



Article

Harvesting-Aware Energy Management for Environmental Monitoring WSN[†]

James Rodway^{1,‡} and Petr Musilek^{1,2,*,‡}

¹ Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada; jrodway@ualberta.ca

² Faculty of Electrical Engineering and Computer Science, VŠB-TU Ostrava, 708 33 Ostrava, Czech Republic

* Correspondence: petr.musilek@ualberta.ca; Tel.: +1-780-492-5368

† This paper is an extended version of our paper published in Rodway, J.; Musilek, P. Harvesting-aware energy management for environmental monitoring WSN. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016.

‡ These authors contributed equally to this work.

Academic Editor: Rodolfo Araneo

Received: 14 February 2017; Accepted: 21 April 2017; Published: 1 May 2017

Abstract: Wireless sensor networks can be used to collect data in remote locations, especially when energy harvesting is used to extend the lifetime of individual nodes. However, in order to use the collected energy most effectively, its consumption must be managed. In this work, forecasts of diurnal solar energies were made based on measurements of atmospheric pressure. These forecasts were used as part of an adaptive duty cycling scheme for node level energy management. This management was realized with a fuzzy logic controller that has been tuned using differential evolution. Controllers were created using one and two days of energy forecasts, then simulated in software. These controllers outperformed a human-created reference controller by taking more measurements while using less reserve energy during the simulated period. The energy forecasts were comparable to other available methods, while the method of tuning the fuzzy controller improved overall node performance. The combination of the two is a promising method of energy management.

Keywords: wireless sensor networks; energy forecast; differential evolution; energy management

1. Introduction

Wireless sensor networks (WSN) can be used to collect data of interest with high spatial and temporal resolution. Such networks are made up of nodes consisting of the required sensors, wireless communications hardware, energy storage, and a microprocessor to facilitate data processing and other tasks [1]. When outfitted with energy harvesting devices, these sensor nodes can be deployed without the need for large power supplies or frequent maintenance visits. This is of particular interest for long-term deployment in remote locations where trips to the field may be prohibitively expensive, impossible, or otherwise troublesome. Additionally, especially in environmental monitoring applications, certain energy sources are undesirable (e.g., large lead acid batteries that are difficult to transport) and therefore cannot be utilized to increase deployment length [2]. Such applications include ecosystem monitoring in tropical dry forests, the arctic, and on glaciers [3–5].

In order to realize the longest deployment possible while still collecting useful data, a node's energy must be managed. There are a number of different methods of managing energy within WSNs. They can be divided into three different levels [6]. At the component level, the choice of individual parts with low power operating modes and other similar optimizations are important for reducing the overall energy usage of the nodes. At the node level, adaptive duty cycling, as well as data transmission reducing schemes can be used to manage energy. Network level strategies include dynamic selection

of network sinks and energy aware message routing to distribute energy usage intelligently [7–10]. The focus of this work is on the use of node level adaptive duty-cycling.

Without energy harvesting, duty-cycle based management is simple, as estimates of node energy usage and storage capacity can be used to directly determine its duty cycle. Management becomes more complex with the addition of harvesting, where differences in harvesting opportunities and limited storage necessitate striking a balance between energy conservation and consumption [11]. On one side, liberal energy consumption runs the risk of missed measurements or outright node failure. On the other side, highly conservative energy management can lead to its inefficient use, e.g., when incoming energy can be neither used directly nor stored because the energy storage device is already full. With proper energy management, harvesting allows for smaller energy storage devices, as well as for longer deployment times, potentially approaching perpetual operation [12].

Energy management strategies vary in complexity, based on the information available for decision making. Foreknowledge of harvesting opportunities may be included, e.g., extracted from the experienced harvesting opportunities. Adaptive duty cycling for energy-harvesting sensor nodes is discussed in [13] with energy prediction based on an exponentially weighted average. The implemented controller allowed 58% more environmental energy utilization when compared to the case where power management is used without harvest awareness. Adaptation of sensor node parameters based on a prediction of future energy is shown in [14]. There, the parameters are set based on the incoming energy, 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 [15]. This scheme affects energy intensive node operations, such as the duration of video transmitted back to the base station.

Fuzzy logic control is often used to implement adaptive duty cycling for individual nodes [4,6,16]. Inputs to such fuzzy controllers generally include the state-of-charge of available energy storage devices, but may also include the amount of data stored on a node. Of particular interest to increasing usage of harvested energy is an input associated with energy forecast. Early estimates of incoming energy may allow for increased energy usage in cases when the incoming energy would support it, or to reduce energy consumption when a dearth of incoming energy is expected.

In this contribution, forecasts of daily solar energies are developed using measurements of atmospheric pressure. These forecasts are then combined with fuzzy logic controllers tuned using differential evolution. The ensuing system provides node-level energy management through adaptive duty cycling. This system was then applied to a simulation of a simple WSN in order to examine its behavior. While not fully representative of actual deployment conditions, simulations allow initial development of energy management strategies. The use of sophisticated simulation tools allows WSN development to occur faster and with lower costs compared to experimental trials with hardware prototypes. At the same time, they also provide repeatability that facilitates comparison between different strategies [2]. This is especially important when applying optimization methods that require consistent approach to solution evaluation.

This paper is comprised of six sections. The materials and methods used are described in Section 2. Section 3 shows the development of a solar energy forecast based on measurements of atmospheric pressure. Design and optimization of the fuzzy logic controller is discussed in Section 4. Results of simulations for a small wireless sensor network are presented in Section 5. Major conclusions and potential directions of future work are outlined in Section 6.

2. Materials and Methods

2.1. Data

The meteorological data used in this study was obtained from a network of automated meteorological stations operated by the Washington State University (WSU) [17]. In this network, individual stations have a wide array of instruments with some measuring variables of interest for

this study: temperature, atmospheric pressure, and solar irradiance. Data is measured once every five seconds, summarized by the data logger and reported every 15 min. Four stations with the necessary measurements were selected for use in this study. Locations of the selected stations are tabulated in Table 1. The distances between these stations are larger than may be expected for many WSN applications. However, this large spatial separation provides exaggeration of the differences in harvesting opportunities between the sensor nodes in the network, allowing for more thorough testing of the energy management controller during simulations.

Table 1. 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

2.2. Software

The developed solar energy forecasters are based on R [18]. The multilayer perceptron neural network is from the R RSNNS library [19], the regression tree from the R tree library, and the random forests the R randomForest library [20].

Network simulations have been carried out using the shawn network simulator [21]. This simulator is fast and extensible, allowing the energy considerations (such as charging, discharging and other node operation energy costs) to be added to modeled devices. Simulations can easily be further extended through the addition of additional phenomena as well as by increasing the sophistication of various component models. Outputs of the simulated nodes were taken from the logged transmissions, then post-processed to extract information regarding energy reserve and buffer charge levels, as well as the measurements themselves. As opposed to trials with hardware prototypes, simulations allow for estimated values to be obtained without actually collecting them. In other words, no extra values have to be measured, stored and transmitted - operations that would incur additional computational and energy costs, possibly affecting the outcome of the trials. Fuzzy logic was implemented using the fuzzylite library [22]. To support sampling the time series at arbitrary points, they were fit using interpolating splines from the GNU's Not Unix (GNU) Scientific Library [23].

2.3. Methods

2.3.1. Regression Methods

A number of different methods are available for estimating a target value based on several explanatory variables. The focus here is on methods that can be run with the limited resources of a wireless sensor node. Four regression methods have been tested: classification and regression trees (CART), random forests (RF), multilayer perceptron (MLP) neural networks (NN) trained using backpropagation, and MLP NN trained using the extreme learning machine (ELM) method.

CARTs were originally developed by Brieman et al. in 1984 [24,25]. This method can deal with non-linear relationships and high order interactions, while still retained easy interpretability [26]. 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, and 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. After their creation, regression trees are a simple set of branching comparisons with fixed values. With no need for normalized data, simple implementation and low computational requirements, CARTs are an attractive option for use on limited hardware.

