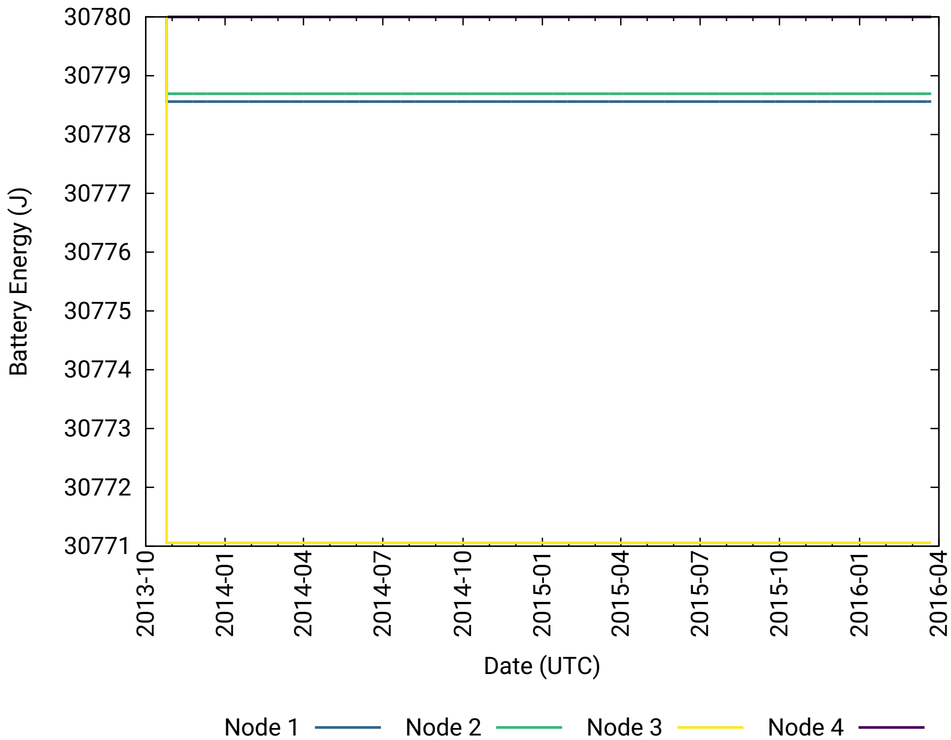


Figure 5.10: Battery reserve energies for the CC3/$\hat{E}_A$ simulation.

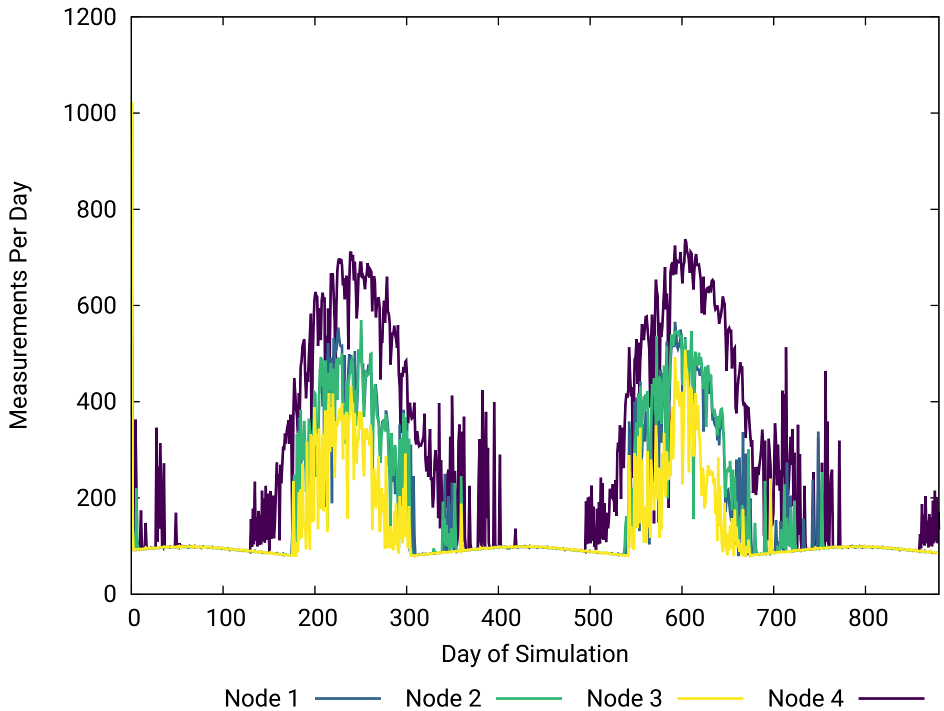


Figure 5.11: Number of measurements per day for the CC3/$\hat{E}_A$ simulation.

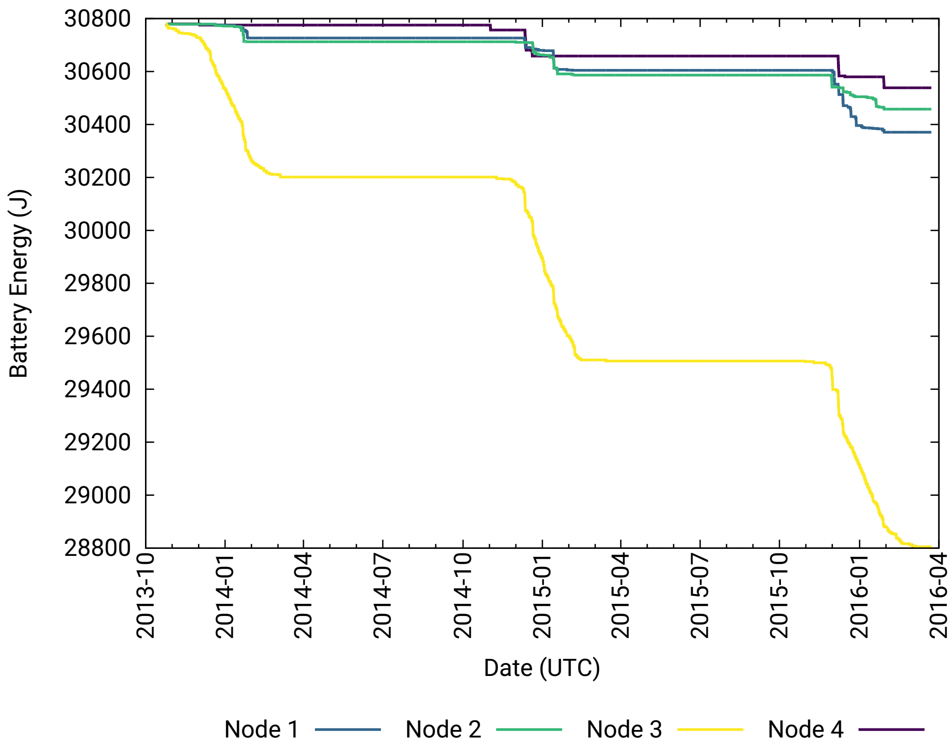


Figure 5.12: Battery reserve energies for the CC3/CART simulation.

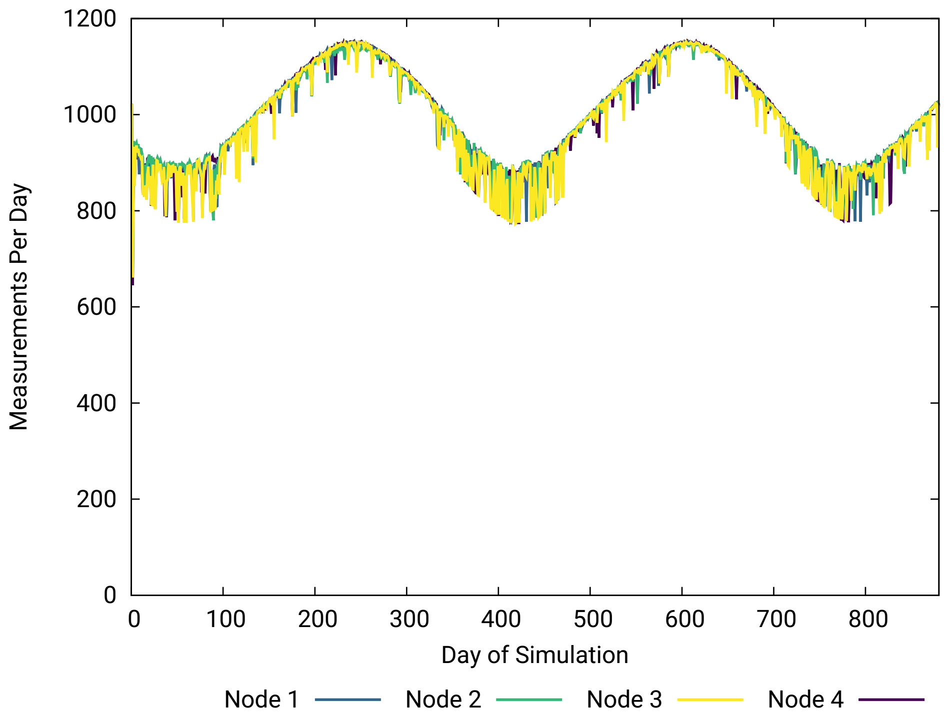


Figure 5.13: Number of measurements per day for the CC3/CART simulation.

