

Information Quality Improvement With Task Selection Algorithm For IoT Energy Harvesting Devices

Minh Huan Vo

Ho Chi Minh University of Technology and Education, Vietnam

*Corresponding author. Email: huanvm@hcmute.edu.vn

ARTICLE INFO

Received: 10/04/2023
Revised: 27/05/2023
Accepted: 05/07/2023
Published: 28/08/2023

KEYWORDS

Machine learning;
IoT;
Wireless Sensor Network;
Battery life time;
Task selection.

ABSTRACT

The purpose of study is to propose a task selection algorithm that both keeps information quality and saves power consumption in IoT energy harvesting devices. The proposed algorithm not only keeps stable information quality but saves power loss also. The sensor node operation is divided into four tasks depending on the input data including battery capacity, solar panel charging current, and input sensor data variation. The task selector based on a neural network consists of an input layer, a hidden layer of 20 neurons, and an output layer. The proposed algorithm is different from the predefined task algorithm, which mainly focused on deep sleep mode or scheduled tasks. Our proposed algorithm helps the sensor node to be more adaptive to the environment based on real-time execution at each node. The collected information amount varies according to the input data variation. The experiment results show that the proposed algorithm collects higher quality information at large input data variation. The battery lifetime is also improved by up to 22%.

Doi: <https://doi.org/10.54644/jte.78A.2023.1375>

Copyright © JTE. This is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

1. Introduction

Sensor nodes are powered with limited power to save energy consumption in low-energy IoT devices. The node's supply power can be obtained from renewable energy sources such as solar energy [1]. The operation of a sensor node is divided into three parts: sensing, processing, and data transmission. Of the three tasks above, data transmission consumes the most energy, about 80% of the node's energy [2]. It is important to extend the operating life of the sensor node but still improve the quality of the information. The trade-off between information quality and energy availability is still a challenge for researchers. In recent studies, the number of algorithms have been proposed to optimize energy consumption by using scheduling algorithms, managing energy consumption, or optimizing routing of sensor networks [3]-[4]. In this paper, we propose a neural network-based task selection algorithm that can be implemented on microcontrollers to classify and predict executed tasks to continue the quality of information but still prolong the execution time and the node's lifetime.

Normally, sensor nodes save energy by putting the node into deep sleep [5]-[6]. However, when the node is in deep sleep, a large amount of information is lost and causes network latency by transitioning from deep sleep to active state. L. Wang proposes dynamic power management algorithm for sensor network with five modes that are based on the battery power level [6]. In fact, switching states from deep sleep to active or switching between active modes causes waste of energy and causes latency. Therefore, the decision to switch states needs to determine the threshold level and calculate the tradeoff. In the study of X. Fan [7], the author's algorithm proposes to improve the threshold calculation to decide whether a node should be in deep sleep or not. More specifically, the paper considers a comparison between the energy consumed and the energy saved when waking the sensor node back to the active state [5]. Sujesha and Sudevalayam optimize the battery life on the sensor node based on the energy source charged to the battery [8]. The author changes the parameters of the node based on the prediction of the battery power level and the current energy level in the battery to optimize the time of the battery being charged.

Some other published studies to optimize the battery life are the method of reducing the operating frequency of the microcontroller and reducing the consumption power on the sensor at the typical nodes [9]. High precision sensors are reduced or replaced with lower resolution and accuracy sensors. The above method has disadvantages that adversely affect the quality of the collected information. In addition, lowering the signal transmission power of the sensor network is mentioned by G. Amato in [10] and changing the data collection period of X.Chen in the study [11]. For sensor networks with short distances and low obstacles, this method is optimal. However, locations with a wide sensor network and many obstacles, the loss of information as well as the quality is significantly reduced. The above issues are investigated in the power management techniques of energy-collecting sensor networks [12].