A RF consists of a number of regression trees, where random subsets of variables are considered at each split [20,27]. The results of all 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, variables are never compared to one another, and therefore normalization is not required.

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. Activation of each neuron is obtained using a squashing function depending on the result of comparison between its inputs and weights through inner product. The activations are propagated through subsequent layers of neurons, until reaching the output layer. In these types of networks, the weights of all neurons are determined using an iterative method based on gradient descent [28].

The ELM network has the same structure as the MLP network, but uses a different, non-iterative training method [29]. Weights of the hidden nodes are assigned randomly, and the output weights are solved for using a matrix pseudo-inverse. The primary benefit of this method of training is a significantly shorter training time.

2.3.2. Fuzzy Logic

The energy usage of the nodes is managed by a fuzzy logic controller. This type of controller can be implemented without high computational requirements while still enabling complex control. This way, an effective controller can be created without the consideration of hardware technical details (e.g., discharge characteristics and energy costs of node operations [16]).

In this contribution, a 0th order Takagi-Sugeno (TS) fuzzy logic controller is used to manage the number of operations a wireless sensor node performs in a day. In this type of TS fuzzy controller, the output is an average of the output singletons weighted by the activations of their associated rules. Levels of activation of individual fuzzy sets are calculated using a possibility measure [30]:

$$\text{Poss}(\{x\}, A_i) = \sup_{x \in X} (\min(\{x\}, A_i(x))) = A_i(x),$$

where $\{x\}$ is a fuzzy singleton corresponding to the value of the input and A_i is one of the fuzzy membership functions of input variable A defined on the universe of discourse X . Activity of a rule r from the fuzzy rule base of the form IF $X = A_i$ AND $Y = B_i$ THEN $Z = C_i$ is calculated using an algebraic-product t -norm operation:

$$\lambda_r = \text{Poss}(\{x\}, A_i) \, t \, \text{Poss}(\{y\}, B_i) = A_i(x) \cdot B_i(x),$$

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

$$C = \frac{\sum \lambda_r \{C_i\}}{\sum \lambda_r},$$

where $\{C_i\}$ is a fuzzy singleton from domain Z associated with rule r from the rule base, and C is the controller output.

2.3.3. Differential Evolution

To optimize the controllers, differential evolution (DE) has been used as a fast and simple method of global optimization [28,31]. It is attractive because of its performance, low number of control parameters, and low space complexity [32]. In order to reduce the number of function evaluations, an adaptive differential evolution method was used [33,34]. This method, termed WDE, allows any number of DE variants to be used, as well as differential weights and crossover probabilities.

Additionally, the size of the population is allowed to contract and expand, depending on how values of the fitness function change during the optimization. The optimization process starts with an initial population of size N_p . For each population member, a differential weighting (F) and crossover probability (CR) are randomly assigned, as well as a differential operator. DE then proceeds normally until a predetermined generation interval, γ_1 , when the number of successes and failures of each differential operator and (F , CR) pairs are examined. The goal of this examination is to change the probability with which each individual differential operator is assigned (i.e., more successful operators have a higher probability of being assigned to a population member) and which (F , CR) pairs should be reassigned (i.e., pairs with lower success rates than the average are reassigned randomly). At a longer interval, γ_2 , the size of the population is shrunk by $\alpha\%$, to a minimum population size of $N_p/2$. Finally, at the longest interval, γ_3 , if the fitness function values have not improved by a set threshold, the top $\alpha\%$ performers are retained and the rest of the population is reinitialized such that the total population size returns to its initial value, N_p .

The DE variants used to create new population members in this work include *DE/rand/1/bin*, *DE/current-to-best/2/bin*, and *DE/best/1/bin* [33]. An empirical study of the application of DE to high dimensional problems suggests that the best results for multimodal, non-separable problems may be obtained using the DE variants *DE/best/2/bin* and *DE/rand/2/bin* for 100 dimensions [35]. These operators were also included.

3. Energy Forecast

An energy forecast is an important component of an energy management scheme that allows to increase the amount of harvested energy used. With foreknowledge of harvesting opportunities, energy buffers may be depleted in advance of high amounts of incoming energy, or preserved if low amounts of energy are expected. Forecasts of daily solar energy have also become more important with the increased interest in renewable energy and smart agriculture. Tools for their modeling often require daily solar energy as an input. However, it may not have been measured due to the high cost of the required instrumentation [36–38]. More common meteorological observations may be available and can be used to estimate the amounts of solar energy. These alternative methods usually make use of the difference between observed maximum and minimum daily temperatures, with some models including other variables (e.g., minimum relative humidity) to improve the estimates [39–43]. Difficulty arises when applying regressions performed for one location to another, as differences in prevailing weather conditions may be quite significant.

With the limited hardware available in a sensor node, simple forecasts are preferable to more complex ones requiring intense computation. Additionally, meteorological variables useful for prediction of solar energy may not be required for the sensor nodes actual purpose, meaning that the inclusion of these values would require additional measurement instrumentation as well as energy expenditure and storage space that may not contribute to the main goal of the sensor network deployment.

In this work, we selected atmospheric as the predictor for solar energy forecasts, because of its relationship with changing weather conditions and cloud cover [44]. Measurements of pressure have already been used for simple weather forecasts in consumer products for a number of years [45,46]. Proper measurement of this variable may also be simpler than others, e.g., siting requirements of temperature measurements require the instrument to be shaded and provided with adequate airflow [47]. Additionally, appropriate sensors are available with low energy requirements (e.g., 2.7 mJ [48]) when compared to some other sensors and other node operations [2].

Creating the forecast begins by making an analytical estimate of the amount of incoming solar energy for a given location. The actual forecast target is an estimate of the transmissivity coefficient

associated with cloud cover, τ_F . This forecast value is then multiplied by the analytical estimate of incoming daily solar energy, E_A , which can be calculated as [49,50]:

$$E_A = 3600\tau G \int_{\omega_{\text{sunrise}}}^{\omega_{\text{sunset}}} \sin \theta d\theta, \quad (1)$$

where ω_{sunrise} and ω_{sunset} are, respectively, the sunrise and sunset angles of the location, while G is an estimate of the incoming solar radiation given by:

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

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

A simple model of clear sky atmospheric transmissivity, τ , is added to account for some of the absorption and scattering in the atmosphere:

$$\tau = 0.75 + 0.00002h,$$

where h is the site elevation in meters [51]. This model was developed through a linearization of Beer's Law (see [52]) 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 vapor 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 may be used instead for increased accuracy, but it would require calibration that may not be possible in practice.

Dividing the measured daily solar energy by this analytical value, one can obtain an estimate of the transmissivity to be used as the forecast target, τ_F . Samples of the measured incoming solar energy, the analytical estimate and the estimated transmissivity are shown in Figure 1.

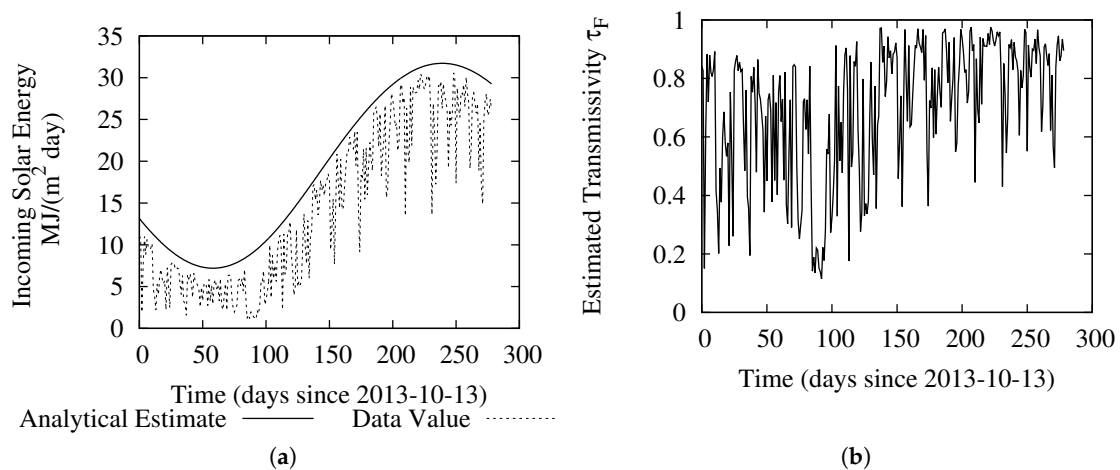


Figure 1. Solar values and transmissivity estimates. (a) Measured and analytical daily solar energies; (b) Transmissivity estimate.

