Article

Predicting the Energy Consumption of a Robot in an Exploration Task Using Optimized Neural Networks

Liesle Caballero ^{1,*} , Álvaro Perafan ² , Martha Rinaldy ² and Winston Percybrooks ²¹ Faculty of Engineering, Universidad ITSA, Barranquilla 080002, Colombia² Biomedical Signal Processing and Artificial Intelligence Laboratory BSPAI, Universidad del Norte, Barranquilla 080002, Colombia; perafana@uninorte.edu.co (Á.P.); mrinaldy@uninorte.edu.co (M.R.); wpercyb@uninorte.edu.co (W.P.)

* Correspondence: licaballero@itsa.edu.co; Tel.: +57-315-257-7588

Abstract: This paper deals with the problem of determining a useful energy budget for a mobile robot in a given environment without having to carry out experimental measures for every possible exploration task. The proposed solution uses machine learning models trained on a subset of possible exploration tasks but able to make predictions on untested scenarios. Additionally, the proposed model does not use any kinematic or dynamic models of the robot, which are not always available. The method is based on a neural network with hyperparameter optimization to improve performance. Tabu List optimization strategy is used to determine the hyperparameter values (number of layers and number of neurons per layer) that minimize the percentage relative absolute error (%RAE) while maximize the Pearson correlation coefficient (R) between predicted data and actual data measured under a number of experimental conditions. Once the optimized artificial neural network is trained, it can be used to predict the performance of an exploration algorithm on arbitrary variations of a grid map scenario. Based on such prediction, it is possible to know the energy needed for the robot to complete the exploration task. A total of 128 tests were carried out using a robot executing two exploration algorithms in a grid map with the objective of locating a target whose location is not known a priori by the robot. The experimental energy consumption was measured and compared with the prediction of our model. A success rate of 96.093% was obtained, measured as the percentage of tests where the energy budget suggested by the model was enough to actually carry out the task when compared to the actual energy consumed in the test, suggesting that the proposed model could be useful for energy budgeting in actual mobile robot applications.

Keywords: machine learning; energy consumption; energy prediction; artificial neural network

Citation: Caballero, L.; Perafan, Á.; Rinaldy, M.; Percybrooks, W. Predicting the Energy Consumption of a Robot in an Exploration Task Using Optimized Neural Networks. *Electronics* **2021**, *10*, 920. <https://doi.org/10.3390/electronics10080920>

Academic Editors:
Francisco Serradilla-García,
Alberto Díaz-Álvarez and Luis
Fernando de Mingo López

Received: 31 January 2021

Accepted: 4 March 2021

Published: 13 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Energy consumption is a key issue for mobile robotics [1]. Many mobile robots are currently deployed on industrial scenarios, and many more are expected to be in operation in the near future. Typically, like in shipping warehouses, these robots are powered by batteries and are required to autonomously search for and move objects from one location to another. It is important that these robots accomplish their jobs in an energy efficient and secure way [2], so productivity will not be affected due to a robot depleting its batteries in the middle of an assignment [3,4]. Battery management becomes then a critical aspect in the operation of these robots [5]. Having a mechanism to compute the energy than the robot will require to finish a particular job will be a useful tool for a battery management system [6], allowing a smoother scheduling of the available robots.

Different approaches for predicting energy consumption in mobile robotics can be currently found in the literature. For example, by applying terramechanical models on deformable terrains [7]. However, in industrial applications, robots commonly move on surfaces with little or no deformation. Other state of the art model presented in [2,8–13] consider the energy requirements of several of its components to make the

energy prediction and/or power analysis. For his part [8] proposes that the robot calculates and predicts energy consumption by considering three main factors: the sensor system, the control system and the motion system. A mathematical model is developed for each of those factors and the relationship between the three systems is elaborated using explicit formulas. The calculations are related to the speed of the robot and the characteristics of the robot itself.

“Usually, energy consumption modeling for industrial robots is based on dynamic parameters identification. However, since the parts of industrial robots are not easy to be disassembled and the sensor modules cannot be installed easily inside robots, it is difficult to obtain all such physical parameters through sensing method, in particular the torque data” [14]. In this context, [14] proposes a practical method of energy modeling by measuring the total power consumed by the robots, thus avoiding the problem of directly measuring the relevant parameters within the robots. The experimental results of [14] show that the proposed energy modeling method can be used to predict the energy consumption of the robot.

Therefore, the lack of availability of the robot model on many occasions, the large number of variables and parameters that are required to know to build a dynamic and/or kinematic model and the particularity of the existing models, are problems that must be addressed.