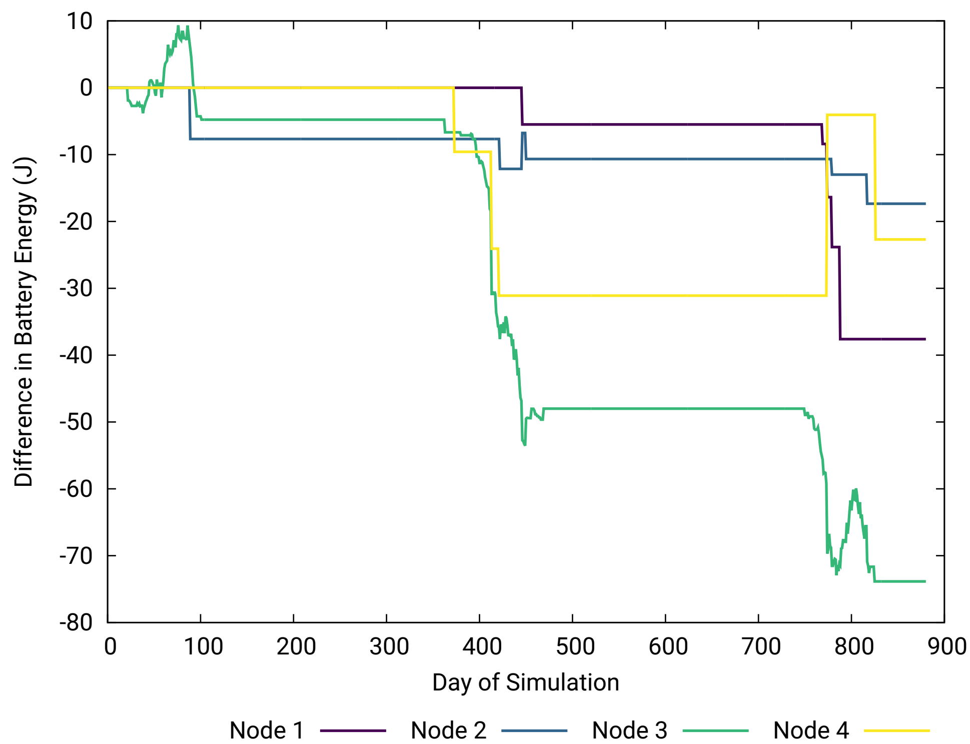


Figure 5.14: Difference between battery backup energies for CC2 controller using perfect and pressure-based forecasts. Values less than zero represent more energy usage of the pressure-based compared to the perfect forecast.

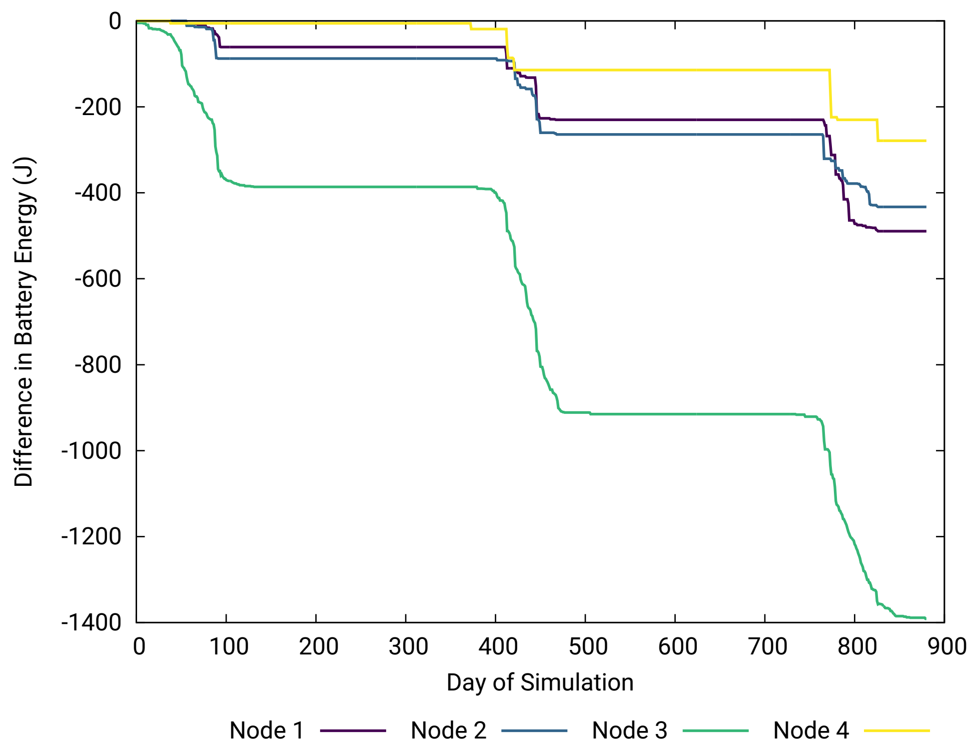
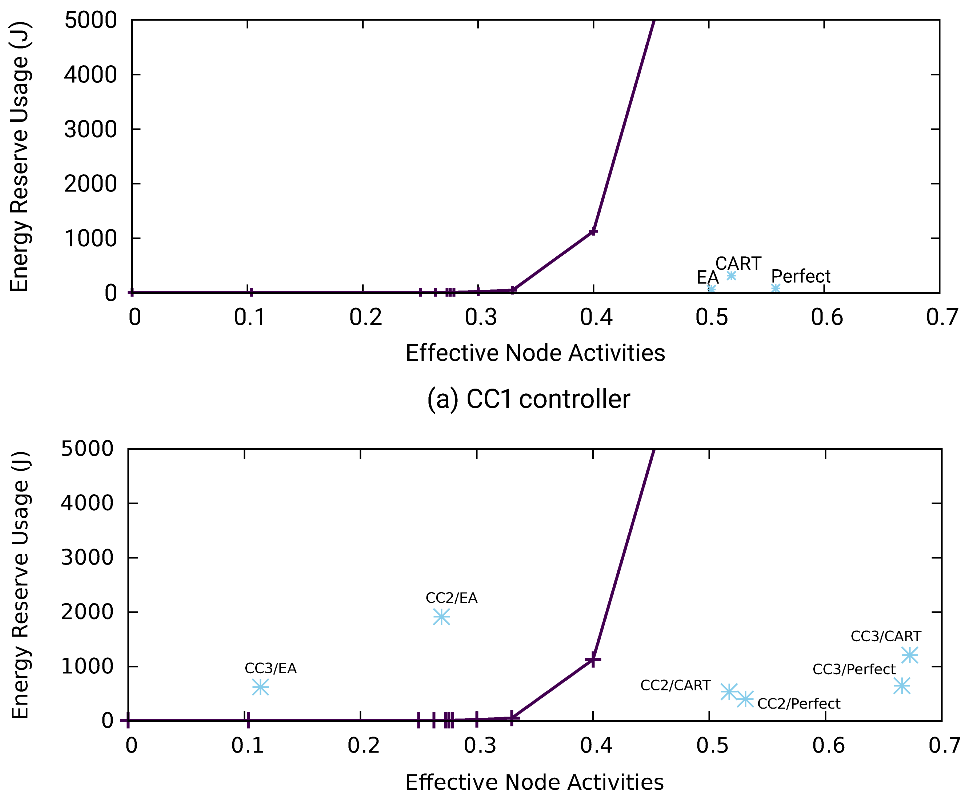


Figure 5.15: Difference between battery backup energies for CC3 controller using perfect and pressure-based forecasts. Values less than zero represent more energy usage of the pressure-based compared to the perfect forecast.



(a) CC1 controller

(b) CC2 and CC3 controllers

Figure 5.16: Effective node activities versus network energy reserve usages for CC1, CC2 and CC3 controllers using different forecasts.