Using the calculated analytical estimate as the solar energy forecast for the different stations results in the mean absolute percentage errors (MAPE) and root mean squared errors (RMSE) shown in Table 2.

Table 2. MAPE values when E_A is used as a predictor.

Dataset	MAPE (%)	RMSE (MJ/(m ² day))
Moxee	86.80	5.76
Lind	85.61	5.45
Prosser	84.92	5.62
Garfield East	113.16	6.90

For comparison purposes, an alternative possible model for estimating the incoming daily solar energy has been evaluated:

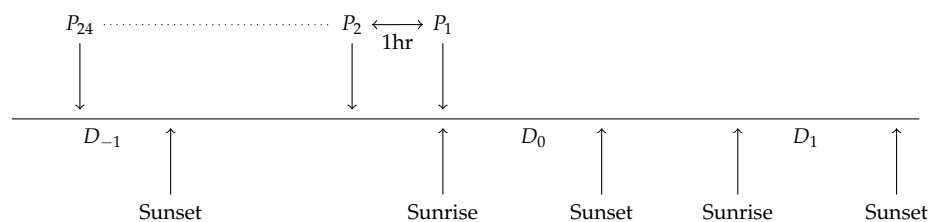
$$E_{SH} = K_r \sqrt{T_{max} - T_{min}} E_A,$$

where E_{SH} is the energy estimated using the Samandi-Hargreaves relationship, K_r is an empirical constant (a value of 0.17 is used here), T_{max} and T_{min} are the measured maximum and minimum temperatures of the day, and E_A is the analytical energy estimate [41]. Applying this relationship to the available data sets for the prediction of the next-day solar energy resulted in the MAPE and RMSE values tabulated in Table 3. These values are an improvement based on just using E_A as the predictor, both in terms of MAPE and RMSE.

Table 3. Error values for D_0 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

In our previous work, sets of pressure pairs and their differences were included as predictors, and only the energy for the upcoming day was predicted [53]. Here, at sunrise of a given day D_0 , the incoming solar energy for that day is predicted, as well as the incoming energy for the next five days (D_1, D_2, \dots, D_5). This approach is based on the most recent 24 hourly measurements (P_1, P_2, \dots, P_{24}), as shown in Figure 2. The differences between the individual pressure measurements were calculated and supplied as inputs as well, since the change in pressure over time is considered a major indicator of weather changes.

**Figure 2.** Measurement and prediction timeline using 24 hourly pressure measurements.

In the previous exploration, there was no input corresponding either to the time of the year or to the expected amount of incoming 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. Attempting to lower these errors and improve the overall error metrics, the analytic estimate of the total incoming energy (Equation 1) was also provided as an input. Forecasts for the upcoming daylight hours (i.e., D_0) and beyond (D_1 and onwards) can also be made using the same measurements with, with the expectation that the longer forecasting horizon may improve performance of the controller to be developed later.

However, because of the limited capacity of a sensor node's energy buffer, there is expected to be a limit to the size of a useful forecasting horizon.

To design desired predictor, CART, RF, ELM, and MLP methods were used. The hidden layer sizes of the ELM and MLP neural networks were expanded to 50 nodes, and the maximum number of training iterations allowed for the network trained with backpropagation increased to 1000. The inputs for the neural network methods were scaled to $[0, 1]$, while the CART and RF inputs retained their original values. 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.

Trials were run for both cases where forecast models were created for each individual station, as well as for the case where all data was combined and used to create one model for all stations. Single tailed t -tests were used to compare the mean absolute errors of these forecasts and not found to provide a significant improvement. This could simplify deployment of WSN if this method were used for energy management.

Calculated MAPE values for D_0 training set predictions for the various methods are shown in Table 4, while RMSE values are shown in Table 5. Selecting the trial runs with the lowest MAPE from the training set and applying them to the reserved test set yields the values shown in Table 6. These error measures show that the NN based methods see larger increases in the error values than the tree based method when applied to the test set. Compared to the error values for these sites using the temperature measurements, the MAPE values were slightly higher, but the RMSE values were lower, meaning that there were fewer instances of large errors with these methods. This is an important finding, as larger errors can cause greater problems when used for energy management.

A time series of the predictions made for one of the stations is shown in Figure 3. This figure shows that the CART and RF regressions have less deviation from the analytical estimate, while the NN methods had larger deviations. The ELM prediction had a number of impossible values, i.e., negative solar energies, that could be filtered out, but this is unlikely to make this method competitive.

Table 4. WSU MAPE training values for current day (D_0) prediction.

Method	Min	Average (σ)	Max
CART	32.76	34.29 (0.9874)	36.23
RF	32.62	33.81 (0.9608)	35.72
ELM	33.59	34.13 (0.4326)	35.04
MLP	24.63	26.42 (0.9008)	27.92

Table 5. WSU RMSE training values for current day (D_0) prediction ($\text{MJ}/(\text{m}^2 \text{ day})$).

Method	Min	Average (σ)	Max
CART	3.01	3.13 (0.060)	3.19
RF	2.82	2.93 (0.039)	2.96
ELM	3.32	3.38 (0.034)	3.43
MLP	2.48	2.66 (0.098)	2.77

Table 6. WSU test values for current day (D_0) prediction using minimum MAPE training model.

Method	MAPE (%)	RMSE
CART	38.34	3.41
RF	38.37	3.25
ELM	45.54	4.63
MLP	37.39	3.25

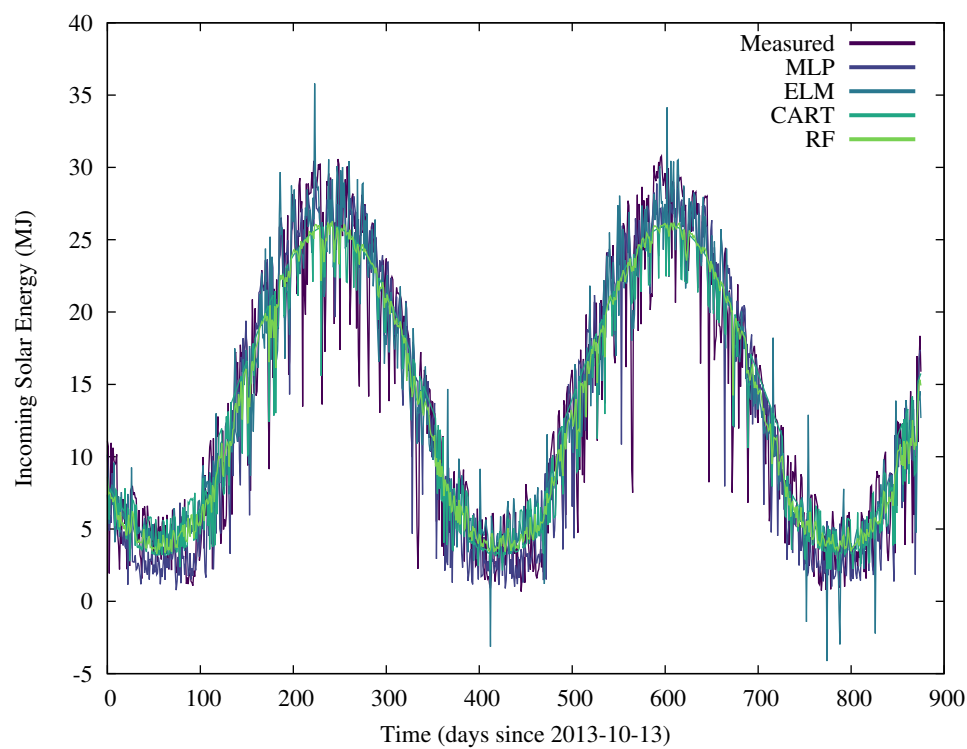


Figure 3. Single station solar energy values for different forecasts.