In addition to the above methods, the task scheduling algorithm is also mentioned [13]-[18]. In the article [13], the author proposed a programming algorithm to optimize the total power consumption based on the usefulness as well as the energy consumed on each task. This algorithm schedules tasks, it uses the weather forecast information available at the beginning of each scheduling interval (usually a day) and the battery level available at the time of scheduling, thereby determining an optimized schedule. The main goal of this method is to find an average energy schedule over a long period of time [13]. In the article [14], M. Severini also presents an algorithm to collect data in a sequence, schedule like the article [13] but using the LSA (Lazy Scheduling Algorithm) model. The Lazy Scheduling Algorithm (LSA) model is a programming algorithm to schedule the operation of a rechargeable sensor node and limit the task of a sensor node [14]-[15]. LSA creates a running schedule for parallel tasks and will run permanently according to the schedule of Moser implemented in the paper [16]. The task scheduler will schedule the execution of tasks based on the energy storage parameter. The output of the algorithm is a list of tasks that will be executed sequentially and will not be changed while the sensor node runs. This is a suboptimal point of this algorithm. However, LSA minimizes the inputs to increase the performance of the algorithm and is a static algorithm. It does not add additional states at the input which cannot cancel the scheduled tasks as well as add the task during execution mentioned in the research during the running of the scheduled task [14]-[15], [17]. Caruso proposes a dynamic algorithm to program the schedule that can change based on the input parameters of the battery and the charging current of the battery [18]. Since this dynamic algorithm can change the schedule, it is more adaptive than LSA. However, with a dynamic algorithm, the memory management of the dynamic algorithm is very important. In order to minimize repetitive computations when using recursion, the previously calculated values of the dynamic algorithm will be put into an array. When sensor nodes perform a lot of computation, the memory increases and becomes unmanageable during the computation algorithm. Thus, there may arise a hang or restart causing the microcontroller to recalculate from the beginning. Loreti leverages the Lyapunov drift-plus-penalty theory to produce a StableSENS algorithm [19]. It provides an adaptive sampling algorithm aimed at maximizing a stable sampling rate. Recently, Reinforcement Learning was proposed to schedule parallel tasks for autonomous driving. Based on different computing ability and resource, these tasks are assigned to nodes by machine learning algorithms [20].

The methods mentioned above do not take into account the variability of the data. This is one of the important factors for making accurate predictions and decisions. In some practical applications, such as aquaculture, sudden changes in the data pose the risk of significant economic harm. For example, in the shrimp culture model [21], the temperature and pH of the water need to be stable. When the data has a sudden change of the above values, farmers need more information to predict and make quick and accurate decisions. In a forest data collection application, we consider temperature variations that can predict problems with wildfires and floods. Therefore, data variability is an important input component in the data collection model.

To overcome the above inadequacies, we propose an algorithm that uses task selector based on neural network algorithm to classify and predict the defined task levels from input data including battery quantity, charging current and temperature data variation, thereby selecting tasks with different levels of energy loss. This model is suitable for programming problems with real-time operation cycles and can be implemented on microcontrollers. We designed the algorithm to work more flexibly with changing tasks. Instead of continuous sampling and sending in a cycle, this model depends on the input

data of the node to determine the task, which is more adaptive than the conventional task scheduler [13]. Traditionally, the classification algorithm mainly uses classical machine learning models such as decision tree, support vector machine, logistic regression, reinforcement learning and naïve Bayes [3], [19], [20]. These algorithms have achieved good performance on classification tasks. However, they manually require feature extraction which is costly and complicated. Thus, when classification task is changed, feature extraction is inefficient to be adjusted dynamically. Recently, the deep learning models have developed rapidly, the task classification has gradually shifted from models based on these classical machine learning models to deep learning models based on neural network. The neural network is able to support feature extraction. Thus, it shows a better model in multi-task than the traditional machine learning based models. We rely on a model of a neural network [22]-[23]. The proposed algorithm can perform algorithmic computation on hardware with limited processing speed and memory, such as 32-bit arm-cortex microcontrollers, without the need to do so on high-speed computer. The calculation is not repeated like when using recursion and can manage memory when using dynamic problems [18]. The research paper considers more in the model and focuses on the problem of programming the optimal task selection at each sensor node at that time. In addition, the algorithm can be effectively implemented in low-power nodes in real-world conditions.

The purpose of this research is to implement this task selection algorithm based on neuron network on limited resource hardware. The task selection algorithm is implemented on hardware to gain not only keep quality information but also lower power loss. Thus, the sensor node is autonomous in the surrounding environment. Thus, the autonomous sensor is capable of independently collecting and processing data without direct human intervention. It means it can function and make decisions without constant human oversight.

The research paper is organized into 4 sections. Section 1 introduces and provides an overview of previous studies. Section 2 describes the process of designing task selectors based on neural algorithms. Section 3 presents and discusses the results obtained. Finally, conclusions and future directions will be given in section 4.