For their part, scientists from Stanford University, the Massachusetts Institute of Technology and the Toyota Research Institute applied machine learning models to predict how many more cycles each battery could last. The combination of experimental data and artificial intelligence revealed the key to diagnose and predict the condition of batteries. The results obtained from the predictions were correct 95 percent of the time. This method of machine learning speeds up and reduces research costs [15].

In this work, we propose an energy consumption estimation approach that does not need an explicit dynamic or kinematic model of the robot. In STAGE 1, a machine learning predictor of the performance (number of movements needed to finish the task) of the particular exploration algorithm employed by the robot is created. Next, in STAGE 2, a linear model relates the electrical discharge in the robot’s battery with the variable predicted in STAGE 1, thus estimating the energy needed for the robot to finish the task using that particular exploration algorithm. The core of the proposed method is a machine learning system with supervised training which exploits our previous work in [16–18].

For the work presented here, the correlation coefficient (R), which computes the similarity between actual measurements and predicted data, and the Percentage Relative Absolute Error (%RAE), which computes the difference between measured values and their estimates but relative to the magnitude of the measured values, were selected as evaluation metrics for the predictive model in STAGE 1. This is consistent with the metrics used for evaluation in our previous related works described in [16–18].

The results in [17,18] show that machine learning models can estimate well the performance of robot exploration algorithms, both deterministic and non-deterministic, for untested experimental conditions. These previous results also demonstrated that among the machine learning techniques that were considered, Artificial Neural Networks (ANNs) attained better prediction performance. These previous works also evidenced that when comparing the Bayesian Regularization training algorithm with other training algorithms such as Levenberg-Marquardt and Scaled conjugate gradient, prediction error was reduced. In [16,18] it was shown that the proposed method works well not only for non-deterministic exploration algorithms, such as Random Walk (RW) and Random Walk without step (RW_WSB), but also for other exploration algorithms such as Q_Learning, which, after training, have a deterministic behavior.

The work in [19] defines the optimization of a problem P, as: “the task of looking for parameter values that applied to P, satisfy certain expectations. For each problem P, an objective function f that measures the suitability of each possible solution of P is defined. The domain of f , that is, the set of points that can be proved as a solution to the

problem, is called the space of solutions S . In an optimization problem, it is possible to find many areas of S with relatively good solutions (local optimums), while a single area of S , or a few ones in the best case, provide the best overall solution (global optimum)". Under this definition, Tabu list optimization is proposed to search the ANN topology (i.e., number of layers and number of neurons per layer) that maximizes R and minimizes %RAE. This metaheuristic method was selected because it has proven its effectiveness in combinatorial approaches with ANNs [20], despite there are other optimization algorithms applied to similar problems, such as the Grey wolf optimizer which is used in [21] to tune PI-fuzzy controllers with a reduced process parametric sensitivity. It is also used in [22] to tune a second order intelligent proportional-integral fuzzy control of twin rotor aerodynamic systems. Other recent optimization algorithm is the Island-based cuckoo search with highly disruptive polynomial mutation which mimics the nesting and parasitic reproduction behaviors of some species of cuckoo (CS) and includes improved variation to empower its capability to control the diversity of its population and tries to improve the exploration of the CS [23].

The remainder of this document is organized as follows: Section 2 describes the experimental setup; Section 3 explains the implemented method; in Section 4 experimental results are presented and discussed and finally, Section 5 contains the conclusions.

2. Experimental Setup

The industrial environments, such as warehouses, where mobile robots are currently being deployed are typically setup to favor straight line movements, with intersections as turning points and some sort of guiding lines, either physical, radio frequency or other, to mark the possible paths for the robots [4]. Therefore, for this work we selected a test scenario based on a two-dimensional grid, where a robot (iRobot Create 2) [24] explores the grid by moving from intersection to intersection limited to four directions: up, down, left or right, until it locates a target (colored square on the floor) whose location is unknown a priori by the robot, as shown in Figure 1a,b. The following variables are used to setup the grid: id size, target location, number of obstacles and their locations, exploration algorithm. The robot's staring point is fixed to the upper-left corner of the grid.

Figure 1. Experimental scenario. (a) Paths; (b) Real scenario with a 7×7 square grid.

Two microprocessors (Arduino nano and Raspberry Pi 3) are added to the Create 2, connected by a communication cable to the robot's internal controller in order to provide additional sensors and computing capabilities, see Figure 2a,b. A color sensor (TCS3200) is connected to the Arduino nano to check for the target square at the current intersection reached by the robot, see Figure 2b. When the target is detected, a signal is sent from the Arduino to the Raspberry Pi, indicating that the target has been located. Additionally, two optical-reflective sensors (TCRT5000) see Figure 2c, are connected to the Raspberry Pi computer to implement a line following function. Square grids with various sizes: 3×3 ,

4×4 , 5×5 , 6×6 and 7×7 , are selected considering the physical dimensions of the area available for validation and testing experiments.