Figure 4 shows the test set results for models with the lowest MAPE values for larger forecasting horizons. With the increasing prediction time, the error values generally increased, as expected. Compared to the other methods, ELM had much higher error values for all forecasting horizons.

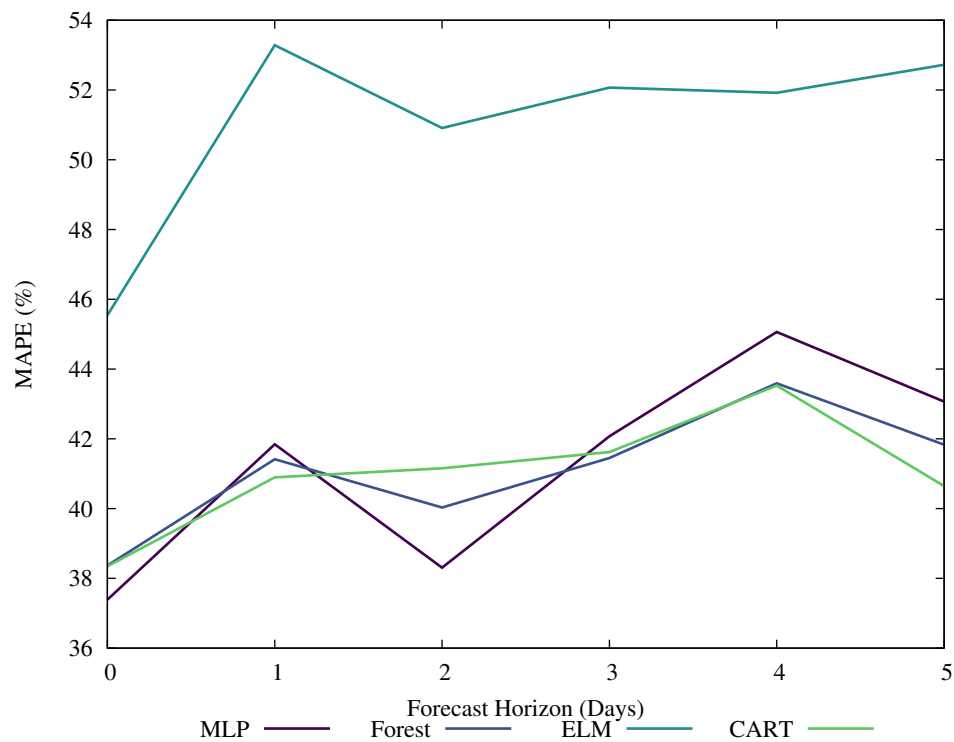


Figure 4. Test set values for lowest MAPE training set performance for different forecasting horizons.

4. Controller Optimization

One strength of fuzzy logic is the ability to use linguistically meaningful terms to describe the system's desired behavior, and to form the controller based on this description. However, the overall performance of the controller can be improved through optimization of the fuzzy membership functions. These functions are encoded as vectors of real valued numbers representing their individual parameters and operated on by an optimization method. In order to evaluate candidate solutions, the vectors are decoded into a fuzzy controller that is executed, and fitness values are calculated based on its performance.

After the randomization of the optimization algorithm, additional considerations should be made to ensure that a sound fuzzy controller is created. Preference for a semantically sound fuzzy set during optimization ensures consistency of the resulting controller and increases its transparency. Two important aspects of semantic soundness are coverage and distinguishability [30]. Metrics related to these two aspects can be calculated and added to the fitness function.

Our previous work on optimization of similar controllers included 5 membership functions for each input [54]. The resulting partitions for that optimization contained membership function with a large amount of overlap, leading to the selection of a lower number in the current work, i.e., 3 membership functions per input.

The membership functions of a fuzzy logic controller that determine the duty cycle of the sensor nodes were tuned using DE (c.f. Section 2). The inputs for the candidate controllers were the state of the energy buffer, and either one or two days of future energy. The output of the controller was a node activity value, N_A , that was used to determine the number of operations a node attempted to complete in a day.

The time between measurements is allowed to vary from 60 to 3600 s (1 min–1 h), while the time between transmissions from 120 to 86,400 s (2 min–1 day). In these simulations, N_A is used to modulate both operations. The relationship between the number of operations occurring in a 24 h period, O , and N_A is expressed using:

$$O = N_A(O_{\text{Max}} - O_{\text{Min}}) + O_{\text{Min}}, \quad (2)$$

where O may be either measurement or transmission. Selection of O_{min} and O_{max} is used as a simple way of ensuring that measurements of certain variables are made with a sufficient frequency. Two updates of node activities are performed, one at the sunrise of the current day and one at the sunset. For cases where energy forecasts are used, new forecasts are not made as part of the sunset update, but those made during the sunrise update are used if appropriate.

In order to compare the performances of candidate solutions, a fitness function, f , is defined as:

$$f = aE_L + bE_R + c(S_p + S_c),$$

where E_L represents unused environmental energy, E_R is the energy used from the primary battery reserves, and the final terms, S_p and S_c , represent penalties related to increasing the input variables distinguishability (i.e., that the membership functions do not overlap so much as to not be meaningfully unique) and coverage (i.e., that the entire input space is covered), respectively. Scaling constants a , b and c define the relative importance of the individual fitness components. In this work, E_L is defined as the energy present in the environment that is neither immediately used, nor stored for future use. That is, energy that is not harvested because the energy buffer is already full and the node operates at its highest activity level.

The relative importance of E_L and E_R in this fitness function depends 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, a has been set to 1×10^{-5} since the typical amount of uncaptured and unused energy from previous simulations is on the order of 1 MJ.

The reserve battery usage has a relatively high weight, $b = 1$. To encourage formation of meaningful fuzzy sets, c is set to a value of 5.

Energy values used for energy costs of various operations, including energy usage estimates for transmission, reception, and measurements, were taken from [55]. Estimations of the energy used for prediction of daily solar energy [56] were presented in [57]. Different combinations of microcontroller, operating frequency, compilers and optimization levels were tested, resulting in power consumption ranging between 5.4 μJ and 5817.6 μJ with the bulk of the tested configurations requiring less than 100 μJ . As many of these energy estimates are below the estimated energy usage during sleep (56.67 μJ), the energy cost associated with the creation of an the energy forecast can be safely neglected.

The optimization was performed using a single location (Moxee) and the first 670 days of the 880 day data set. A single node was used in order to keep processing times low. This did remove a degree of interaction between nodes, as well as reduced the overall energy usage of the node as it did not receive any transmissions during this simulation. To compensate for the lower energy consumption, the cost of transmission was increased compared to the value used for the simulation of the entire network.

A controller was optimized for the case where no forecast was provided, and only the energy buffer's state of charge. Three membership functions were used in this fuzzy set. The resulting input and output fuzzy sets for this controller are shown in Figure 5.

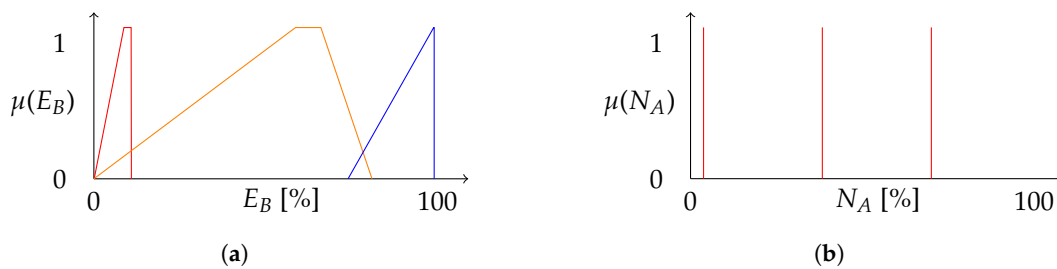


Figure 5. Fuzzy partitions for controller with no forecast. (a) Fuzzy input partition for energy buffer state; (b) Fuzzy output partition for node activity.