2. IoT energy harvesting device design

The IoT sensor node is designed in Figure 1. The sensor collects the temperature, the charging current from the solar cell and the battery voltage. The input parameters for the Task selector algorithm block include sensed data such as battery voltage, solar panel charging current and temperature. Task selector algorithm is performed based on the neural network model. The sensed data is preprocessed before applying to the neural network model in the task selector algorithm. The battery voltage is transformed into the percentage of remaining energy in the battery. The temperature is transformed into the delta deviation of the temperate data over a period. When the data changes suddenly, the deviation will be high and vice versa. The final input parameter is the charging current of the solar battery. The output gives the ability to predict and select the most appropriate task.

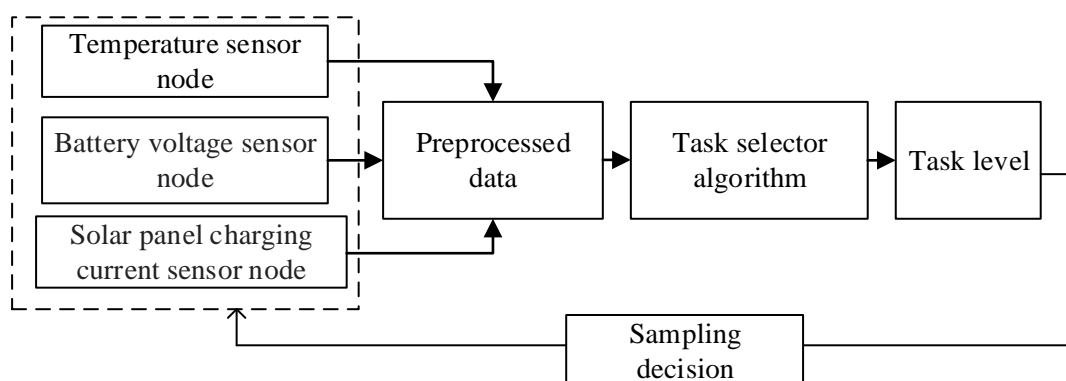


Figure 1. Low power IoT sensor node diagram

The transformed data is calculated according to formulas (1) (2) and (3) and then fed into the proposed algorithm. The remaining battery percentage is the range of the battery capacity limited to [0, 100], which is read through the microcontroller ADC. The maximum battery level is $V_{max} = 4.3V$ and the depleted battery level is $V_{min} = 3.7V$. The remaining battery voltage percentage can be presented by the following equation (1).

$$E = \frac{V_i - V_{min}}{V_{max} - V_{min}} * 100\% \quad (1)$$

Where V_i is input battery level.

The charging current of the panel [0, 250] mA is obtained from the sensor INA-219 based on the voltage drop across the shunt resistor $R_{sensing}$ with low error 0.01Ω . V_1 and V_2 are the voltages on the two shunt resistors, G is the amplification factor. The experimental INA-219 module has $G = 1$.

$$I_{charge} = G * \frac{(V_2 - V_1)}{R_{sensing}} \quad (2)$$

Delta data deviation is defined as the standard deviation of a data collection cycle. In one cycle, N samples are taken in time T . The standard deviation of the data for a period is expressed as follows

$$Delta = \mu \sqrt{\frac{\sum_i^N (x_i - X)^2}{N - 1}} \quad (3)$$

In equation (3), the coefficient μ is a selection factor and it is adjusted for different surrounding with different temperature variations. Where Σ symbol means sums of terms. x_i is a value in dataset. i is from 1 to n samples. X is the mean of dataset. N is the number of data points in the population.

The desired task output consists of four task levels corresponding to 4 levels 1, 2, 3, 4. Task operation, power consumption level and task level output are shown in the following table. The listed tasks consume different amounts of energy with four different levels from very low to high.

Table 1. Classification of four task levels corresponding to amount of power consumption.

Task level	Task operation	Power consumption level
1	Do not send data	Very low
2	Once a day to send data	Low
3	Sending K samples per day	High
4	Sending data continuously in 2 K samples a day	Very high

Task level 1 is executed when no data is sent. In fact, the data is still sampled in a certain period, but the data may overlap with the previous data, or the data has little change compared to the previous one. Thus, it is blocked by the filter. To perform task operation 1, the sensor node is designed with a data comparison filter to extract features that are different from the previous data. If task level 1 is executed, the node power consumption is very low because no data is sent. Task level 2 sends data at the end of a data collection day. This process updates data information day by day even if the data has not changed. Tasks level 3 and 4 execute sending data more often because of data changes. More information needs to be sent to ensure the quality of collected information when the input changes a lot. However, task level 4 will consume the most power.