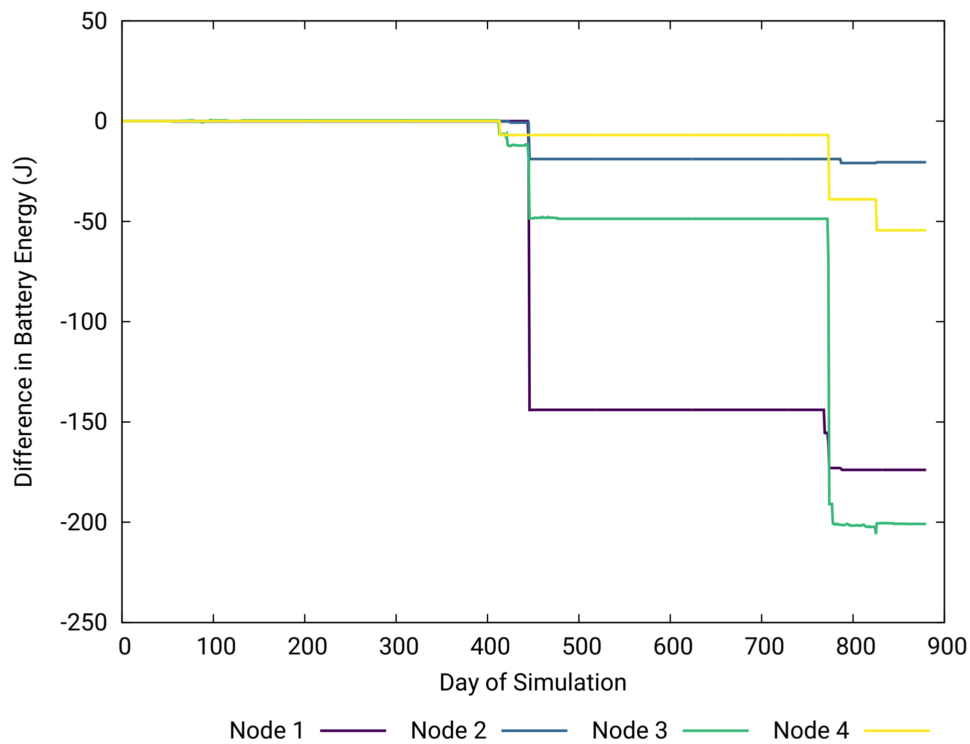


Figure 5.17: Difference between battery backup energies for CC7 controller using perfect and pressure-based forecasts. Values less than zero represent more energy usage of the pressure-based compared to the perfect forecast.

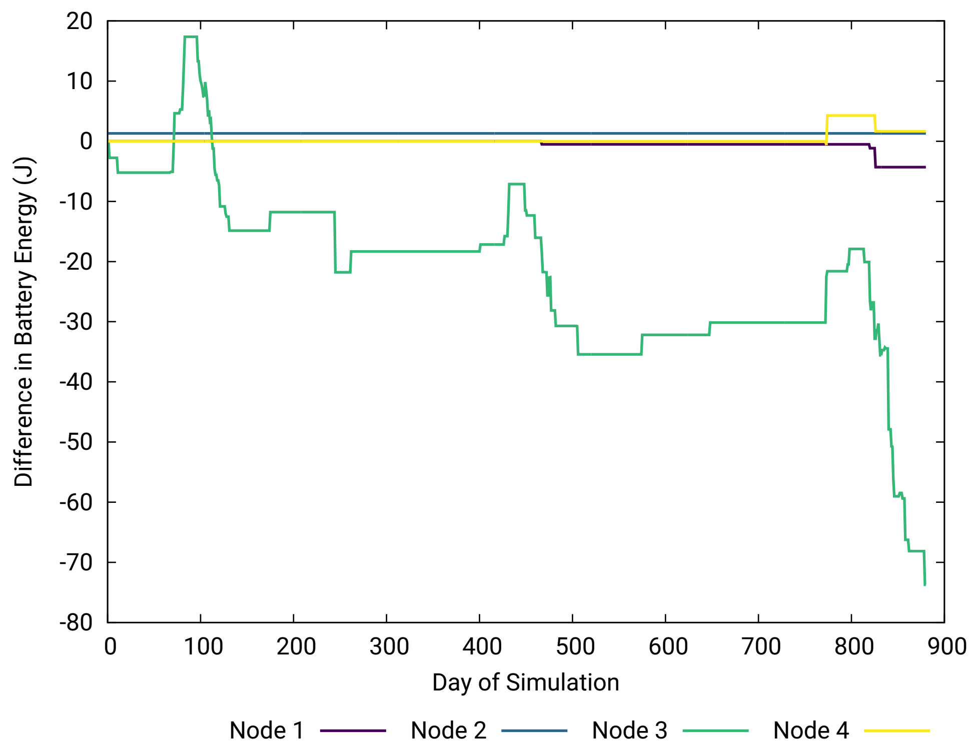


Figure 5.18: Difference between battery backup energies for CC8 controller using perfect and pressure-based forecasts. Values less than zero represent more energy usage of the pressure-based compared to the perfect forecast.

5.1 Comparisons

There are several criteria that should be met to by the candidate controllers. They are:

- For a controller, more accurate forecasts should yield better simulation performance, approaching the performance of the perfect forecast.

- For controllers using a forecast, simulations where a perfect forecast is provided should be objectively equivalent or better than others. If this is not the case, it can be said that the controller is not properly making use of the provided forecasts.

- For the simulation results obtained from each controller/forecast pair, they should ideally all be an improvement over the simulations with fixed node activity values, at least in terms of energy reserve usage. In this way, it is ensured that a controller will not cause the premature failing of a node because of poor forecasts.

- Better controllers will have lower or equal energy usage compared to the simulation of using the constant value of their effective node activity values.

- Better controllers should have higher average activity values, equating to more recorded and transmitted measurements. They will also avoid large periods of low activity values, equating to more even measurements.

Using calculated effective activity values, the various simulations are plotted against their energy reserve usages in Fig. 5.19. The energy reserve usages for the constant node activities have also been placed on this plot. Simulations on the curve are equivalent to the simulation of the given constant node activity, while simulations above the curve are worse. While these cases are few in number, they generally include those using poor forecasts.

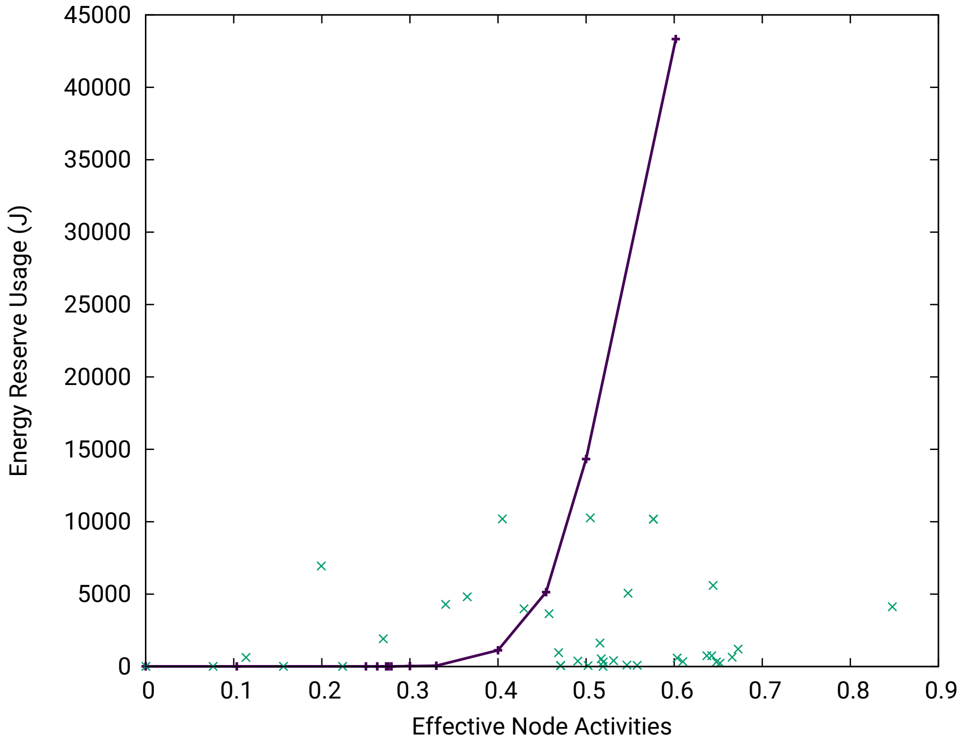


Figure 5.19: Simulation effective activities vs. energy reserve usage. Controller forecasts pairs are shown as points while constant node activities are shown with a solid line.

Of more interest are the simulations lying below the curve, with the best cases being found with higher effective activities and lower reserve energy usages (i.e., further to the bottom right of the plot). Fig. 5.20 provides a magnified region this plot, showing only selected candidate controller/forecast pairs.

