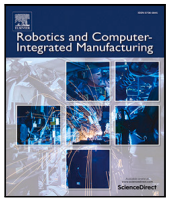
Robotics and Computer–Integrated Manufacturing 85 (2024) 102629

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



Full length article

BN-LSTM-based energy consumption modeling approach for an industrial robot manipulator

Hsien-I Lin [a,*], Raja Mandal [b], Fauzy Satrio Wibowo [b]

[a] Institute of Electrical and Control Engineering, National Yang Ming Chiao Tung University, 1001 University Road, Hsinchu, 30010, Taiwan
[b] Graduate Institute of Automation Technology, National Taipei University of Technology, 1, Sec. 3, Chung-Hsiao E. Road, Taipei, 10608, Taiwan

ARTICLE INFO

Keywords:
Industrial robot (IR)
Energy consumption (EC)
BN-LSTM
Machine learning (ML)
Deep learning (DL)

ABSTRACT

Industrial robots (IRs) are widely used to increase productivity and efficiency in manufacturing industries. Therefore, it is critical to reduce the energy consumption of IRs to maximize their use in polishing, assembly, welding, and handling tasks. This study adopted a data-driven modeling approach using a batch-normalized long short-term memory (BN-LSTM) network to construct a robust energy-consumption prediction model for IRs. The adopted method applies batch normalization (BN) to the input-to-hidden transition to allow faster convergence of the model. We compared the prediction accuracy with that of the 1D-ResNet14 model in a UR (UR3e and UR10e) public database. The adopted model achieved a root mean square (RMS) error of 2.82 W compared with the error of 6.52 W achieved by 1D-ResNet14 model prediction, indicating a performance improvement of 56.74%. We also compared the prediction accuracy over the UR3e dataset using machine learning and deep learning models, such as regression trees, linear regression, ensemble trees, support vector regression, multilayer perceptron, and convolutional neural network-gated recurrent unit. Furthermore, the layers of the well-trained UR3e power model were transferred to the UR10e cobot to construct a rapid power model with 80% reduced UR10e datasets. This transfer learning approach showed an RMS error of 3.67 W, outperforming the 1D-ResNet14 model (RMS error: 4.78 W). Finally, the BN-LSTM model was validated using unseen test datasets from the Yaskawa polishing motion task, with an average prediction accuracy of 99%.

1. Introduction

Industrial robots (IRs) are pivotal to the automation industry in this new era of manufacturing, which has led to a significant increase in the output. Manufacturers can lower production costs, increase production speed, and enhance product quality using IRs [1,2]. However, environmental issues have recently become a priority for sustainable manufacturing. Manufacturers have started to focus on energy consumption (EC) because the world's energy usage has increased over the last few decades. European Union policy aims to reduce EC by up to 30% by 2030, and the rising awareness of energy prices encourages reducing consumption [3]. Therefore, researchers have proposed several approaches to reduce the EC of IRs in modern industries. These strategies include energy-saving methods, that is, mathematical, experimental, or those based on simulation modeling and deep learning approaches.

EC modeling for an IR system is not easy because IRs have several components, such as controllers, drivers, cooling fans, and servo motors. These factors lead to considerably different EC characteristics. In [4–6], researchers explored and analyzed IR energy flow to enhance the modeling accuracy of the motor core, windage–friction, stator–rotor, and stray load losses. Dynamic parameter identification is required for a direct modeling approach to obtain the parameters of individual joints and servomotors. Therefore, the modeling process is time-consuming and complex. The identification characteristics of each IR depend on the manufacturer, making them challenging to handle and execute in real-world applications.

Recently, researchers [7–10] have focused on data-driven techniques, applying intelligent algorithms to develop more profound prediction models from the complex, nonlinear relationship between operational parameters and power consumption. The data-driven modeling solution eliminates this complicated procedure by developing advanced algorithm models to estimate the actual power and EC of an IR manipulator. However, there is still scope for enhancing the accuracy of the prediction model using a less complex model architecture. [10] proposed EC feedback models for UR cobots based on three processes: motion planning, dynamic modeling, and EC modeling. [9] proposed the 1D-ResNet14 model using a similar UR dataset because the accuracy test on the unseen test dataset was not completed in [10]. According

* Corresponding author.
E-mail address: sofin@nycu.edu.tw (H.-I. Lin).

https://doi.org/10.1016/j.rcim.2023.102629
Received 24 May 2023; Received in revised form 11 July 2023; Accepted 8 August 2023
Available online 22 August 2023

to the research findings, their proposed IR power modeling method accurately constructed the power model and achieved 21.83% smaller validation error. Consequently, our study aimed to identify the best data-driven power model that outperforms other suggested deep learning models [9,10] with a straightforward design architecture. In this study, we adopt a model that can perform better on unseen test datasets with an RMS error improvement of 56.74%. Consequently, the adopted model predicts the unseen target data with an average accuracy of 99%.