The data sets were collected directly in Thu Duc district, Ho Chi Minh. We design a device using Arduino Pro Mini, 3.7V 2000mAh Lithium battery, 4.5W solar battery and 250mA max current charging circuit. The data set was collected by three sensors of temperature, battery voltage and charge current with approximately 1000 samples. A sample is taken every 30 minutes. The data sample covers the

temperature variation and intensity of the solar during a day. Therefore, the data collected for training and testing can be considered random variables. The data samples are distributed widely to cover all possible data in one day, making the evaluation reliable.

The conventional predefined schedule algorithm will run tasks continuously according to the predetermined schedule without changing the desired task. It is different from the fixed task scheduler programming algorithm, the proposed task selector is based on the neural network model. This proposed model can respond to its ability to change its working task from the data input variation. We design the task selector based on neural network model. Neural network has 3 main layers including 1 input layer, 1 hidden layer and 1 output layer. The input layer consists of three neurons that are responsible for battery energy, solar battery charging current and variation of collected data. After collecting data, we proceed to train the model with Matlab tool using supervised learning. We use a feed-forward back-propagation network with parameters epochs = 1000000, learning rate = 0.3, goal = 0.05, using the activation function sigmoid.

The input layer consists of the battery capacity, the difference between the charging current and the average load current of the running task, and finally the standard deviation of the latest data with the average of the data. The output layer has 1 neuron corresponding to the value that the next task to perform for the Sensor Node. The output value corresponding to tasks level 1, task level 2, task level 3, and task level 4 is an element of a vector [1 2 3 4].

In the first level task, the sensor node is put to sleep mode. At this time, the sensor node stops all wireless transmission operations. In the second level task, the sensor node still collects temperature data and stores it in the microcontroller's internal memory. However, these values are transmitted at the end of a day's cycle. In the third level task, the sensor node is returned to normal operation mode, the data will be sampled periodically and sent when the cycle is over. The four-level task is the most intensive energy consumption because the sensor node collects and sends twice as much information as the third level task. Task level 4 will only be active when the temperature suddenly change and the remaining power in the battery is high.

3. Experiment Result

It is different from the fixed task scheduler algorithm such as the predefined schedule algorithm as mentioned in Section 2. The proposed classifier can change its working task according to the input data variation, respectively. Meanwhile, the predefined schedule algorithm runs continuously by the fixed schedule. It does not change to the desired task regardless of variation in environment conditions. The neuron network model can be implemented to adjust the desired tasks even in the limited hardware devices. The neuron network is designed by a small number of hidden layers and a small number of neurons. It shows their advantages in terms of computation or memory. In order to reduce computation and memory, a neural model-based task selector is basically designed with a hidden layer of 20 neurons. The error value converges to 0.05 value after 15611 iterations as shown in Table 2. It takes 35 minutes to train the neural network to reach the desired error value. In the case of the network with 25 and 50 neurons, the values converge to 0.005 with over 20,000 iterations as shown in Table 2.

Table 2. Parameters after training

Hidden neurons	The number of epochs when converged to 0.005	Training time (minutes)
20	15611	35
25	21299	45
50	22020	47

A data set of 100 random samples is included in the Simulink model to test the model's best ability. The neural network output shows 4/100 wrong tasks. It means the accuracy is acceptable to 96%.

The following delta values are derived during a day at a random location. Then, this value is calculated according to formula (3) with N slots of 10. From the above data, we change the coefficient μ corresponding to many different environments (here we performed with $\mu=1$).

Table 3. Prediction rate of system with task selector model based on network with hidden layer including 20, 25, 50 different neurons.