Figure 2. Robot's hardware setup. (a) Raspberry Pi 3 added to the Create 2 Robot; (b) Color sensor connected to the Arduino nano; (c) Two optical-reflective sensors.

Regarding software, the Raspberry Pi has the Raspbian operating system installed and the Python programming language was used to implement the exploration algorithms, communications, control and sensing on the robot; the code is available at [25]. Python was selected because of the availability of the PyCreate2 library [26] which facilitates the communication between the Raspberry Pi and the robot to program the exploration algorithms. Likewise, an Arduino program is used to implement the algorithm that detects the target square.

On the other hand, the neural networks and optimization algorithms were implemented in Matlab and the code is available at [27].

Taking into account that we worked with a small database, the data processing did not have special memory requirements. A computer with a 1.8 GHz Intel Core i7 processor (Intel, Santa Clara, CA, USA), with 8 GB of RAM and a 64-bit operating system was used to perform the neural network training in 20 h. It should be noted that network training is a one-time process. Once the model is obtained, a prediction takes approximately 1 s to 2 s to execute.

3. Method

Artificial neural networks are a computational model that consists of a set of artificial neurons connected to each other to transmit signals. Each neuron is connected to others through links. In these links the output value of the previous neuron is multiplied by

a weight value that can increase or inhibit the activation state of the adjacent neurons. Similarly, at the output of the neuron, there may be a non-linear activation function, which modifies the result value or imposes a limit that must not be exceeded before propagating to following neurons. These systems are trained by minimizing a loss function that compares known outputs against the current outputs of the network. The values of the weights of the neurons are updated seeking to reduce the value of the loss function [28]. These systems are widely applied in areas where the detection of solutions is difficult to express with conventional programming.

The results presented in [16–18] demonstrated that a system based on machine learning using ANN with Bayesian regularization predicts well the performance of exploration algorithms applied to mobile robotics. In particular, such ANN model can estimate the maximum number of steps that the robot will spend to locate the target in a two-dimensional grid, given the following inputs: grid size, number of obstacles and the target location. Two purposely built databases [29,30] are used to train, validate and test an ANN model with supervised learning.

In [16,18] different optimization strategies for the ANN model architecture were compared: hand tuning, Hill Climbing, Hill Climbing-Random Restart and Tabu List; concluding that Tabu list converges faster than the others and delivers the best performing ANN architecture [16].

Tabu List is a mathematical optimization method, belonging to the class of local search techniques. The Tabu List increases the performance of the local search method by using memory structures: once a potential solution is determined, it is marked as “Tabu” so that the algorithm does not revisit that possible solution [31].

The parameters of the optimization algorithm that are selected by the user are: Maximum number of iterations and stop condition. This last parameter is the allowed error value and is selected taking as a reference the R and % RAE values to be improved. Other parameters such as: Starting point of the search for local minimums (for the case of % RAE) or local maximums (for the case of R) are random.

The block diagram in Figure 3 shows our proposed system to use the ANN model introduced above for the prediction of the energy consumption of a robot in an exploration task. The process consists of two stages: STAGE 1 is responsible for training a predictive model based on an ANN with Tabu list-based hyperparameter optimization to obtain maximum R and minimum %RAE. This stage was developed in previous works referenced in [15–17] and the contribution for the present work is in the next stage. In STAGE 2, the predictions made by the trained model for a particular test scenario are used as input for computing, through a battery electrical discharge model, the energy required to successfully complete the search task.

STAGE 1. Training of the exploration performance prediction model (nonlinear model): The process is initiated with data collection to obtain a database with which to carry out the supervised training of the neural network, as well as the validation and testing of the resulting prediction model. The dataset for training and validation of the optimized artificial neural network to predict the performance of the exploration algorithm Random Walk is available at [29] and for the exploration algorithm Random Walk_WSB is available at [30]. The characteristics of the dataset are described in Table 1.

STAGE 1. Training of the exploration performance prediction model (nonlinear model)

STAGE 2. Energy estimation model (linear model)

Figure 3. Prediction of the energy consumption of a robot in an exploration task using an optimized neural network.

Table 1. Characteristics of the datasets used for training, validation and testing.