With respect to the usage of solar energy, Fig. 5.21 shows the plot of effective node activity levels vs. the amount of unused solar energy for the simulation. In this case, values above the curve demonstrate that the controller/forecast combination was unable to properly exploit the harvestable energy, and their activity level came at the expense of reserve energy usage. As opposed to the plot of reserve energy usage, the unused solar energy generally follows a much more linear trend for the different simulations. With a few exceptions, many of the simulations fell on or below the constant node activity curve. Fig. 5.22 shows the same plot, magnifying

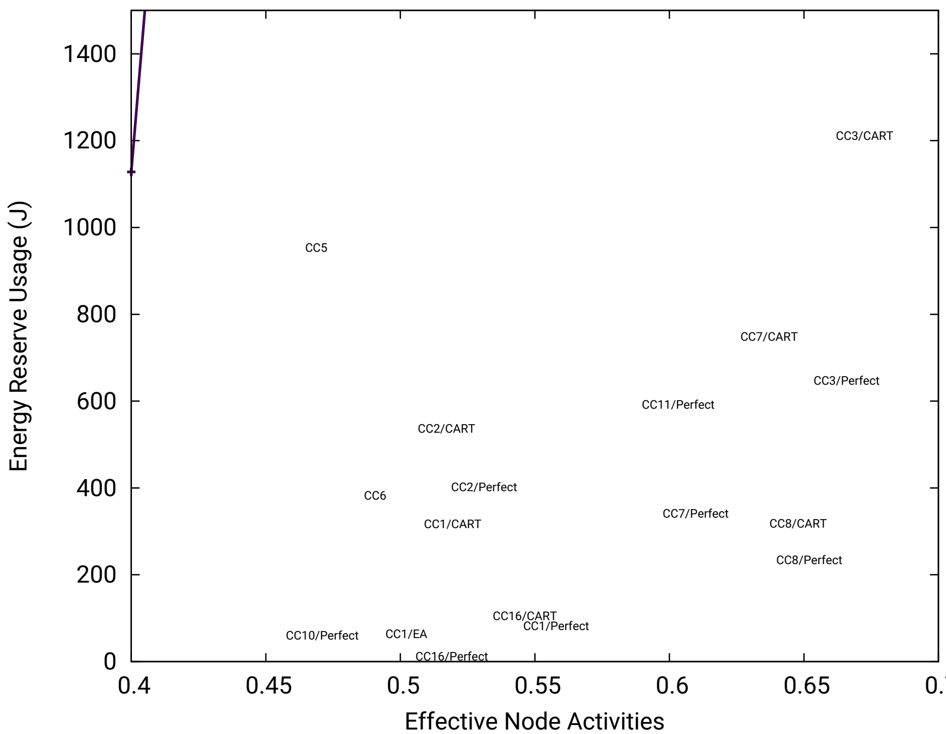


Figure 5.20: Selected simulation effective activities vs. energy reserve usage (magnified).

the lower right area of the plot.

Unfortunately, selection of the best controller is difficult. With some exceptions, the bulk of the optimized controllers outperformed the human-created reference controller by operating at a higher effective activity level, while making more use of available environmental energy and using less of reserve energy. Unfortunately, even comparing controllers created using the same fitness function, there was still a trade off between node activity levels and reserve energy usages. This makes selection of the best controller ultimately dependent on the requirements of a given application.

Even with the fitness function placing a high penalty on the use of reserve energy, when viewed together, there is still an obvious trade off between effective activity and energy usage. Where three forecasts were included, the resulting controllers performed poorly. This is likely due to the energy buffer capacity and general

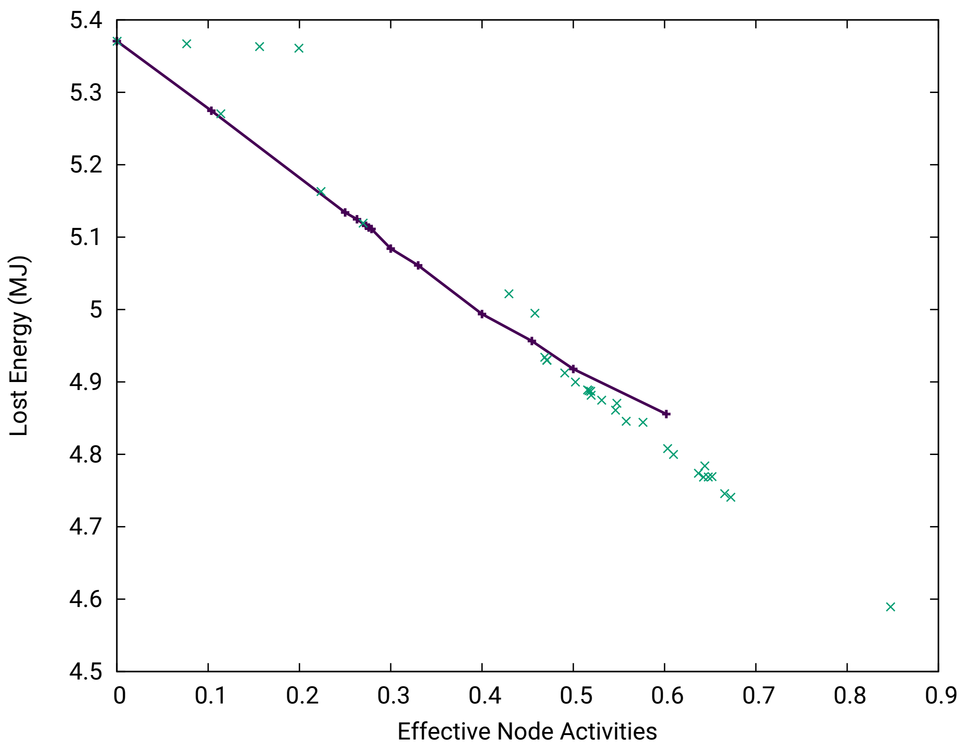


Figure 5.21: Simulation effective activities vs. unused solar energy (constant activity levels shown with solid line).

weather pattern rendering the longer forecasting horizons useless. This, combined with the fitness function penalties for incomplete coverage, led to the case where the additional input was forced to exist, but could do little else but degrade performance.

Tabulated results of the various simulations referenced in this section are available in Table C.1 in Appendix C.

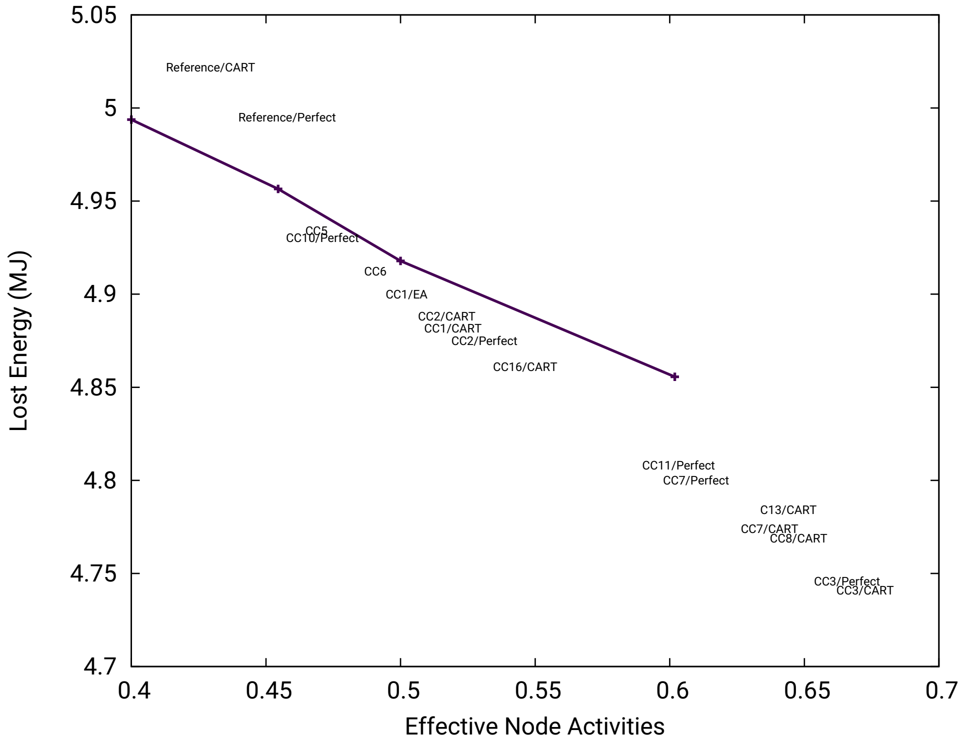


Figure 5.22: Selected simulation effective activities vs. unused solar energy (magnified, constant activity levels shown with solid line).

5.2 Chapter Summary

This chapter showed results combining the developed pressure-based solar energy forecasts from Chapter 3 with the tuned controllers of Chapter 4. Generally, the better performing tuned controllers from Chapter 4 offered better performance than the static node activities and reference controller case. However, it was not always the case where the perfect forecast provided an objectively better performance with respect the network energy usage, as some cases showed an energy/data trade-off.

Chapter 6

Conclusions and Future Work

Forecasts of solar energy can be used to allow a solar energy harvesting sensor node to make the best use of the energy in its environment. To this end, in the preceding chapters, a forecast of daily solar energy based on atmospheric pressure measurements was developed. Forecasts were created for two locations using different techniques, and their performance compared to a perfect forecast. Different schemes were tested, including the use of a limited number of pressure pairs, hourly pressure measurements, and distributed measurements.

Fuzzy logic based controllers using different input values were created to manage the energy use of wireless sensor nodes. The inputs used were the status of an available energy buffer, and various estimates of future values of incoming solar energy. These controllers were optimized using differential evolution by simulating a single node's operation using measured meteorological data. Simulations of the controller applied to a small wireless sensor network were performed, first with perfect forecasts, then with the energy forecasts developed based on atmospheric measurements.