The optimization for the controller utilizing the current-day energy forecast resulted in the fuzzy sets shown in Figure 6. In each of these inputs, one of the trapezoidal membership functions was compressed to a fuzzy singleton at 100%, overlapping with another function.

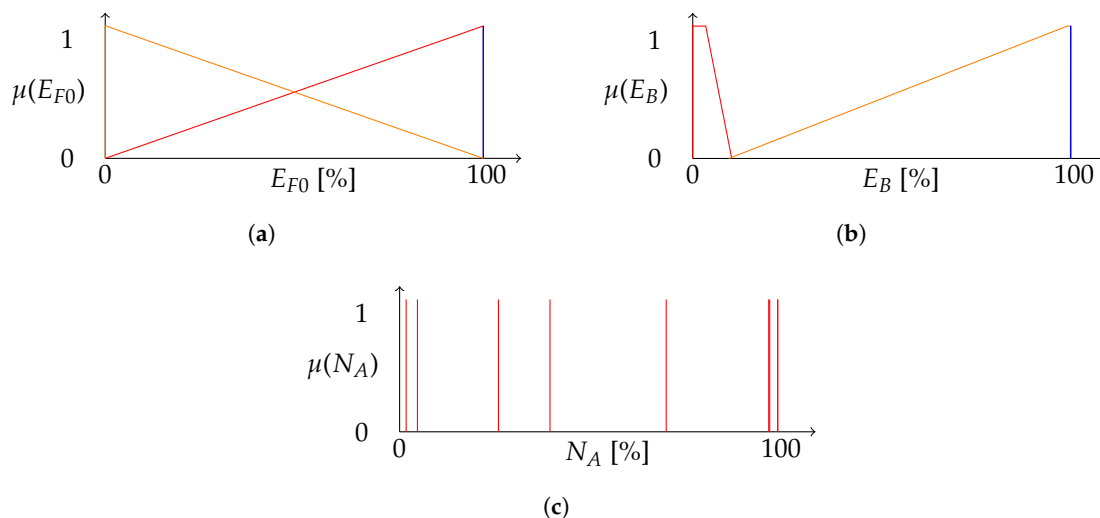


Figure 6. Fuzzy partitions for the one-day controller. (a) Fuzzy input partition for forecast; (b) Fuzzy input partition for energy buffer state; (c) Fuzzy output partition for node activity.

Results of the optimization where the controller was provided with current and next-day energy forecast are shown in Figure 7. In both cases, the close proximity of singletons in the output partition may allow for reduction in the number of rules. The controller using two day forecasts optimized to a lower fitness value compared to the controller using only one day forecast (11.65 vs. 12.29).

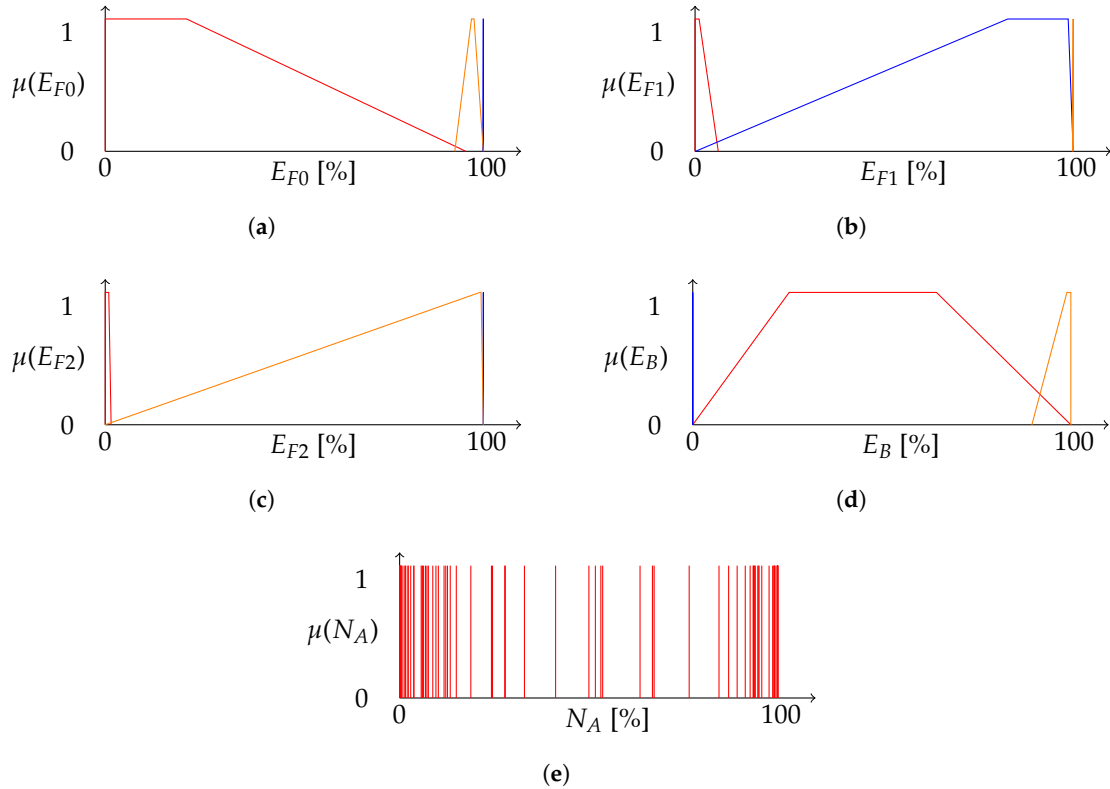


Figure 7. Fuzzy partitions for the two-day controller. (a) Fuzzy input partition for first forecast interval; (b) Fuzzy input partition for second forecast interval; (c) Fuzzy input partition for third forecast interval; (d) Fuzzy input partition for energy buffer; (e) Fuzzy partition for node activity output.

5. Simulation Results

The topology of the simulated network is shown in Figure 8. Nodes 1, 2, 3 and 4 correspond to data from the Moxee, Prosser, Lind and Garfield East stations, respectively. In this figure, the solid lines denote communication channels. This layout roughly mimics the relative positions of the actual stations. For the full simulation, all 880 days of the data set were used. During simulations, node activity levels were updated at sunrises and sunsets. An initial node activity was used prior to the first update which resulted in the bulk of the energy usage for many of the best performing simulations.

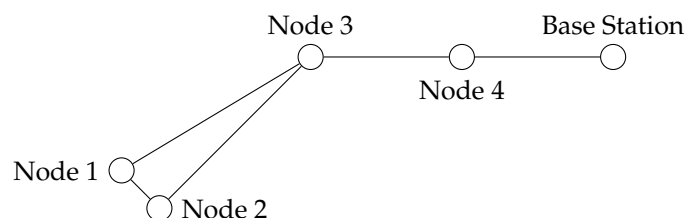


Figure 8. Layout of WSU nodes used in simulations.

Realistic parameters associated with a node's energy harvesting and usage have been used in this simulation [55]. Energy storage for each node is comprised of a non-rechargeable battery and

a supercapacitor used as an energy buffer. Nodes use a simple store-and-forward scheme to pass messages through the network and to the base station [58]. The flexible nature of the simulation also allows for simple changes to be made to quickly examine a variety of different configurations, including the use of different sensors.

A human-created reference controller was used for comparison with the generated controllers [6,54,55]. This controller consisted of two inputs, one for the energy forecast of the upcoming day and one for the state of the energy buffer. The fuzzy partitions for these inputs had triangular membership functions, with the energy buffer functions evenly dividing the input space and the energy forecast membership functions dividing the space based on 20% quantiles. The outputs were five fuzzy singletons, equally spaced between 0 and 100%.