The number of Neurons at hidden layer	Accuracy
20	93%
25	96%
50	97%

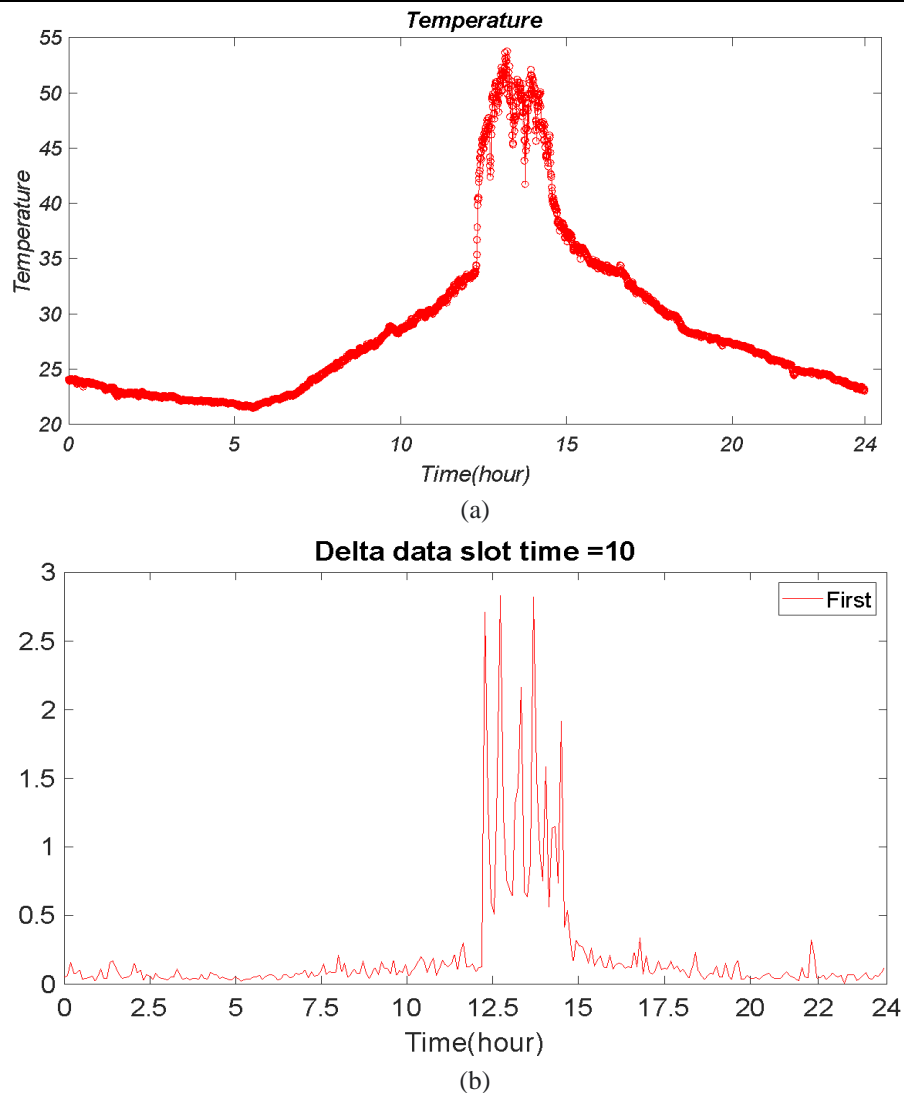


Figure 2. (a) Actual temperature collected in a day with sensor error 0.5°C (b) Value of delta data with $N = 10$.

The delta deviation value of the temperature is an important factor determining the task level selection by the neural network. In a short period of time, the temperature background has many fluctuations, for example abnormally up and down, the deviation value in formula (3) will increase based on the standard deviation of a data set.

Fig. 2 (a) shows the temperature in a day collected in the real environment. The temperature changes from 27 to 35 degrees during the period of 9 to 11 hours. These degree values increase slowly without sudden changes. However, the information of temperature changes suddenly from 13h to 15h, leading to the standard deviation in formula (3) will increase. In this period, more information needs to be sampled for analysis and timely processing. Thus, the algorithm should move to a suitable task level for

more data collection. The effects of sudden temperature fluctuations are important in aquaculture applications.

The N factor of formula (3) is selected based on the microcontroller's memory. Microcontrollers have small RAM memory. The reduction of N factor helps to do less computation and maintain higher performance, lower power consumption. In addition, subdividing the data set so that the calculation can more accurately evaluate the temperature change in a fixed time period. We chose N as 10. The delta value falls from 0 to close to 3 as shown in Fig. 2 (b). Delta values fluctuate sharply at noon from 11am to 2pm. The conventional algorithms usually do not determine the delta coefficient during this time period that has frequent spikes. Thus, the delta value is usually not considered to decide a task in scheduled tasks as mentioned above. During time period from 7h to 9h, the delta values need not be taken much because the delta deviation do not deviate much.