6.1 Conclusions

In this contribution, forecasts of daily values of incoming solar energy were made based on measurements of atmospheric pressure. In order to make forecasts, a transmissivity value was predicted for the upcoming day and multiplied by an analytical estimate of the expected incoming solar energy. The result was an estimated value of the solar energy expected to reach the location over the course of the upcoming daylight hours. Where there were immediately comparable results, the methods used generally performed well. The usefulness of distributed information in improving the accuracy of the predictions was found to be insignificant for the locations examined. Of the applied techniques, multilayer perceptron neural networks were found to perform consistently well. However, in cases of extremely limited hardware, regression trees also provided an improvement over static forecasts.

The tuning of fuzzy logic controllers was carried out using differential evolution. For cases where at least one forecast was used as an input, the number of membership functions necessary in the input fuzzy sets was estimated at less than 5. Optimized fuzzy controllers were found to outperform the human-created reference controller and the case where constant node activity values were used, in terms of both reserve energy usage and overall node activity. Optimized controllers using only the status of the energy buffer as an input also outperformed the reference and constant control cases, but did not outperform the majority of the controllers utilizing a forecast. Attempts at optimizing a controller using more than two days worth of forecast information did not yield positive results, likely because the relative size of the energy buffer and the constraints placed on the coverage of the membership functions in making different fuzzy sets.

6.2 Contributions

This thesis makes a number of contributions to the area of wireless sensor node energy management.

As a component for adaptive node duty cycling, forecasting models of daily values of solar energy from measurements of atmospheric pressure were created using machine learning techniques. The developed models are simple in order to be evaluated on an individual sensor node. This allows a node to create an in-situ energy forecast for use as part of an energy management strategy. The use of solely atmospheric pressure as an input allows for these models to be used with the addition of a single sensor with simple deployment requirements, for cases where other meteorological variables are not required for the main purpose of the WSN. Multiple combinations of different pressure measurements were explored, leading to the conclusion that more measurements were generally better, and that the simplest scheme of creating a single forecast model for use on different nodes can provide acceptable error values. While the models themselves are not likely to be applicable to other locations, the process of creating them is.

The simulation software developed for use in this work was split into two parts, one for the optimization and the other for simulation of the full network. Both are extensible, allowing for the addition of nodes to the network, removal of simplifying assumptions, and the use of different optimization methods. This provides a head-start for future research.

The presented work regarding the tuning of fuzzy logic controllers for duty cycle control of sensor nodes outlines a working method of performing this optimization that may be applied to other networks and locations. Inclusion of different numbers of forecasts was explored and found to be beneficial up to a point determined by the relative size of the energy buffer when compared to the node's highest daily energy usages. Additionally, the experimentation with different numbers of membership functions provides a starting point for performing the tuning for other cases.

Application of the developed energy forecasting models to the tuned controllers demonstrates the ability of the tuned controllers to provide adaptive control in instances of non-perfect forecasts. A method of comparing the performance of different controller/forecast pairs has also been presented, allowing for the evaluation of the potential energy usage/data trade-off that would be experienced during a deployment.

Overall, an entire process of providing adaptive, energy-aware duty cycling for wireless sensor nodes has been shown. It began with the creation of an energy forecast, which provides an estimate of the energy available to be rationed for upcoming days, and moved to the tuning of a controller in order to make the best use of that energy estimate. The process ended with a number of simulations combining those two parts in order to validate this combination. With availability of the necessary meteorological data and information regarding the energy requirements of the nodes, this process can be applied to any number of different networks and locations.

6.3 Future Work

There are a number of avenues for improvement of the strategies presented in this thesis. While improvements to the simple forecast method and fuzzy controller will improve node performance within the simulations, improvements to the simulations themselves will allow them to be more representative of deployment conditions, and therefore be more useful in developing energy management strategies.

One important improvement to the method of controller optimization would be the automatic determination of the relative weighting of the lost energy and battery reserve usage in the fitness function. The importance of these two values will vary depending on a number of factors such as the expected amount of harvestable energy (e.g., tropical vs. arctic environments), the capacity of the energy buffer, and the desired length of deployment. Estimating these coefficients would be an important part of applying similar optimizations to controllers for use in other regions, including more advanced techniques for the optimization of fuzzy logic controllers could be included (e.g., the automatic merging of similar fuzzy sets and rules).

With respect to node energy usage, further improvement might be expected if the state of the data buffer was included as a third input variable. Allowing measurement and transmission rates to be determined separately may also be helpful. This would allow for a more flexible balance between the costly transmission operations and inexpensive measurements, and ultimately yield a more effective usage of available energy. Computational optimization of the controller becomes more important as the number of variables, and the complexity of relationships between them, increase beyond what may be manually programmed.

In regard to the solar energy forecast for actual deployment conditions, one detrimental effect on the ability of solar cells to harvest energy is the presence of objects between panel and the sun. Important objects that can occlude sunlight, especially in outdoor installations are dirt, dust and plant foliage. For the case of dirt and dust, a cumulative decrease in the amount of solar energy may be experienced, while the impact of foliage may be a seasonal effect. Models of these phenomenon could be integrated into the simulation to make it more realistic. With these effects integrated, the energy forecast could be extended to include a more adaptive element, such as an extra transmissivity constant controlled by a moving average, able to correct the pressure-based forecast for these factors.

The realism of the simulations can be improved by applying more sophisticated node charging models by accounting for the nonlinear behaviour of the solar panels, including the costs of computation, and more strict enforcement the limited data storage present on the nodes [33]. Additional improvements to simulator realism could be a transmission energy cost that varies with node separation distance and an imperfect communication channel. More advanced models of other sensor node components such as the supercapacitors used as an energy buffer could also be considered [133].

Additional gains from adaptive duty cycle control may be realized by performing more frequent updates of node activity. In the presented scheme, performing updates at sunrise and sunset allows a node to ration the use of energy currently in the buffer and of that expected to be harvested before the next scheduled update. More frequent updates would require changes to how the forecast is handled, but would be better able to adapt the to rate of incoming energy, errors in the forecast, and changing weather conditions over the course of the day.

Finally, in further support of complete energy management strategy development, simulation of denser networks could be explored. These would allow for the application of energy-aware message routing algorithms to test their interaction with the presented duty cycle control.

References

[1] M. Prauzek, P. Krömer, J. Rodway, and P. Musilek, "Differential evolution of fuzzy controller for environmentally-powered wireless sensors," *Applied Soft Computing*, vol. 48, pp. 193–206, Nov. 2016.

[2] S.-H. Yang, *Wireless sensor networks: principles, design and applications*, ser. Signals and Communication Technology. London: Springer, 2014.

[3] J. A. Stankovic and T. He, "Energy management in sensor networks," *Phil. Trans. R. Soc. A*, vol. 370, no. 1958, pp. 52–67, Jan. 2012.

[4] W. Seah, Z. A. Eu, and H. Tan, "Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges," in *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009*, May 2009, pp. 1–5.

[5] D. Pimentel and P. Musilek, "Power management with energy harvesting devices," in *2010 23rd Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2010, pp. 1–4.

[6] D. Pimentel, P. Musilek, and A. Knight, "Energy harvesting simulation for automatic arctic monitoring stations," in *2010 IEEE Electric Power and Energy Conference (EPEC)*, Aug. 2010, pp. 1–6.

[7] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs, "Long-duration solar-powered wireless sensor networks," in *Proceedings of the 4th Workshop on Embedded Networked Sensors*. ACM Press, 2007, p. 33.

[8] C. Renner, S. Unterschütz, V. Turau, and K. Römer, "Perpetual Data Collection with Energy-Harvesting Sensor Networks," *ACM Trans. Sen. Netw.*, vol. 11, no. 1, pp. 12:1–12:45, Sep. 2014.

[9] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, 2005.

[10] A. Alemdar and M. Ibnkahla, "Wireless sensor networks: Applications and challenges," in *9th International Symposium on Signal Processing and Its Applications, 2007. ISSPA 2007*, Feb. 2007, pp. 1–6.

[11] S. Gajjar, S. Pradhan, and K. Dasgupta, “Wireless Sensor Network: Application led research perspective,” in *2011 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, Sep. 2011, pp. 025–030.

[12] A. Watts, M. Prauzek, P. Musilek, E. Pelikan, and A. Sanchez-Azofeifa, “Fuzzy power management for environmental monitoring systems in tropical regions,” in *2014 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2014, pp. 1719–1726.

[13] E. Fadel, V. C. Gungor, L. Nassef, N. Akkari, M. G. A. Malik, S. Almasri, and I. F. Akyildiz, “A survey on wireless sensor networks for smart grid,” *Computer Communications*, vol. 71, pp. 22–33, Nov. 2015.

[14] K. Martinez, P. Padhy, A. Elsaify, G. Zou, A. Riddoch, J. K. Hart, and H. L. R. Ong, “Deploying a sensor network in an extreme environment,” in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC’06)*, vol. 1, Jun. 2006, p. 8.