RW and RW_WSB Dataset Characteristics	
4 Inputs (Experimental conditions)	-Number of rows/columns in the square grid -Number of obstacles -Target object X coordinate -Target object Y coordinate
1 Output (Exploration algorithm performance measure)	Maximum Number of steps (Max_NS)
Grid sizes	$3 \times 3, 4 \times 4, 5 \times 5, 6 \times 6, 7 \times 7, 8 \times 8$
Type of data	Integer
Type of variables	Quantitative
No. instances	475
Prediction model evaluation (metrics)	%RAE and R

After obtaining the database, the ANN is trained using the Bayesian regularization algorithm. During each iteration, the evaluation metrics, %RAE and R, are calculated. Subsequently, the stop conditions are evaluated. If these are not fulfilled then a new cycle executes the Tabu list optimization algorithm to propose new hyperparameters (number of layers and number of neurons per layer) to carry out a new training and validation cycle of the ANN. This iterative algorithm is completed when the stop conditions are met.

In the previous work [16], Tabu List algorithm is proposed to solve an optimization problem that consists of searching for the ANN topology that minimizes %RAE and, at the same time, maximizes the Pearson correlation coefficient (R) between the predicted data and the actual data in a number of experimental conditions. The Tabu list optimization algorithm calculates the number of layers and neurons per layer that maximizes the performance of the neural network for one or two layers and therefore, gets the best descriptive metrics of system behavior.

The Figure 4 shows a grid with the all possible structures of the neural networks. The number of neurons has been limited to an integer number in the range (1–20) and the number of layers has been limited to the integer range (1–2). According to [17] neural networks of 1 or 2 layers with few neurons generated predictors with good performance, for this reason, in this work the number of possible ANN layers has been limited to 1 or 2,

while the number of neurons has been limited to an integer number in the range (1–20). As a result, the solution space S for this optimization problem is bounded, which also increases the chances of finding the global optimum using Tabu list.

Figure 4. Search grid of the optimal neural network structure. Source: [16].

“To ensure that optimization algorithms include single and two-layer neural network structures, we vary the number of neurons in layer two between zero and twenty (0–20). Then, if a local optimum is found in the zero column of the search grid, it means that the structure of the neural network is single layer. Variations with respect to the rows of the zero column between one and twenty (1–20) determine the number of neurons in layer 1 when these structures are presented. For example, for the coordinate (20, 0) shown in Figure 4, a structure of 20 neurons in layer 1 and zero neurons in layer 2 (structure that does not have layer 2) is tested, obtaining a prediction model which is evaluated according to the metrics (% RAE and R) defined. Now, for the coordinate (20, 10) of the same figure a neural network structure of 20 neurons in layer 1 and 10 neurons in layer 2 is tested” [16].

According to the results in [16], Tabu List converges faster than Hill Climbing when used to optimize the hyperparameters of the ANN. This result was verified with the simulations performed for this work. Both, Hill Climbing and Tabu List, reach about the same optimal value for the target metric, but Tabu list does it faster as shown in Figure 5. In general, the comparison of the resulting evaluation metrics evidences that the proposed Tabu list optimization method definitely have a significant and positive impact on model performance.

By replicating the experiment in [16], the optimal structure obtained is presented in Table 2.

A %RAE of 2.55 and an R very close to 1 suggest a very reliable predictor for the RW exploration algorithm using a [19:19] structure for the ANN, that is, a first hidden layer with 19 neurons and a second hidden layer with 19 neurons as is shown in Figure 6.

For its part, for the case of the RW_WSB exploration algorithm, a % RAE of 3.31 was obtained with an R very close to 1, when the structure of the ANN is [18:20], that is, a first hidden layer with 18 neurons and a second hidden layer with 20 neurons as is shown in Figure 7.

Figure 5. Search with Hill Climbing Random Restart and Tabu List algorithms of the minimum %RAE for the optimal ANN structure for the RW (a) and RW_WSB (b) exploration algorithms.

Figure 6. Optimized neural network structures for predictor of the RW performance.

Figure 7. Optimized neural network structures for predictor of the RW_WSB performance.

Table 2. Results for prediction of the RW and RW_WSB performance.

Exploration Algorithms	%RAE	R	Structure ANN
RW	2.55	0.9898	[19:19]
RW_WSB	3.31	0.9881	[18:20]

To take advantage of the ability of ANNs to learn complex or non-linear relationships between variables, sigmoidal activation functions are used for the neurons of the hidden layers. An identity activation function is used in the output layer, as usually done in regression tasks.

STAGE 2. Energy estimation model (linear model): The ANN model is then used to predict the maximum number of steps (one step is a single displacement between adjacent intersections), Max_NS, required for the robot to complete the navigation task for a particular environment configuration. Max_NS is then used as input to a discharge battery model to obtain the desired energy budget.