For additional comparison, simulations were performed where a constant node activity was used. A plot of energy usages for the increasing node activities is shown in Figure 9. There was a constant amount of energy usage for simulations with node activities up to approximately 0.276, after which the amount of energy usage increased rapidly. For values of node activity above approximately 0.6, nodes began to fail due to depletion of the primary energy source. The total number of measurements taken by the network, as well as the amount of unharvested energy were both roughly linear with the increasing node activities. The linear relationship between total network measurements and the node activity allows for a regression line to be fit. The line can be described as $M_T = 5024013N_A + 91183$, where M_T is the total number of measurements taken during the simulation and N_A is the node activity.

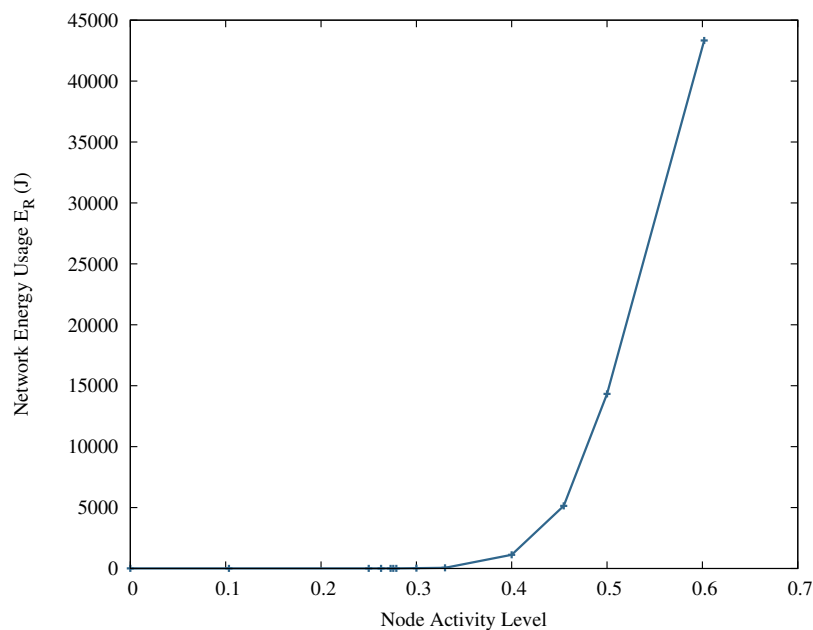


Figure 9. Total network energy usage for constant node activities.

Energy values for the reference controller using a perfect energy forecast are shown in Figure 10. The total amount of primary battery energy usage for this network during the simulation was 3646.95 J. In this simulation, the node using the largest amount of energy was Node 3, due to its central location. The node using the least amount of primary energy was Node 4, which only accepted transmissions from Node 3. The plots of supercapacitor energies in Figure 10 support this observation, as the energy buffer for Node 4 did not drop to zero for the bulk of the simulation, while the other nodes experienced supercapacitor energy values of zero for winter months. This simulation resulted in a total of 2,391,647 measurements, with a daily minimum of 381, a daily maximum of 1162 and a daily average number of measurements of 679. Using the total number of measurements to calculate an equivalent constant node activity resulted in a value of 0.458.

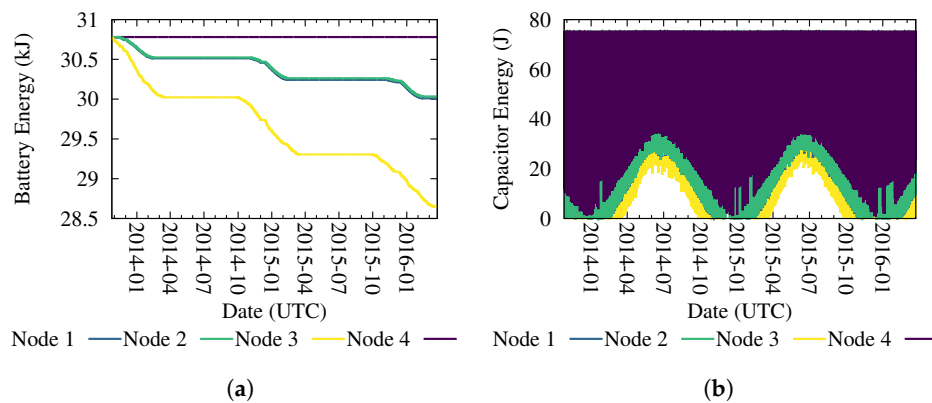


Figure 10. Energy values for simulated reference controller using a perfect energy forecast. (a) Battery energy values; (b) Supercapacitor energy values.

Simulation results for the evolved controller using no forecast are tabulated in Table 7. Using the total number of measurements taken, the effective activity was calculated at 0.516.

Table 7. No forecast evolved controller results.

Node	E_R (J)	M_T	Min M_D	Max M_D	Mean M_D
1	245.46	678,358	442	1032	771
2	141.47	681,126	446	1030	774
3	908.90	622,515	416	1032	707
4	321.88	700,005	244	1032	795
Tot/Comb	1617.25	2,682,004	244	1032	762

The optimized controller that used only one day of energy forecast with a perfect forecast resulted in the values shown in Table 8. The effective activity for this controller was calculated as 0.531.

Table 8. 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	704,370	424	1139	800
2	19.53	707,452	396	1137	804
3	310.11	631,803	396	1145	718
4	31.98	716,118	415	1137	814
Tot/Comb	401.68	2,759,743	396	1145	784

When the number of forecasts was increased to two, the simulation resulted in the values shown in Table 9. The effective activity for this controller using a perfect forecast was calculated as 0.666.

Table 9. 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	862,099	504	1152	980
2	7.71	860,482	358	1150	978
3	585.51	855,833	511	1154	973
4	19.44	858,081	194	1154	975
Tot/Comb	647.26	3,436,495	194	1154	976

Comparison of the simulations using the fixed values of N_A to the reference controller shows that the human-designed controller had slightly better performance, collecting a larger number of measurements while using less energy than the equivalent constant controller. The tuned controllers improved on this performance further: they had higher equivalent constant node activities, but also used less of the total reserve energy. The no-forecast controller had a slightly lower effective activity when compared to the D_0 forecast case, but used roughly 4 times more total energy over the simulated period. The D_0/D_1 controller uses slightly more energy while operating at a higher general activity level when compared to the D_0 controller. However, this controller demonstrates a larger variance in the number of measurements taken during a day, suggesting its greater ability to adapt to the amount of harvestable energy. The increased energy usage during this simulation is completely due to higher usage at Node 3, meaning that further improvements could be gained by allowing different nodes to employ different controllers, or through the use of an energy-aware message routing method.

When the pressure-based CART forecast is used, the summaries of the daily measurements and amounts of reserve energy usages are shown in Tables 10 and 11 for the D_0 and D_0/D_1 controllers, respectively. The calculated equivalent constant node activities for these simulations were 0.520 and 0.673, respectively. Comparing these equivalent values to the values from the perfect forecast case showed a slight decrease in the case of the D_0 controller, and an increase in the case of the D_0/D_1 controller. Comparing these values to the no-forecast version, the evolved D_0 controller had similar effective activity while still using less energy. The D_0/D_1 controller still had a higher effective activity level, but at the cost of roughly 135% of the total energy usage compared to the no-forecast simulation. The increases in energy usage for the controllers utilizing a forecast arose from the errors in the generated forecast that was used as an input. The D_0/D_1 controller suffered a larger increase due to the higher error associated with the forecast for the second day.

Table 10. D_0 evolved controller results, CART forecast.

Node	E_R (J)	M_T	Min M_D	Max M_D	Mean M_D
1	36.68	687,776	464	958	782
2	40.95	687,799	464	957	782
3	403.01	617,331	420	941	702
4	56.08	711,291	168	993	808
Tot/Comb	536.61	2,704,197	168	993	768

Table 11. D_0/D_1 evolved controller results, CART forecast.

Node	E_R (J)	M_T	Min M_D	Max M_D	Mean M_D
1	329.59	868,936	664	1152	987
2	243.90	868,935	661	1150	987
3	1395.43	863,339	621	1154	981
4	222.83	859,234	645	1154	988
Tot/Comb	2191.75	3,470,444	621	1154	985

In this simulation, network energy usage climbs very quickly as the network activity level climbs past a certain point, as previously shown in Figure 9. The D_0/D_1 controller operated at a higher node activity level, which resulted in a higher energy use. When forecast errors were introduced, they took a greater toll on the energy usage for this controller because of where its usual operating point was located.