[15] D. Niyato, E. Hossain, M. Rashid, and V. Bhargava, “Wireless sensor networks with energy harvesting technologies: a game-theoretic approach to optimal energy management,” *IEEE Wireless Communications*, vol. 14, no. 4, pp. 90–96, Aug. 2007.

[16] A. Watts, P. Musilek, and L. Wyard-Scott, “Managing the energy-for-data exchange in remote monitoring systems,” in *2013 IEEE Electrical Power Energy Conference (EPEC)*, Aug. 2013, pp. 1–4.

[17] J. A. Khan, H. K. Qureshi, and A. Iqbal, “Energy management in Wireless Sensor Networks: A survey,” *Computers & Electrical Engineering*, vol. 41, pp. 159–176, Jan. 2015.

[18] C. Kompis and P. Sureka, “Power management technologies to enable remote and wireless sensing,” May 2010.

[19] A. Kansal, J. Hsu, M. Srivastava, and V. Raghunathan, “Harvesting Aware Power Management for Sensor Networks,” in *Proceedings of the 43rd Annual Design Automation Conference*, ser. DAC ’06. New York, NY, USA: ACM, 2006, pp. 651–656.

[20] F. Chen, L. Guo, and C. Chen, “A Survey on Energy Management in the Wireless Sensor Networks,” *IERI Procedia*, vol. 3, pp. 60–66, 2012.

[21] P. Gyorke and B. Pataki, “Application of energy-harvesting in wireless sensor networks using predictive scheduling,” in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, May 2012, pp. 582–587.

[22] O. S. Ezeora, J. Heckenbergerova, P. Musilek, and J. Rodway, “Sampling control in environmental monitoring systems using recurrent neural networks,” in *2016 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, May 2016, pp. 1–4.

[23] O. S. Ezeora, J. Heckenbergerova, P. Musilek, and P. Krömer, "Evaluation of Errors Caused by Lowering Frequency of Measurements of an Environmental Sensor Station," in *2015 International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, Sep. 2015, pp. 204–210.

[24] K.-Y. Shin, J. Song, J. Kim, M. Yu, and P. S. Mah, "REAR: Reliable Energy Aware Routing Protocol for Wireless Sensor Networks," in *The 9th International Conference on Advanced Communication Technology*, vol. 1, Feb. 2007, pp. 525–530.

[25] P. Zhang, G. Xiao, and H. Tan, "A preliminary study on lifetime maximization in clustered wireless sensor networks with energy harvesting nodes," in *Communications and Signal Processing (ICICS) 2011 8th International Conference on Information*, Dec. 2011, pp. 1–5.

[26] M. Akbar, N. Javaid, M. Imran, N. Amjad, M. I. Khan, and M. Guizani, "Sink mobility aware energy-efficient network integrated super heterogeneous protocol for WSNs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–19, Feb. 2016.

[27] P. H. Zadeh, C. Schlegel, and M. MacGregor, "Distributed optimal dynamic base station positioning in wireless sensor networks," *Computer Networks*, vol. 56, no. 1, pp. 34–49, 2012.

[28] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, Sep. 2007.

[29] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Fourth International Symposium on Information Processing in Sensor Networks, 2005. IPSN 2005*, Apr. 2005, pp. 463–468.

[30] D. Pimentel, "Energy management for automatic monitoring stations in arctic regions," PhD Thesis, University of Alberta, 2012.

[31] D. J. Vergados and G. I. Stassinopoulos, "Adaptive Duty Cycle Control for Optimal Stochastic Energy Harvesting," *Wireless Personal Communications*, vol. 68, no. 1, pp. 201–212, Nov. 2011.

[32] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, and V. Raghunathan, "Adaptive Duty Cycling for Energy Harvesting Systems," in *Proceedings of the 2006 International Symposium on Low Power Electronics and Design*, ser. ISLPED '06. New York, NY, USA: ACM, 2006, pp. 180–185.

[33] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Adaptive Power Management for Environmentally Powered Systems," *IEEE Transactions on Computers*, vol. 59, no. 4, pp. 478–491, Apr. 2010.

[34] N. Ferry, S. Ducloyer, N. Julien, and D. Jutel, "Energy estimator for weather forecasts dynamic power management of wireless sensor networks," in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization,*

and Simulation, ser. Lecture Notes in Computer Science, J. L. Ayala, B. García-Cámara, M. Prieto, M. Ruggiero, and G. Sicard, Eds. Springer Berlin Heidelberg, Jan. 2011, no. 6951, pp. 122–132.

[35] H. Yoo, M. Shim, and D. Kim, "Dynamic Duty-Cycle Scheduling Schemes for Energy-Harvesting Wireless Sensor Networks," *IEEE Communications Letters*, vol. 16, no. 2, pp. 202–204, Feb. 2012.

[36] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-time scheduling for energy harvesting sensor nodes," *Real-Time Systems*, vol. 37, no. 3, pp. 233–260, Jul. 2007.

[37] M. Prauzek, A. G. Watts, P. Musilek, L. Wyard-Scott, and J. Koziorek, "Simulation of adaptive duty cycling in solar powered environmental monitoring systems," in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2014, pp. 1–6.

[38] M. Prauzek, P. Musilek, and A. Watts, "Fuzzy algorithm for intelligent wireless sensors with solar harvesting," in *2014 IEEE Symposium on Intelligent Embedded Systems (IES)*, Dec. 2014, pp. 1–7.

[39] P. Krömer, M. Prauzek, and P. Musilek, "Harvesting-aware control of wireless sensor nodes using fuzzy logic and differential evolution," in *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking Workshops (SECON Workshops)*, Jun. 2014, pp. 51–56.

[40] Y. Zhang, S. He, J. Chen, Y. Sun, and X. S. Shen, "Distributed Sampling Rate Control for Rechargeable Sensor Nodes with Limited Battery Capacity," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 3096–3106, Jun. 2013.

[41] A. Kansal and M. B. Srivastava, "An environmental energy harvesting framework for sensor networks," in *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003. ISLPED '03*, Aug. 2003, pp. 481–486.

[42] M. A. Green, K. Emery, Y. Hishikawa, W. Warta, and E. D. Dunlop, "Solar cell efficiency tables (Version 45)," *Progress in Photovoltaics: Research and Applications*, vol. 23, no. 1, pp. 1–9, Jan. 2015.

[43] D. Myers, *Solar Radiation: Practical Modeling for Renewable Energy Applications*, ser. Energy and the Environment. CRC Press, Mar. 2013.

[44] K.-N. Liou, *An Introduction to Atmospheric Radiation*, ser. International Geophysics Series. Amsterdam: Academic Press, 2002, vol. 2nd ed.

[45] P. A. Lynn, *Electricity from Sunlight : An Introduction to Photovoltaics*. Chichester :: Wiley,, 2010.

[46] J. Twidell and T. Weir, *Renewable Energy Resources*. Taylor & Francis, Oct. 2003.

[47] R. B. Stull, *Meteorology for Scientists and Engineers*. Brooks/Cole, 2000.

[48] W. I. Milham, *Meteorology; a text-book on the weather, the causes of its changes, and weather forecasting, for the student and general reader*. New York Macmillan, 1912.

[49] E. Brewer, *A guide to the scientific knowledge of things familiar*. C. S. Francis & Co., 1851.

[50] W. M. Organization, *Guide to meteorological instruments and methods of observation 7 ed*, ser. WMO (Series). Geneva, Switzerland: Secretariat of the World Meteorological Organization, 2008.

[51] H. B. Diab and J. J. Saade, "Weather Prediction Using Fuzzy Sets and Inference Methodology," *J. Intell. Fuzzy Syst.*, vol. 7, no. 3, pp. 283–305, Aug. 1999.

[52] J.-L. Chen, H.-B. Liu, W. Wu, and D.-T. Xie, "Estimation of monthly solar radiation from measured temperatures using support vector machines – A case study," *Renewable Energy*, vol. 36, no. 1, pp. 413–420, Jan. 2011.

[53] L. Wang, O. Kisi, M. Zounemat-Kermani, G. A. Salazar, Z. Zhu, and W. Gong, "Solar radiation prediction using different techniques: model evaluation and comparison," *Renewable and Sustainable Energy Reviews*, vol. 61, pp. 384–397, Aug. 2016.

[54] L. Wang, O. Kisi, M. Zounemat-Kermani, Z. Zhu, W. Gong, Z. Niu, H. Liu, and Z. Liu, "Prediction of solar radiation in China using different adaptive neuro-fuzzy methods and M5 model tree," *International Journal of Climatology*, pp. 1141–1155, May 2016.

[55] R. D. Jong and D. W. Stewart, "Estimating global solar radiation from common meteorological observations in western Canada," *Canadian Journal of Plant Science*, vol. 73, no. 2, pp. 509–518, Apr. 1993.

[56] M.-F. Li, H.-B. Liu, P.-T. Guo, and W. Wu, "Estimation of daily solar radiation from routinely observed meteorological data in Chongqing, China," *Energy Conversion and Management*, vol. 51, no. 12, pp. 2575–2579, Dec. 2010.