To complete this task, we determined experimentally an electric discharge model for the robot's battery. With the battery fully charged, the robot moved across the grid describing a pentagonal trajectory. The battery voltage and stored charge measurements are taken every five seconds. This experiment was replicated 2 times (Experiment 1, replicate 1 and replicate 2). In a second experiment the robot moved across the grid on a four-step square path that was repeated until the battery was completely discharged. This square trajectory is considered the worst case power consumption scenario because at each grid intersection the robot has to make a 90 degree turn. In actual exploration tasks most paths do not require turns at every step. Battery voltage and stored charge measurements are taken at every intersection. This experiment was replicated 2 times (Experiment 2, replicate 1 and replicate 2). As shown in Figure 8, a very linear behavior was observed in the electrical discharge of the battery during its operation, therefore, it is to be expected to obtain an adjusted linear model that shows a correlation coefficient very close to one. The measured experimental data are referenced in [32,33].

The discharge behavior of the robot's battery, shown in Figure 8, fits very well a linear model:

$$\text{Electrical discharge [mAh]} = 2687.05 - 0.112086 \times \text{Time [s]} \quad (1)$$

with a Correlation Coefficient of -0.999978 and a R^2 of 99.9955% . Then, every 5 s there is a loss of charge of 0.56 mAh on the robot's battery.

The battery voltage remains very close to 15 volts with a deviation of approximately ± 1 volt in the initial and final part of the discharge curve as shown in Figure 9.

Figure 8. Behavior of electrical discharge variable in the robot's battery during its operation in experiment 1 and 2.

Figure 9. Behavior of voltage variable in the robot's battery during its operation.

Theoretically, at a constant speed (10 m/s) and a constant distance between the intersections of the grid (40 cm), the time spent by the robot taking a single step is 4 s. However, the time in which the robot performs angular movements to make direction changes in its path must be added. We took 3806 measurements of the battery's discharge while the robot was rotating and moving between intersections to compute the well-fitting linear model shown in Figure 10a,b, to describe the relationship between electrical discharge and number of steps taken by the robot. Equation (2) shows the equation that describes the fitted model:

$$\text{Electrical discharge [mAh]} = 2699.29 - 0.669792 \times \text{Number of steps} \quad (2)$$

Therefore, for every step that the robot advances it expends 0.669792 mAh of the battery charge.

For this linear regression model the correlation coefficient is equal to 0.999992, indicating a strong relationship between predicted and measured values. The R-squared statistic indicates that the model explains 99.9984% of the variability of the Electrical discharge [mAh]. Taking into account the previous results, it is observed that the relationship between the variables Max_NS and electric discharge is well described by a linear model, and that this proposed model can be used to estimate the maximum energy required by the robot to carry out an exploration task.

Figure 10. Model lineal graphs. (a) Graph of fitted model; (b) Graph of Predicted values Vs. Observed values.

4. Results and Discussion

Once we have the model described above, relating the number of steps taken by the robot and the electric discharge of its battery, we can answer the problem question: Is the current energy stored in the robot's battery enough for the robot to successfully complete the task of exploring and searching for an object? Our proposed model is applicable to deterministic and non-deterministic exploration algorithms.

For the present work we are considering non-deterministic algorithms, where each time they are executed the robot reaches its goal using a different number of steps. Therefore, there is no point in trying to predict exactly how much energy the robot will expend for a particular run, but rather the maximum amount of energy that the robot will need according to the expected worst performance of the exploration algorithm as determined by the ANN prediction model.

For our evaluation, 64 tests were performed with the iRobot Create 2, using the RW exploration algorithm with different experimental conditions and 64 tests using the RW_WSB algorithm, for a total of 128 tests. Knowing the energy stored in the robot's battery at the beginning of each test run, we predict if the robot will be able to successfully complete its exploration task. After completion of the run, we determined the accuracy of the model prediction.

As can be seen in Figure 11, in 96.88% of the cases the actual number of steps taken by the robot during experimentation did not exceed the predicted number of steps, which suggests a good prediction performance.

Figure 11. Comparison between real number of steps and predicted maximum number of steps for RW algorithm.

When the RW_WSB exploration algorithm is used, the percentage of tests where the predicted number of steps is higher than the actual number of steps (which is good since we are trying to predict the worst case) was 95.31%, which also suggests a very good prediction, as shown in Figure 12.

Figure 12. Comparison between real number of steps and predicted maximum number of steps for RW_WSB algorithm.

In terms of energy consumption, as shown in Figure 13, out of the 128 tests carried out, 96.09% of the time the amount of energy needed by the robot according to the model would have been enough for the robot to actually finish the task in practice, indicating that only in a small percentage of cases the energy-budget suggested by the model would fail short.