As shown in Figure 3, the collected information amount is the most in one minute sample cycle. It means that information is sampled every minute. There are 120 samples for two hours. However, there is a lot of duplicates, useless information, and a lot of wasted energy in this case because of information is not fluctuated so much. In two-minute sample cycle, the variation temperature sample amount has no adaptation that we need more information because of temperature variation in that period time. The proposed algorithm offers a flexible sample method. The more data is fluctuated, the more values are collected. The less temperature changes in a day, the less sampling data is taken to ensure the limitation energy of the sensor node. The battery voltage is collected through the microcontroller's internal ADC. The current from the battery is passed through a passive RC low pass filter (LPF) to minimize power noise and retain the main DC component.

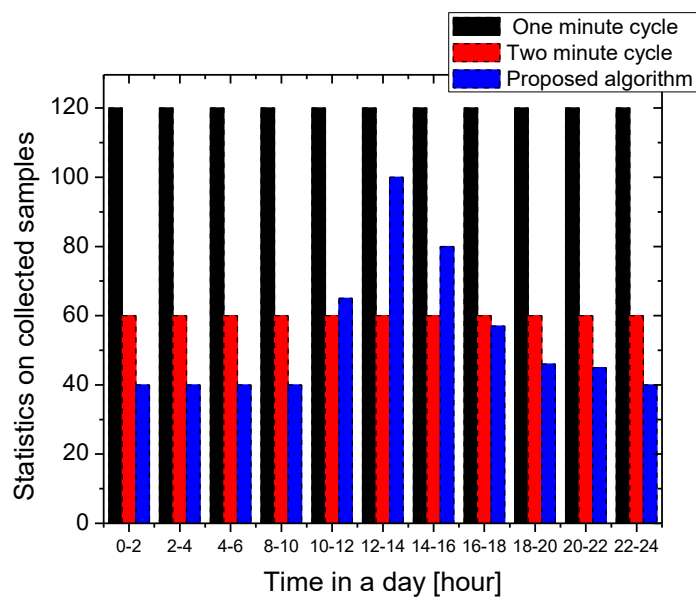


Figure 3. Statistics on collected samples, the Ox-axis is the time of day, the Oy axis is the number of collected samples.

The battery voltage fluctuates unevenly for many objective reasons such as cloud cover from the sun. The battery capacity reaches its maximum during midday. On rainy days, the battery will not reach 6V voltage. With a recommended battery capacity of 2000 mAh, the ability to fully charge the battery reaches 100% in a day with ideal sunny conditions. For days with low sun coverage (2h-4h/day), the battery is charged at least 80%. Carrying out the proposed algorithm with the condition that the charging current of the battery is zero, the voltage of the battery is collected simultaneously to demonstrate the energy responsiveness of the algorithm. The hardware to perform the above 2 methods includes a microcontroller STM32F103, lora SX1278 433Mhz, 2000mah battery, CN3065 charge management MCU, 6V 4.5W solar battery, DS18B20 temperature sensor. Current consumption of STM32F103= 1.19 mA, with the μ C running on its internal 8 MHz RC clock. The lora SX1278 433Mhz in receiving mode

is less than 10.8mA and in transmitting mode is less than 120mA. CN3065 charge management is 650 uA in average current. The sensor DS18B20 operates on a 3V to 5.5V power supply and draws only 1mA during active temperature conversions. The storage battery uses 2000mAh Lithium battery. Thus, the battery capacity can supply the system for a full day operation. The purpose of study is to gain both low power system and stable information quality. After compiling the program, the proposed algorithm takes up 12% of the flash memory.