[57] Z. Samani, G. Hargreaves, V. Tran, and S. Bawazir, "Estimating solar radiation from temperature with spatial and temporal calibration," *Journal of Irrigation and Drainage Engineering*, no. 11, p. 692, 2011.

[58] J. Almorox, "Estimating global solar radiation from common meteorological data in Aranjuez, Spain," *Turkish Journal of Physics*, vol. 35, no. 1, pp. 53–64, Mar. 2011.

[59] M. S. Adaramola, "Estimating global solar radiation using common meteorological data in Akure, Nigeria," *Renewable Energy*, vol. 47, pp. 38–44, Nov. 2012.

[60] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct. 2011, pp. 528–533.

[61] P. Krömer, P. Musilek, J. Rodway, M. Reformat, and M. Prauzek, "Estimating Harvestable Solar Energy from Atmospheric Pressure Using Support Vector Regression," in *2015 International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, Sep. 2015, pp. 192–199.

[62] P. Musilek, P. Krömer, J. Rodway, and M. Prauzek, "Pressure-based forecasting of next-day solar energy availability using evolutionary fuzzy rules," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Aug. 2015, pp. 1–8.

[63] F. Wang, Z. Zhen, Z. Mi, H. Sun, S. Su, and G. Yang, "Solar irradiance feature extraction and support vector machines based weather status pattern recognition model for short-term photovoltaic power forecasting," *Energy and Buildings*, vol. 86, pp. 427–438, Jan. 2015.

[64] S. Mohanty, P. K. Patra, and S. S. Sahoo, "Prediction and application of solar radiation with soft computing over traditional and conventional approach – A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 778–796, Apr. 2016.

[65] R. Iqdour and A. Zeroual, "Prediction of daily global solar radiation using fuzzy systems," *International Journal of Sustainable Energy*, vol. 26, no. 1, pp. 19–29, Mar. 2007.

[66] C. Chen, S. Duan, T. Cai, and B. Liu, "Online 24-h solar power forecasting based on weather type classification using artificial neural network," *Solar Energy*, vol. 85, no. 11, pp. 2856–2870, Nov. 2011.

[67] H.-T. Yang, C.-M. Huang, Y.-C. Huang, and Y.-S. Pai, "A Weather-Based Hybrid Method for 1-Day Ahead Hourly Forecasting of PV Power Output," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 3, pp. 917–926, Jul. 2014.

[68] Z. Jiang, X. Jin, and Y. Zhang, "A Weather-Condition Prediction Algorithm for Solar-Powered Wireless Sensor Nodes," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, Sep. 2010, pp. 1–4.

[69] J. R. Piorno, C. Bergonzini, D. Atienza, and T. S. Rosing, "Prediction and management in energy harvested wireless sensor nodes," in *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, May 2009, pp. 6–10.

[70] S. Kalogirou, *Solar energy engineering: Processes and Systems*. Academic Press, 2013.

[71] S. Singh, *Solar Photovoltaics : Fundamentals, Technologies and Applications*. Prentice-Hall of India Pvt.Ltd, 2011.

[72] J. A. Duffie, *Solar engineering of thermal processes*, 4th ed. Hoboken: Wiley, 2013.

[73] R. G. Allen, L. S. Pereira, D. Raes, M. Smith *et al.*, “Crop evapotranspiration-guidelines for computing crop water requirements – FAO irrigation and drainage paper 56,” *FAO, Rome*, vol. 300, p. 6541, 1998.

[74] M. Gulin, M. Vašak, and M. Baotic, “Estimation of the global solar irradiance on tilted surfaces,” in *17th International Conference on Electrical Drives and Power Electronics (EDPE 2013)*, 2013, pp. 334–339.

[75] Z. Sun and A. Liu, “Fast scheme for estimation of instantaneous direct solar irradiance at the Earth’s surface,” *Solar Energy*, vol. 98, Part B, pp. 125–137, Dec. 2013.

[76] J. McKenzie, “Mean absolute percentage error and bias in economic forecasting,” *Economics Letters*, vol. 113, no. 3, pp. 259–262, Dec. 2011.

[77] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

[78] S. Makridakis, “Accuracy measures: theoretical and practical concerns,” *International Journal of Forecasting*, vol. 9, no. 4, pp. 527–529, Dec. 1993.

[79] E. Trillas and L. Eciolaza, *Fuzzy logic : an introductory course for engineering students*, ser. Studies in Fuzziness and Soft Computing: v. 320. Cham : Springer, 2015., 2015.

[80] L. Breiman, *Classification and regression trees*, ser. The Wadsworth statistics/probability series. Belmont, Calif. : Wadsworth International Group, c1984., 1984.

[81] W.-Y. Loh, “Fifty Years of Classification and Regression Trees,” *International Statistical Review*, vol. 82, no. 3, pp. 329–348, Dec. 2014.

[82] G. De’ath and K. E. Fabricius, “Classification and Regression Trees: A Powerful yet Simple Technique for Ecological Data Analysis,” *Ecology*, vol. 81, no. 11, pp. 3178–3192, Nov. 2000.

[83] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[84] M. Robnik-Sikonja, “Improving Random Forests,” in *Machine Learning: ECML 2004*, ser. Lecture Notes in Computer Science, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds. Springer Berlin Heidelberg, Sep. 2004, no. 3201, pp. 359–370.

[85] A. P. Engelbrecht, *Computational Intelligence*. Chichester, UK: John Wiley & Sons, Ltd, Oct. 2007.

[86] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2, 2004, pp. 985–990 vol.2.

[87] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *Ieee Transactions on Systems Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985, wOS:A1985AFX0900011.

[88] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*. Addison-Wesley, 2005.

[89] W. Pedrycz and F. Gomide, *Fuzzy systems engineering : toward human-centric computing*. Hoboken, N.J. : John Wiley : IEEE, 2007.

[90] M. Blej and M. Azizi, "Comparison of mamdani-type and sugeno-type fuzzy inference systems for fuzzy real time scheduling," *International Journal of Applied Engineering Research*, vol. 11, no. 22, pp. 11 071–11 075, 2016.

[91] A. Marzuki, S. Y. Tee, and S. Aminifar, "Study of fuzzy systems with Sugeno and Mamdanitype fuzzy inference systems for determination of heartbeat cases on Electrocardiogram (ECG) signals," *International Journal of Biomedical Engineering and Technology*, vol. 14, no. 3, pp. 243–276, Jan. 2014. [Online]. Available: http://www.inderscienceonline.com/doi/abs/10.1504/IJBET.2014.059673

[92] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press, 1998.

[93] H.-X. Zhang, F. Wang, and B. Zhang, "Genetic optimization of fuzzy membership functions," in *International Conference on Wavelet Analysis and Pattern Recognition, 2009. ICWAPR 2009*, Jul. 2009, pp. 465–470.

[94] A. Watts, P. Musilek, and L. Wyard-Scott, "Optimizing fuzzy control of energy harvesting remote monitoring systems," in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, Jun. 2013, pp. 661–666.

[95] L. Shuai and S. Wei, "Optimization of fuzzy control rules based on differential evolution algorithm," in *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*, Aug. 2014, pp. 2610–2613.

[96] S.-K. Oh, W.-D. Kim, and W. Pedrycz, "Design of optimized cascade fuzzy controller based on differential evolution: Simulation studies and practical insights," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 520 – 532, 2012.

[97] M. Sheikhan and S. A. Ghoreishi, “Antiviral therapy using a fuzzy controller optimized by modified evolutionary algorithms: a comparative study,” *Neural Computing and Applications*, vol. 23, no. 6, pp. 1801–1813, Sep. 2012.

[98] L. K. Letting, J. L. Munda, and A. Hamam, “Particle swarm optimized T-S fuzzy logic controller for maximum power point tracking in a photovoltaic system,” in *2010 Conference Proceedings IPEC*, Oct. 2010, pp. 89–94.

[99] R. Rahmani, M. Mahmodian, S. Mekhilef, and A. Shojaei, “Fuzzy logic controller optimized by particle swarm optimization for DC motor speed control,” in *2012 IEEE Student Conference on Research and Development (SCOReD)*, Dec. 2012, pp. 109–113.

[100] C. Caraveo, F. Valdez, and O. Castillo, “Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation,” *Applied Soft Computing*, vol. 43, pp. 131–142, Jun. 2016.

[101] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution. [electronic resource] : a practical approach to global optimization*, ser. Natural computing series. Berlin : Springer, 2005, 2005.

[102] S. Das and P. Suganthan, “Differential Evolution: A Survey of the State-of-the-Art,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[103] F. Penunuri, C. Cab, O. Carvente, M. A. Zambrano-Arjona, and J. A. Tapia, “A study of the Classical Differential Evolution control parameters,” *Swarm and Evolutionary Computation*, vol. 26, pp. 86–96, Feb. 2016.