It should be noted that for the scenarios described in this work it is not possible to create an energy budget based on pre-computed paths to the target, since the location of such target is unknown to the robot at the beginning of the task. However, the proposed prediction model can calculate the maximum energy that the robot needs to reach each and every one of the possible destinations considering the particular exploration algorithm executed by the robot. An example for the case of a 6×6 grid is shown in Figure 14 where the highest estimated energy consumption is 263 mAh. The areas with the highest estimated energy consumption are the three remaining corners of the grid, considering that the robot always starts its exploration from the upper left corner.

Figure 13. Predicted energy consumption versus Real energy consumption.

Figure 14. Predicted energy consumption at each intersection for a 6×6 grid [mAh] and RW algorithm.

A final set of tests was carried out using 9 experimental scenarios not previously considered, as listed in Table 3. Each scenario in the list was executed 32 times for a total of 288 experimental tests (9 scenarios by 32 replicas per scenario).

Table 3. Conditions for a final experimental test.

Experimental Condition	Exploration Algorithm	Inputs (Experimental Conditions)			
		Grid Size	Obstacles	x Coordinate	y Coordinate
1	RW	4	0	0	2
2		5	0	1	2
3		6	0	1	2
4		6	0	1	3
5		3	0	2	2
6	RW_WSB	5	0	2	3
7		5	0	1	4
8		5	0	2	1
9		6	0	3	4

Due to the non-deterministic behavior of the exploration algorithms under test, each time an experimental scenario is executed different results are observed. However, as can be seen in Figure 15, there is a concentration of data that presents a higher frequency and probability of occurrence, as observed in the following graphs of frequencies and accumulated probabilities for each of the 9 experimental conditions.

Figure 15. Cont.

Figure 15. Frequency and cumulative probability for experiment 1, with nine experimental conditions and 32 replicates.

For the analysis below, we define the real electrical discharge as energy consumption value at which the robot reached the target object at least 94% of the time, as also indicated in the cumulative probability in Table 4.

Table 4. Predicted Electrical discharge versus real electrical discharge.

Experimental Condition	Predicted Max_NS	Predicted Electrical Discharge	Real Electrical Discharge	Cumulative Probability
1	102	68	42	97%
2	90	61	52	94%
3	104	69	67	94%
4	169	113	71	94%
5	32	21	17	94%
6	55	37	18	97%
7	121	81	45	100%
8	47	32	31	97%
9	96	65	64	97%

Max _NS is predicted using the optimized neural network model. The predicted Electrical discharge is calculated using the battery model that describe the relationship between Electrical discharge and Number of steps.

The real maximum energy used by the robot never exceeded the predicted value, which can be described as an excellent result, since the energy required for the robot to accomplish its task was enough 100% of the time as shown in Figure 16. The difference between the predicted energy and the real energy can be seen as a reserve energy, that on our experiments averaged 15 mAh. It is relevant that the predicted values are always higher than the real values (not necessarily equal) so that the safety of the robot's mission is not affected, which is the preponderant factor in this work.

Figure 16. Comparison of predicted electrical discharge versus observed electrical discharge.

The drawback of this method is the overestimation of the model, because a robot that could strictly perform an exploration task with its stored energy, the overestimation could reject it. However, this estimate has a positive impact on the useful life of the battery since in the execution of the exploration by the robot, the battery could reach its lowest charge levels, thus reducing its charge and discharge cycles. It could be as future work to review that the useful life of the robot's battery is improved with this proposed method.

5. Conclusions

This research work demonstrates that a Machine Learning system with Tabu List hyperparameter optimization can predict the energy consumption for robots considering the performance of the exploration algorithms. Earlier methods in the literature generally require explicit equations to model the dynamics and kinematics of the robot in order to estimate its energy consumption.

Many recent applications of Machine Learning focus on the use of big datasets for training [34], however, there are many problems where the available datasets are small. Our method based on Machine Learning techniques has generated excellent results for small databases in general.

A total of 128 tests were carried out using a robot executing two exploration algorithms with the objective of locating a target whose location is initially unknown. The experimental consumption energy was measured and compared with the prediction of our model. A success rate of 96.09% was obtained, suggesting an adequate energy predictor, which could be useful for energy budgeting in actual mobile robot applications.

To evaluate the performance of the predictor with the two selected, non-deterministic, navigation algorithms 9 experimental scenarios were run, with each one replicated 32 times. The results obtained show that the actual energy expended by the robot did not exceed the predicted energy with probability $\geq 94\%$.