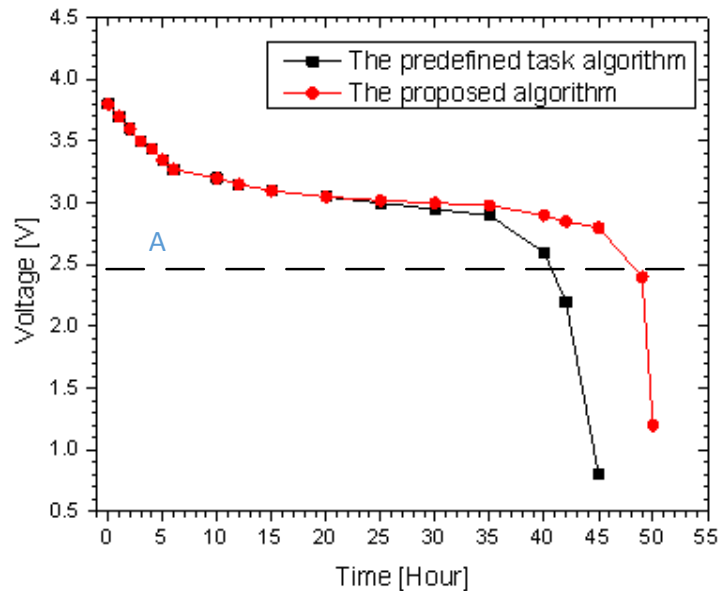


Figure 4. Comparison of battery voltage drop of the predefined task algorithms and the proposed algorithm

Figure 4 compares battery voltage drop in two cases using the predefined task algorithm and neural based algorithm. In the period from 0 to 25th hour, the battery capacity is high, and the temperature variation is quite high, so the algorithms proposes to execute task level 4. The battery decreases quickly but samples are collected more. During the period from hours 25 to 40, when the battery level is low about 3V, the sensor node performs tasks 2 and 3 to save energy of the sensor node. Figure 4 shows that the lifetime of the node is extended. At dashed line A with the same 2.5V voltage drop, the predefined task algorithm works d 41 hours while the proposed algorithm works for 50 hours. That is, the life of the battery increases by 22%.

4. Conclusions

This paper presents an adaptive algorithm for task selector in IoT energy harvesting device. The adaptive selector is based on a neural network with an input layer, a hidden layer, and an output layer. However, the more neurons in the hidden layer, the heavier the computational system, resulting in more energy consumed on the node. The design with 20 hidden layer neurons helps to reduce resources from memory and computing power at sensor nodes. On the other hand, the algorithm has a more optimal balance between the two important factors of a low-power data collection system, that is the quality of the information and the battery life. The amount of battery loss is similar to the predefined task algorithm, but data variation is sampled more and less when the data changes are not significant. In addition, the algorithm is well adapted to each environment because the coefficients of the neural network can be changed based on the collected information. In near future, the study plans to compare the proposed neural network model with the other classical machine learning models such as support vector machine, logistic regression and random forest. The study also collects more data with various weather conditions in different seasons to show the model accuracy.

Acknowledgment

This work belongs to the project in 2023 funded by Ho Chi Minh City University of Technology and Education, Vietnam.