[104] C. Worasucheep, “A new self adaptive differential evolution: Its application in forecasting the index of Stock Exchange of Thailand,” in *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, Sep. 2007, pp. 1918–1925.

[105] ——, “High-dimensional function optimization with a self adaptive differential evolution,” in *IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009. ICIS 2009*, vol. 1, Nov. 2009, pp. 668–673.

[106] R. Storn, “On the usage of differential evolution for function optimization,” in *Proceedings of North American Fuzzy Information Processing*, Jun. 1996, pp. 519–523.

[107] R. Mallipeddi and P. Suganthan, “Empirical study on the effect of population size on differential evolution algorithm,” in *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 3663–3670.

[108] S. Chen, J. Montgomery, and A. Bolufe-Rohler, “Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution,” *Applied Intelligence*, vol. 42, no. 3, pp. 514–526, Apr. 2015.

[109] G. Jeyakumar and C. Shanmugavelayutham, "An empirical comparison of differential evolution variants for high dimensional function optimization," in *International Conference on Intelligent Agent Multi-Agent Systems, 2009. IAMA 2009*, Jul. 2009, pp. 1–6.

[110] E. Mezura-Montes, J. Velazquez-Reyes, and C. A. Coello Coello, "A Comparative Study of Differential Evolution Variants for Global Optimization," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '06. New York, NY, USA: ACM, 2006, pp. 485–492.

[111] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016. [Online]. Available: https://www.R-project.org

[112] C. Bergmeir and J. M. Benítez, "Neural networks in R using the Stuttgart Neural Network Simulator: RSNNS," *Journal of Statistical Software*, vol. 46, no. 7, pp. 1–26, 2012.

[113] S. P. Fekete, A. Kröller, S. Fischer, and D. Pfisterer, "Shawn: The fast, highly customizable sensor network simulator," in *Proceedings of the Fourth International Conference on Networked Sensing Systems (INSS 2007)*, Jun. 2007.

[114] J. Rada-Vilela, "fuzzylite: a fuzzy logic control library," 2014. [Online]. Available: http://www.fuzzylite.com

[115] B. Gough, Ed., *GNU Scientific Library Reference Manual - Third Edition*, 3rd ed. Bristol: Network Theory Ltd., Jan. 2009.

[116] *BMP085 Digital pressure sensor data sheet*. Bosch Sensortec, Oct 2009, no. BST-BMP085-DS000-05, Rev 1.2.

[117] A. Brown, "Severe weather detector and alarm," U.S. Patent US5 978 738 A, Nov., 1999.

[118] ——, "Weather detector," U.S. Patent US6 076 044 A, Jun., 2000.

[119] ——, "Severe weather detector and alarm," U.S. Patent US6 597 990 B2, Jul., 2003.

[120] F. Vizard, "Storm warnings at home," *Popular Mechanics*, p. 72, Oct 1994.

[121] R. Cleveland, W. Cleveland, J. Mcrae, and I. Terpenning, "STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion)," *Journal of Official Statistics*, vol. 6, pp. 3–73, 1990.

[122] R. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. Otexts: Melbourne, Australia, 2013. [Online]. Available: http://otexts.org/fpp/

[123] J. Rodway, P. Musilek, E. Lozowski, M. Prauzek, and J. Heckenbergerova, "Pressure-based prediction of harvestable energy for powering environmental monitoring systems," in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, Jun. 2015, pp. 725–730.

[124] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[125] A. G. Watts, "Managing energy-harvesting environmental monitoring systems," Master's Thesis, University of Alberta, 2014.

[126] N. Roseveare and B. Natarajan, "A structured approach to optimization of energy harvesting wireless sensor networks," in *2013 IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2013, pp. 420–425.

[127] I. Ahmedy, M. A. Ngadi, and S. N. Omar, "Using store-forward technique to conserve energy in wireless sensor networks: Initial step for routing mechanism," in *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, vol. 2, Apr. 2012, pp. 671–676.

[128] P. Musilek, M. Prauzek, P. Krömer, J. Rodway, and T. Bartoň, *Intelligent Energy Management for Environmental Monitoring Systems*, ser. Intelligent Data Centric Systems. Elsevier, 2017, pp. 1–30.

[129] M. Prauzek, P. Musilek, P. Krömer, J. Rodway, M. Stankus, and J. Hlavica, "Energy Availability Forecasting for Harvesting-aware Wireless Sensor Networks: Analysis of Energy Demand of a Predictor Based on Evolutionary Fuzzy Rules," in *2015 International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, Sep. 2015, pp. 200–203.

[130] J. Rodway and P. Musilek, "Wireless sensor networks with pressure-based energy forecasting: A simulation study," in *2016 29th Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2016.

[131] J. Rodway, P. Krömer, S. Karimi, and P. Musilek, "Differential evolution optimized fuzzy controller for wireless sensor network energy management," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2016)*, Jul. 2016.

[132] J. Rodway and P. Musilek, "Harvesting-aware energy management for environmental monitoring WSN," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, Jun. 2016, pp. 1–6.

[133] R. Chai and Y. Zhang, "A Practical Supercapacitor Model for Power Management in Wireless Sensor Nodes," *IEEE Transactions on Power Electronics*, vol. 30, no. 12, pp. 6720–6730, Dec. 2015.

Appendix A

Reference Controller Results

Static Energy Forecasts

Table A.1: Reference controller results, static energy forecasts.

Energy Forecast	VL		L		M		H		VH	
Node	E_R (J)	M_T	E_R (J)	M_T	E_R (J)	M_T	E_R (J)	M_T	E_R (J)	M_T
1	959.15	480681	805.90	453502	2082.34	526224	2100.58	652341	2082.35	741840
2	918.35	481073	761.72	454632	2009.88	526230	2028.36	652171	1990.01	741076
3	2937.01	481198	2721.36	440726	6101.59	492682	6137.26	613908	6095.89	703961
4	0.00	482248	0.00	453117	0.04	580874	3.12	708718	10.84	799927
Total	4814.51	1925200	4288.98	1801977	10193.85	2126010	10269.32	2627138	10179.09	2986804

Appendix B

Candidate Controllers

Parameter Summary

Table B.1: Candidate Controller Parameter Summary.

Controller	Number of Inputs	Number of MF/input	MF Type
Reference	2	5	Triangular
CC1	2	5	Trapezoidal
CC2	2	3	Trapezoidal
CC3	4	3	Trapezoidal
CC4	1	3	Trapezoidal
CC5	1	5	Trapezoidal
CC6	1	9	Trapezoidal
CC7	2	3	Trapezoidal
CC8	4	3	Trapezoidal
CC11	6	3	Trapezoidal
CC9	2	5	Gaussian
CC10	2	3	Gaussian
CC15	2	5	Static Triangular
CC13	2	3	Trapezoidal
CC16	2	5	Trapezoidal

Appendix C

Simulation Results

Tabulated Values

Table C.1: Summary of Simulation Results

Simulation	M_T	Effective N_A	E_R (J)	E_L (J)
Reference/Perfect	2391647	0.4579	3646.96	4994900.16
Reference/EA	2986804	0.5764	10179.10	4844185.46
Reference/CART	2248726	0.4295	3989.38	5021685.09
CC4	2682004	0.5157	1617.71	4888969.57
CC5	2446397	0.4688	953.20	4934078.17
CC6	2556287	0.4907	381.75	4912301.15
CC7/Perfect	3154930	0.6098	340.73	4799879.78
CC7/EA	1092913	0.1994	6942.04	5360941.05
CC7/CART	3291989	0.6371	748.12	4773810.70
CC16/Perfect	2699539	0.5192	11.69	4887156.60
CC16/EA	475519	0.0764	11.69	5366953.68
CC16/CART	2835971	0.5463	105.30	4860924.74
CC1/Perfect	2894214	0.5579	82.29	4845747.24
CC1/EA	2614728	0.5023	63.80	4899759.99
CC1/CART	2701279	0.5195	316.44	4881591.46
CC1/PD	1214682	0.2236	11.69	5162948.46
CC2/Perfect	2759743	0.5312	401.69	4874789.08
CC2/EA	1446295	0.2697	1914.09	5119387.00
CC2/CART	2689458	0.5172	536.50	4887935.22
CC10/Perfect	2457726	0.4710	60.31	4930093.13
CC10/EA	2457726	0.4710	60.31	4930093.13
CC10/CART	2457726	0.4710	60.31	4930093.13
CC8/Perfect	3366899	0.6520	234.09	4769315.98
CC8/EA	92580	0.0003	11.69	5370608.40
CC8/CART	3346323	0.6479	317.50	4768668.31
CC3/Perfect	3436495	0.6657	647.25	4745695.03
CC3/EA	663485	0.1139	623.58	5270415.88
CC3/CART	3470444	0.6726	1210.53	4740740.16
CC15/Perfect	3122369	0.6033	592.13	4807935.91
CC13/Perfect	4349760	0.8475	4126.90	4589276.32
CC13/EA	876853	0.1564	11.69	5363140.93
CC13/CART	3327212	0.5477	5598.38	4784066.53