For future work, the prediction of energy consumption will be applied for deterministic exploration algorithms such as those based on reinforcement learning.

Author Contributions: Conceptualization, L.C., W.P.; methodology, L.C., W.P.; software, L.C., W.P., Á.P., M.R.; collect and processing data, L.C., Á.P.; formal analysis, L.C., W.P.; validation and testing, L.C., W.P., Á.P., M.R.; writing—original draft preparation, L.C., W.P., Á.P., M.R.; writing—review and editing, L.C., W.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Canfield, S.L.; Hill, T.W.; Zuccaro, S.G. Prediction and Experimental Validation of Power Consumption of Skid-Steer Mobile Robots in Manufacturing Environments. *J. Intell. Robot. Syst.* **2018**, *94*, 825–839. [CrossRef]
2. Parasuraman, R.; Kershaw, K.; Pagala, P.; Ferre, M. Model Based On-Line Energy Prediction System for Semi-Autonomous Mobile Robots. In Proceedings of the 5th International Conference on Intelligent Systems, Modelling and Simulation—ISMS 2014, Langkwaki, Malaysia, 27–29 January 2014. [CrossRef]
3. Robotics Industry Insights—Industrial Mobile Robot S. Available online: https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Industrial-Mobile-Robot-Safety-Standards-on-the-Forefront/content_id/6710 (accessed on 27 October 2020).
4. Safety of Mobile Robot Systems in Industrial Applications. Available online: https://www.researchgate.net/publication/337339431_Safety_of_Mobile_Robot_Systems_in_Industrial_Applications (accessed on 27 October 2020).
5. Mei, Y.; Lu, Y.H.; Lee, C.S.G.; Hu, Y.C. Energy-Efficient Mobile Robot Exploration. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, Orlando, FL, USA, 15–19 May 2006. [CrossRef]
6. Tampubolon, M.; Pamungkas, L.; Chiu, H.-J.; Liu, Y.-C.; Hsieh, Y.-C. Dynamic Wireless Power Transfer for Logistic Robots. *Energies* **2018**, *11*, 527. [CrossRef]
7. Guo, T.; Guo, J.; Huang, B.; Peng, H. Power consumption of tracked and wheeled small mobile robots on deformable terrains—model and experimental validation. *Mech. Mach. Theory* **2019**, *133*, 347–364. [CrossRef]
8. Hou, L.; Zhang, L.; Kim, J. Energy Modeling and Power Measurement for Mobile Robots. *Energies* **2018**, *12*, 27. [CrossRef]
9. Bonani, M.; Longchamp, V.; Magnenat, S.; Rétornaz, P.; Burnier, D.; Roulet, G.; Vaussard, F.; Bleuler, H.; Mondada, F. The marXbot, a Miniature Mobile Robot Opening New Perspectives for the Collective-Robotic Research. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010. [CrossRef]
10. Mei, Y.; Lu, Y.H.; Hu, Y.C.; Lee, C.S.G. Deployment of mobile robots with energy and timing constraints. *IEEE Trans. Robot.* **2006**, *22*, 507–522. [CrossRef]
11. Liu, S.; Sun, D. Modeling and Experimental Study for Minimization of Energy Consumption of a Mobile Robot. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, Kaohsiung, Taiwan, 11–14 July 2012; pp. 708–713. [CrossRef]