In all these cases, the tuned controllers performed better than the reference controller, with all combinations of controllers and forecasts taking more measurements while using less reserve energy during the simulation. Using perfect forecasts, the D_0/D_1 controller had a greater equivalent node activity with a modest increase in reserve energy usage when compared to the D_0 controller, indicating

a better utilization of harvestable energy. Both controllers utilizing the energy forecasts had higher activity levels and lower energy reserve usages compared to the no-forecast controller when using a perfect forecast. Comparing the differences between the perfect and pressure-based forecasts for the D_0 controller, there was an increase in reserve energy usage and a slight decrease in equivalent node activity, underscoring the importance of accurate forecasting values. For the case where two days of energy forecast were used, the controller showed a greater jump in usage of the energy reserve when comparing the perfect and pressure-based forecasts. Compared to the no-forecast controller, in this case, the D_0/D_1 controller used more reserve energy while operating at a greater activity level.

6. Conclusions and Future Work

This contribution describes a new energy management scheme for wireless sensor nodes with solar energy harvesting capability. While energy harvesting allows nodes to replenish reserves in the field, consumption of the collected energy should be managed for its most effective usage. Estimates of energy available in the future can be used to inform rates of consumption. To that end, in this work, predictions of daily solar energies are based on hourly measurements of atmospheric pressure. These predictions are used in combination with a fuzzy logic controller to improve the usage of energy harvested by a sensor node. The membership functions of the fuzzy logic controller have been tuned using differential evolution to further improve energy utilization.

With respect to the energy forecast, the resulting forecasts using machine learning techniques provide comparable results to the estimates based on the minimum and maximum measured daily temperatures when MAPE is considered, but some methods have lower values of RMSE.

In comparison to forecasts based on minimum and maximum measured daily temperatures, the proposed machine learning forecasting techniques provide better accuracy in terms of RMSE and comparable results when MAPE is considered. The methods explored here have the additional benefits of the ease of implementation on the limited hardware that is available in sensor nodes, and a predictor variable that is easier to measure.

The controllers tuned with differential evolution outperformed the human-created reference controller, even when the perfect energy forecast was replaced with a simple solar energy forecast based on measurements of atmospheric pressure. The D_0 controller performed better than the no-forecast controller for both the perfect and generated forecasts, while the D_0/D_1 controller saw an increase in both network activity and energy usage with respect to the no-forecast case while using the generated forecast. However, the D_0/D_1 controllers higher network activity and larger variance in number of daily measurements with relatively modest increases in energy usage points to its more effective use of harvested energy. The relative weighting of the reserve and harvested energy values used to evaluate the potential controllers can be changed to reflect the requirements of a given application.

There are a number of possible avenues for future work. With respect to possible improvement of the simulation, a number of additional factors could be included, such as the effects of temperature, self discharge, and finite data storage capacity. An additional layer of complexity can also be included to more closely reflect deployment conditions such as occlusion of solar panels due to seasonal vegetation. These more complex facets, when added to a simulation, remain easily repeatable. The energy management scheme can be improved by increasing the frequency of activity level updates, to better follow the amount of energy present in the buffer. In addition, splitting the transmission and measurement scaling values can be considered, as these can have very different associated energy costs. Inclusion of more nodes in the simulated network will allow for the addition of the network level energy management schemes, such as energy-aware message routing and dynamic cluster head selection.

Acknowledgments: This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Author Contributions: James Rodway and Petr Musilek conceived and designed the experiments; James Rodway performed the experiments; James Rodway and Petr Musilek wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

CART	Classification and regression tree
DE	Differential evolution
ELM	Extreme learning machine
MAPE	Mean absolute percentage error
MLP	Multilayer perceptron
NN	Neural network
RMSE	Root mean squared error
WSN	Wireless sensor network
WSU	Washington State University

References

1. Yang, S.H. *Wireless Sensor Networks: Principles, Design and Applications*; Signals and Communication Technology; Springer: London, UK, 2014.
2. Musilek, P.; Prauzek, M.; Krömer, P.; Rodway, J.; Bartoň, T. Intelligent Energy Management for Environmental Monitoring Systems. In *Smart Sensors Networks: Communication Technologies and Intelligent Applications*; Khafa, F., Leu, F.Y., Hung, L.L., Eds.; Intelligent Data Centric Systems; Elsevier: San Diego, CA, USA, 2017; pp. 67–94.
3. Pimentel, D.; Musilek, P.; Knight, A. Energy harvesting simulation for Automatic Arctic monitoring stations. In Proceedings of the 2010 IEEE Electric Power and Energy Conference (EPEC), Halifax, NS, Canada, 25–27 August 2010; pp. 1–6.
4. Watts, A.; Prauzek, M.; Musilek, P.; Pelikan, E.; Sanchez-Azofeifa, A. Fuzzy power management for environmental monitoring systems in tropical regions. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 1719–1726.
5. Martinez, K.; Padhy, P.; Elsaify, A.; Zou, G.; Riddoch, A.; Hart, J.K.; Ong, H.L.R. Deploying a sensor network in an extreme environment. In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), Taichung, Taiwan, 5–7 June 2006; Volume 1, p. 8.
6. Prauzek, M.; Krömer, P.; Rodway, J.; Musilek, P. Differential evolution of fuzzy controller for environmentally-powered wireless sensors. *Appl. Soft Comput.* **2016**, *48*, 193–206.
7. Shin, K.Y.; Song, J.; Kim, J.; Yu, M.; Mah, P.S. REAR: Reliable Energy Aware Routing Protocol for Wireless Sensor Networks. In Proceedings of the 9th International Conference on Advanced Communication Technology, Dublin, Ireland, 12–14 February 2007; Volume 1, pp. 525–530.
8. Chen, F.; Guo, L.; Chen, C. A Survey on Energy Management in the Wireless Sensor Networks. *IERI Procedia* **2012**, *3*, 60–66.
9. Niyato, D.; Hossain, E.; Rashid, M.; Bhargava, V. Wireless sensor networks with energy harvesting technologies: A game-theoretic approach to optimal energy management. *IEEE Wirel. Commun.* **2007**, *14*, 90–96.
10. Zadeh, P.H.; Schlegel, C.; MacGregor, M. Distributed optimal dynamic base station positioning in wireless sensor networks. *Comput. Netw.* **2012**, *56*, 34–49.
11. Kansal, A.; Hsu, J.; Zahedi, S.; Srivastava, M.B. Power Management in Energy Harvesting Sensor Networks. *ACM Trans. Embed. Comput. Syst.* **2007**, *6*, 32.
12. Renner, C.; Unterschütz, S.; Turau, V.; Römer, K. Perpetual Data Collection with Energy-Harvesting Sensor Networks. *ACM Trans. Sens. Netw.* **2014**, *11*, 12.
13. Hsu, J.; Zahedi, S.; Kansal, A.; Srivastava, M.; Raghunathan, V. Adaptive Duty Cycling for Energy Harvesting Systems. In Proceedings of the 2006 International Symposium on Low Power Electronics and Design (ISLPED'06), Bavaria, Germany, 4–6 October 2006; ACM: New York, NY, USA, 2006; pp. 180–185.
14. Moser, C.; Thiele, L.; Brunelli, D.; Benini, L. Adaptive Power Management for Environmentally Powered Systems. *IEEE Trans. Comput.* **2010**, *59*, 478–491.