REFERENCES

- [1] A. A. Babayo, M. H. Anisi, and I. Ali, "A Review on energy management schemes in energy harvesting wireless sensor networks," *Renew. Sustain. Energy Rev.*, vol. 76, pp. 1176–1184, 2017.
- [2] N. Kimura, V. Jolly, and S. Latifi, "Energy restrained data dissemination in wireless sensor networks," *Int. J. Distrib. Sens. Networks*, vol. 2, no. 3, pp. 251–265, 2006, doi: 10.1080/15501320600642692.
- [3] T. Ujzdowski and R. Piotrowski, "Task Scheduling–Review of Algorithms and Analysis of Potential Use in a Biological Wastewater Treatment Plant," *IEEE Access*, vol. 10, pp. 45230–45240, 2022, doi: 10.1109/ACCESS.2022.3170105.
- [4] K. D. Kang, H. Park, G. Park, and D. Kim, "Improving the Efficiency of Power Management via Dynamic Interrupt Management," in *2020 IEEE 38th International Conference on Computer Design (ICCD)*, 2020, pp. 377–380, doi: 10.1109/ICCD50377.2020.00069.
- [5] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Des. Test Comput.*, vol. 18, no. 2, pp. 62–74, 2001, doi: 10.1109/54.914626.
- [6] L. Wang and Y. Xiao, "A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks," *MONET*, vol. 11, pp. 723–740, Oct. 2006, doi: 10.1007/s11036-006-7798-5.
- [7] X. Fan, "Sensors Dynamic Energy Management in WSN," *Wirel. Sens. Netw.*, vol. 02, pp. 698–702, Jan. 2010, doi: 10.4236/wsn.2010.29084.
- [8] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surv. Tutorials*, vol. 13, no. 3, pp. 443–461, Sep. 2011, doi: 10.1109/SURV.2011.060710.00094.
- [9] P. Pillai and K. Shin, "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems," *ACM SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 89–102, 2001, doi: 10.1145/502034.502044.
- [10] G. Amato, A. Caruso, and S. Chessa, "Application-driven, energy-efficient communication in wireless sensor networks," *Comput. Commun.*, vol. 32, pp. 896–906, Mar. 2009, doi: 10.1016/j.comcom.2008.12.022.
- [11] X. Chen, H. B. Chen, W. Ma, X. Li, and S. X. Tan, "Energy-efficient wireless temperature sensing for smart building applications," in *2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, 2016, pp. 680–683, doi: 10.1109/ICSICT.2016.7999010.
- [12] J. Khan, H. Qureshi, and A. Iqbal, "Energy management in Wireless Sensor Networks: A survey," *Comput. Electr. Eng.*, vol. 41, pp. 159–176, 2015, doi: 10.1016/j.compeleceng.2014.06.009.
- [13] S. Escolar, A. Caruso, S. Chessa, and X. D. Toro, "Statistical Energy Neutrality in IoT Hybrid Energy-Harvesting Networks," in *2018 IEEE Symp. Comput. Commun. (ISCC)*, Natal, Brazil, 2018, pp. 444–449, doi: 10.1109/ISCC.2018.8538532.
- [14] M. Severini, S. Squartini, and F. Piazza, "Energy Aware Lazy Scheduling Algorithm for Energy-Harvesting Sensor Nodes," *Neural Comput. Appl.*, vol. 23, pp. 1899–1908, Dec. 2013, doi: 10.1007/s00521-012-1088-x.
- [15] C. Moser, J. Chen, and L. Thiele, "Dynamic power management in environmentally powered systems," in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2010, pp. 81–88, doi: 10.1109/ASP-DAC.2010.5419916.
- [16] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Lazy Scheduling for Energy Harvesting Sensor Nodes," in *IFIP Working Conference on Distributed and Parallel Embedded Systems*, vol. 225, Springer, Boston, USA, pp. 125 – 134, doi: 10.1007/978-0-387-39362-9_14.
- [17] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Adaptive Power Management in Energy Harvesting Systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2007, pp. 773–778.
- [18] A. Caruso, S. Chessa, S. Escolar, X. Del Toro, and J. C. López, "A dynamic programming algorithm for high-level task scheduling in energy harvesting IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2234–2248, 2018, doi: 10.1109/JIOT.2018.2828943.
- [19] P. Loreti, L. Bracciale, and G. Bianchi, "StableSENS: Sampling Time Decision Algorithm for IoT Energy Harvesting Devices," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 9908–9918, 2019, doi: 10.1109/JIOT.2019.2933335.
- [20] Q. Qi *et al.*, "Scalable Parallel Task Scheduling for Autonomous Driving Using Multi-Task Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13861–13874, 2020, doi: 10.1109/TVT.2020.3029864.
- [21] P. Zhang, X. Zhang, J. Li, and G. Huang, "The effects of body weight, temperature, salinity, pH, light intensity and feeding condition on lethal DO levels of whiteleg shrimp, *Litopenaeus vannamei* (Boone, 1931)," *Aquaculture*, vol. 256, no. 1, pp. 579–587, 2006, doi: <https://doi.org/10.1016/j.aquaculture.2006.02.020>.
- [22] R. E. Uhrig, "Introduction to artificial neural networks," in *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*, vol. 1, pp. 33–37, doi: 10.1109/IECON.1995.483329.
- [23] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996, doi: 10.1109/2.485891.



Vo Huan Minh received the B.S. and M.S.E.E. degrees in Electronics and Communication Engineering from the Ho Chi Minh City University of Technology, Vietnam in 2005 and 2007. and Ph.D. degree in Electronics Engineering from Kookmin University, Seoul, Korea in 2013.

He is currently working as a lecturer at the Faculty of Electrical and Electronics Engineering, University of Technology and Education, Ho Chi Minh City, Vietnam. His current research interests include neuromorphic computation using emerging technology, and low power system design. Email: huanvm@hcmute.edu.vn