12. Meike, D.; Pellicciari, M.; Berselli, G. Energy efficient use of multirobot production lines in the automotive industry: Detailed system modeling and optimization. *IEEE Trans. Autom. Sci. Eng.* **2014**, *3*, 798–809. [CrossRef]
13. Brossog, M.; Bornschlegl, M.; Franke, J. Reducing the energy consumption of industrial robots in manufacturing systems. *Int. J. Adv. Manuf. Technol.* **2015**, *78*, 1315–1328. [CrossRef]
14. Xu, W.; Liu, H.; Liu, J.; Zhou, Z.; Pham, D.T. A practical energy modeling method for industrial robots in manufacturing. *Comput. Vis.* **2017**. [CrossRef]
15. AI Accurately Predicts Useful Life of Batteries | Stanford News. 2019. Available online: <https://news.stanford.edu/2019/03/25/ai-accurately-predicts-useful-life-batteries/> (accessed on 23 February 2021).
16. Caballero, L.; Jojoa, M.; Percybrooks, W.S. Optimized neural networks in industrial data analysis. *SN Appl. Sci.* **2020**, *2*, 1–8. [CrossRef]
17. Caballero, L.; Benavides, C.; Percybrooks, W. Machine Learning-Based System to Estimate the Performance of Exploration Algorithms for Robots in a Bi-Dimensional Grid. In Proceedings of the 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC), Cartagena, Colombia, 18–20 October 2018; pp. 1–6. [CrossRef]
18. Caballero, L.; Jojoa, M.; Percybrooks, W. *Optimized Artificial Neural Network System to Select an Exploration Algorithm for Robots on Bi-dimensional Grids*; Springer: Cham, Switzerland, 2018; pp. 16–27.
19. Rodó, D.M.; Bakx, G.E.; Iglésias, B.E. *Inteligencia Artificial Avanzada*; Editorial UOC—Editorial de la Universitat Oberta de Catalunya: Barcelona, Spain, 2013.
20. al Nsour, H.; Alweshah, M.; Hammouri, A.I.; al Ofeishat, H.; Mirjalili, S. A Hybrid Grey Wolf Optimiser Algorithm for Solving Time Series Classification Problems. *J. Intell. Syst.* **2020**, *29*, 846–857. [CrossRef]
21. Precup, R.E.; David, R.C.; Petriu, E.M.; Szedlak-Stinean, A.I.; Bojan-Dragos, C.A. Grey Wolf Optimizer-Based Approach to the Tuning of Pi-Fuzzy Controllers with a Reduced Process Parametric Sensitivity. *IFAC PapersOnLine* **2016**, *49*, 55–60. [CrossRef]
22. Roman, R.C.; Precup, R.E.; David, R.C. Second Order Intelligent Proportional-Integral Fuzzy Control of Twin Rotor Aerodynamic Systems. *Procedia Comput. Sci.* **2018**, *139*, 372–380. [CrossRef]
23. Abed-Alguni, B.H. Island-based Cuckoo Search with Highly Disruptive Polynomial Mutation. *Int. J. Artif. Intell.* **2019**, *17*, 57–82.
24. iRobot® Create® 2 Open Interface (OI) Specification Based on the iRobot® Roomba® 600. 2018. Available online: www.irobot.com (accessed on 24 October 2020).
25. Biomedical Signal Processing and Artificial Intelligence Laboratory (BSPAI) Research Group GitHub. Alvaro850/Predicting-Energy-Consumption: Predicting the Energy Consumption of a Robot in an Exploration Task Using Optimized Neural Networks. Available online: <https://github.com/Alvaro850/Predicting-energy-consumption> (accessed on 18 February 2021).
26. Python Software Foundation. pyCreate2, PyPI. 2020. Available online: <https://pypi.org/project/pycreate2/> (accessed on 1 November 2020).
27. Jojoa, M.; Caballero, L.; Percybrooks, W. Lieslec/Optimized-Neural-Networks. Available online: <https://github.com/lieslec/Optimized-neural-networks> (accessed on 22 February 2021).
28. Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *Neural Netw.* **2014**, *61*, 85–117. [CrossRef] [PubMed]
29. Biomedical Signal Processing and Artificial Intelligence Laboratory (BSPAI) Research Group. Random_Walk_Dataset.xlsx. Available online: https://www.dropbox.com/scl/fi/5asy6k8vu7izwfn3lxzo9/Random_Walk_dataset.xlsx?dl=0&rlkey=ifzuwf4i6j8oae38nlh0768ub (accessed on 18 February 2021).
30. Biomedical Signal Processing and Artificial Intelligence Laboratory (BSPAI) Research Group. Random_Walk_WSB_Dataset.xlsx. Available online: https://www.dropbox.com/scl/fi/567vmblo32o4tt5wgd1ar/Random_Walk_WSB_dataset.xlsx?dl=0&rlkey=xn0lkhhrygbmmx1mqtotdbrf8 (accessed on 18 February 2021).
31. Glover, F. *Tabu Search and Adaptive Memory Programming—Advances, Applications and Challenges*; Springer: Boston, MA, USA, 1997; pp. 1–75.
32. Caballero, L.; Perafán, Á.; Rinaldy, M.; Percybrooks, W. Experiment_1.xlsx. 2020. Available online: https://www.dropbox.com/s/5hi6ndwi2fslqk5/Experiment_1.xlsx?dl=0 (accessed on 24 February 2021).
33. Caballero, L.; Perafán, Á.; Rinaldy, M.; Percybrooks, W. Experiment_2.xlsx. September 2020. Available online: https://www.dropbox.com/s/mwhrpertia6xw74/Experiment_2.xlsx?dl=0 (accessed on 24 February 2021).
34. Qiu, J.; Wu, Q.; Ding, G.; Xu, Y.; Feng, S. A Survey of Machine Learning for Big Data Processing. *EURASIP J. Adv. Signal Process.* **2016**, *2016*, 67. [CrossRef]