15. Ferry, N.; Ducloyer, S.; Julien, N.; Jutel, D. 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*; Number 6951 in Lecture Notes in Computer Science; Ayala, J.L., García-Cámara, B., Prieto, M., Ruggiero, M., Sicard, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 122–132.
16. Prauzek, M.; Musilek, P.; Watts, A. Fuzzy algorithm for intelligent wireless sensors with solar harvesting. In Proceedings of the 2014 IEEE Symposium on Intelligent Embedded Systems (IES), Orlando, FL, USA, 9–12 December 2014; pp. 1–7.
17. AgWeatherNet at Washington State University. Available online: <http://www.weather.wsu.edu> (accessed on 1 April 2016).
18. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2016.
19. Bergmeir, C.; Benítez, J.M. Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS. *J. Stat. Softw.* **2012**, *46*, 1–26.
20. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32.
21. Fekete, S.P.; Kröller, A.; Fischer, S.; Pfisterer, D. Shawn: The fast, highly customizable sensor network simulator. In Proceedings of the Fourth International Conference on Networked Sensing Systems (INSS 2007), Brainschweig, Germany, 6–8 June 2007.
22. Rada-Vilela, J. Fuzzylite: A Fuzzy Logic Control Library. Available online: <http://www.fuzzylite.com> (accessed on 16 May 2016).
23. Gough, B. (Ed.) *GNU Scientific Library Reference Manual*, 3rd ed.; Network Theory Ltd.: Bristol, UK, 2009.
24. Breiman, L. *Classification and Regression Trees*; The Wadsworth Statistics/Probability Series; Wadsworth International Group: Belmont, CA, USA, 1984.
25. Loh, W.Y. Fifty Years of Classification and Regression Trees. *Int. Stat. Rev.* **2014**, *82*, 329–348.
26. De'ath, G.; Fabricius, K.E. Classification and Regression Trees: A Powerful yet Simple Technique for Ecological Data Analysis. *Ecology* **2000**, *81*, 3178–3192.
27. Robnik-Sikonja, M. Improving Random Forests. In *Machine Learning: ECML 2004*; Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D., Eds.; Number 3201 in Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; pp. 359–370.
28. Engelbrecht, A.P. *Computational Intelligence*; John Wiley & Sons, Ltd.: Chichester, UK, 2007.
29. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990.
30. Pedrycz, W.; Gomide, F. *Fuzzy Systems Engineering: Toward Human-Centric Computing*; John Wiley: Hoboken, NJ, USA, 2007.
31. Price, K.V.; Storn, R.M.; Lampinen, J.A. *Differential Evolution. [Electronic Resource]: A Practical Approach to Global Optimization*; Natural Computing Series; Springer: Berlin, Germany, 2005.
32. Das, S.; Suganthan, P. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evolut. Comput.* **2011**, *15*, 4–31.
33. Worasuchee, C. A new self adaptive differential evolution: Its application in forecasting the index of Stock Exchange of Thailand. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 1918–1925.
34. Worasuchee, C. High-dimensional function optimization with a self adaptive differential evolution. In Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems, Shanghai, China, 20–22 November 2009; Volume 1, pp. 668–673.
35. Jeyakumar, G.; Shanmugavelayutham, C. An empirical comparison of differential evolution variants for high dimensional function optimization. In Proceedings of the International Conference on Intelligent Agent Multi-Agent Systems, Chennai, India, 22–24 July 2009; pp. 1–6.
36. Chen, J.L.; Liu, H.B.; Wu, W.; Xie, D.T. Estimation of monthly solar radiation from measured temperatures using support vector machines—A case study. *Renew. Energy* **2011**, *36*, 413–420.
37. Wang, L.; Kisi, O.; Zounemat-Kermani, M.; Salazar, G.A.; Zhu, Z.; Gong, W. Solar radiation prediction using different techniques: Model evaluation and comparison. *Renew. Sustain. Energy Rev.* **2016**, *61*, 384–397.

38. Wang, L.; Kisi, O.; Zounemat-Kermani, M.; Zhu, Z.; Gong, W.; Niu, Z.; Liu, H.; Liu, Z. Prediction of solar radiation in China using different adaptive neuro-fuzzy methods and M5 model tree. *Int. J. Climatol.* **2017**, *37*, 1141–1155.
39. Jong, R.D.; Stewart, D.W. Estimating global solar radiation from common meteorological observations in western Canada. *Can. J. Plant Sci.* **1993**, *73*, 509–518.
40. Li, M.F.; Liu, H.B.; Guo, P.T.; Wu, W. Estimation of daily solar radiation from routinely observed meteorological data in Chongqing, China. *Energy Convers. Manag.* **2010**, *51*, 2575–2579.
41. Samani, Z.; Hargreaves, G.; Tran, V.; Bawazir, S. Estimating solar radiation from temperature with spatial and temporal calibration. *J. Irrig. Drain. Eng.* **2011**, *137*, 692–696.
42. Almorox, J. Estimating global solar radiation from common meteorological data in Aranjuez, Spain. *Turk. J. Phys.* **2011**, *35*, 53–64.
43. Adaramola, M.S. Estimating global solar radiation using common meteorological data in Akure, Nigeria. *Renew. Energy* **2012**, *47*, 38–44.
44. Brewer, E. *A Guide to the Scientific Knowledge of Things Familiar*; C.S. Francis & Co.: New York, NY, USA, 1851.
45. Vizard, F. Storm Warnings at Home. *Pop. Mech.* **1994**, *171*, 72.
46. Brown, A. Severe Weather Detector and Alarm. U.S. Patent 6,597,990 B2, 22 July 2003.
47. Organization, W.M. *Guide to Meteorological Instruments and Methods of Observation*, 7th ed; WMO (Series); Secretariat of the World Meteorological Organization: Geneva, Switzerland, 2008.
48. *BMP085 Digital Pressure Sensor Data Sheet*; Number BST-BMP085-DS000-05, Rev 1.2; Bosch Sensortec: Reutlingen, Germany, 2009.
49. Kalogirou, S. *Solar Energy Engineering: Processes and Systems*; Academic Press: New York NY, USA, 2013.
50. Singh, S. *Solar Photovoltaics: Fundamentals, Technologies and Applications*; Prentice-Hall of India Pvt. Ltd.: New Delhi, India, 2011.
51. Allen, R.G.; Pereira, L.S.; Raes, D.; Smith, M. *Crop Evapotranspiration—Guidelines for Computing Crop Water Requirements—FAO Irrigation and Drainage Paper 56*; FAO: Rome, Italy, 1998.
52. Stull, R.B. *Meteorology for Scientists and Engineers*; Brooks/Cole: Pacific Grove, CA, USA, 2000.
53. Rodway, J.; Musilek, P.; Lozowski, E.; Prauzek, M.; Heckenbergerova, J. Pressure-based prediction of harvestable energy for powering environmental monitoring systems. In Proceedings of the 2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC), Rome, Italy, 10–13 June 2015; pp. 725–730.
54. Rodway, J.; Krömer, P.; Karimi, S.; Musilek, P. Differential Evolution Optimized Fuzzy Controller for Wireless Sensor Network Energy Management. In Proceedings of the 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Nice, France, 21–23 September 2016.
55. Rodway, J.; Musilek, P. Wireless Sensor Networks with Pressure-Based Energy Forecasting: A Simulation Study. In Proceedings of the 2016 29th Canadian Conference on Electrical and Computer Engineering (CCECE), Vancouver, BC, Canada, 15–18 May 2016.
56. Musilek, P.; Kromer, P.; Rodway, J.; Prauzek, M. Pressure-based forecasting of next-day solar energy availability using evolutionary fuzzy rules. In Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Istanbul, Turkey, 2–5 August 2015; pp. 1–8.
57. Prauzek, M.; Musilek, P.; Krömer, P.; Rodway, J.; Stankus, M.; Hlavica, J. Energy Availability Forecasting for Harvesting-aware Wireless Sensor Networks: Analysis of Energy Demand of a Predictor Based on Evolutionary Fuzzy Rules. In Proceedings of the 2015 International Conference on Intelligent Networking and Collaborative Systems (INCOS), Taipei, Taiwan, 2–4 September 2015; pp. 200–203.
58. Ahmedy, I.; Ngadi, M.A.; Omar, S.N. Using store-forward technique to conserve energy in wireless sensor networks: Initial step for routing mechanism. In Proceedings of the 2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT), Seoul, Korea, 24–26 April 2012; Volume 2, pp. 671–676.