This study adopted a batch-normalized long short-term memory (BN-LSTM)-based data-driven method to predict the power consumption of IR manipulators. The model was verified using the UR3e and UR10e public datasets. Moreover, we compared the performance of our prediction model with those of two previous studies [9,10]. Additionally, we performed extensive experiments on several machine learning and deep learning models, such as regression trees (RTs), linear regression (LR), ensemble trees (ETs), support vector regression (SVR), multilayer perceptron (MLP), and convolutional neural network-gated recurrent unit (CNN-GRU). Finally, we adopted the BN-LSTM model to build a reliable EC model. A comprehensive comparison is presented. Then, an unseen dataset testing experiment was conducted to test the ability of the adopted model to generalize the data collected during the polishing motion task. The main contributions of this study are as follows:

1. The BN-LSTM model adopts a strategy called the input-to-hidden transition, which allows for faster convergence of the model. The model outperformed the models proposed in two previous studies [9,10].
2. The adopted BN-LSTM model achieved outstanding performance compared with two previous networks [9,10]. It achieved RMS error improvements of 56.74% and 23.22% on the UR3e and UR10e datasets, respectively.
3. We also performed extensive experiments with several traditional machine learning and deep learning models using the UR3e cobot dataset to measure the prediction accuracy. The models were LR, RTs, ETs, SVR, feedforward neural network, and CNN-GRU models. The adopted BN-LSTM model outperformed these other models.
4. We introduced the Yaskawa GP7 dataset, which contains 64,600 samples with 18 features, for polishing motion. This provides a less complex data format for researchers to test their prediction models.

The adopted BN-LSTM network prevents overfitting problems by using batch normalization (BN) in the input-to-hidden transition. This improved the learning and training speed of the model by using a slightly higher learning rate (0.1–0.2). The adopted model performs consistently with impressive accuracy in predicting sequential data, regardless of the complexity of the datasets. Finally, the difficulty of building a nonlinear power prediction model can be significantly reduced by using the adopted approach.

The remainder of this paper is structured as follows: Section 2 provides a review of related research studies. Section 3 adopts the deep learning model based on the BN-LSTM network. Section 4 reports the results and discussion. Finally, Section 5 provides a conclusion and describes future work.

2. Review of related research

Researchers have developed four primary methods to reduce the energy usage of IRs. They are as follows: (A) point-to-point (PtP) and multi-point trajectory optimization, (B) simulation tools in EC research, (C) the optimization of IR process sequence, and (D) data-driven optimization method using an intelligent model [5–10]. The related studies that mainly highlighted each of the four aforementioned methods are listed below.

2.1. Point-to-point and multi-point trajectory optimization methods

2.1.1. Point-to-point trajectory optimization

The PtP trajectory optimization technique, according to [5], is classified into inverse and direct approaches. The inverse approach [11–16] used a mathematical model to formulate mechatronic and robotic systems, which are frequently used to decrease manual computing work. The trajectory, which is frequently represented as a polynomial function, is then incorporated while accounting for the EC and is optimized. In general, the inverse approach is less complex than the direct approach in terms of the computational effort required to obtain the optimal trajectory. Linear behavior is typically modeled using only the kinematic and dynamic properties of the mechanical system (such as the length, mass, and inertia of links, gear ratios, and friction coefficients), and electrical parameters (such as the motor winding resistance, motor torque, and voltage constants).

In contrast, direct approaches [5,6] are more dependent on models of mechatronic and robotic systems, and consider the associated losses, such as motor, motor drive, and auxiliary losses. Here, the model is used to examine various motion-planning approaches and the influence of different parameters on EC.

To find the best trajectory, a variety of optimization algorithms are applied. These are some examples:

- Genetic algorithms [12,13,17]
- Dynamic programming [14]
- Gradient-based optimization algorithms [15]
- Discrete dynamic programming [16]

[5] demonstrated that, within the feasible domain, it is possible to achieve energy savings of more than 30% of the maximum consumption within a variable cycle time. [18]reported that 3%–8% EC can be saved in IR motor drives. [19] showed an improvement of 16% over a traditional system. [20] suggested a modified parabolic velocity profile that allows savings of up to 12.5% in each repetition of the PtP motion and up to 33% of input energy under the same peak acceleration.

2.1.2. Multi-point trajectory optimization

According to [21], a pick-and-place action can save up to 4% more energy than the fastest ABB RobotStudio trajectory. It uses a multistep trajectory optimization strategy to increase energy usage while decreasing the cycle time. Accordingly, the cycle time results are reduced by 3%. [22] investigated three different types of operations: up to 30% of energy can be saved from a linear trajectory, up to 70% of energy can be saved from a top-to-bottom linear trajectory, and up to 42% of energy can be saved from a circular trajectory.

The multi-point trajectory optimization method enables EC optimization without negatively impacting the cycle time. In some circumstances, the cycle time may be shortened. It is difficult to compare the various techniques in terms of energy savings. Indeed, the amount of energy savings is considerably dependent on numerous factors, such as the system type and task under consideration.

2.2. Simulation tools in energy consumption research

Researchers are increasingly using model-driven software solutions to optimize EC. Simulation software solutions provided by IR manufacturers include ABB RobotStudio® and KUKA KUKASim®. These simulation tools are used to measure the virtual motor power, where the total motor power signal demonstrates the power consumed by the mechanical robot arm and not the power fed into the controller cabinet from the power network. The power used by the controller cabinet was excluded. On the other hand, researchers frequently use mathematical or robotic simulation tools, such as MATLAB® and Delmia®, along with packages and toolboxes to aid robot simulation. The dynamic modeling

approach can accurately replicate the real robotic EC system as compared with ABB and KUKA, once the critical parameters (mechanical and electrical) are known.

[23] implemented a preliminary dynamic module structure of an ABB IRB140 robot in MATLAB to create a dynamic EC prediction model by considering joint torque and joint acceleration. Based on the suggested dynamic model and outcomes of the RobotStudio simulation, they provided EC data under various operating conditions. The estimated accuracy was poor because they failed to consider associated losses and actuator dynamics. Therefore, in subsequent studies, researchers have focused on enhancing the validation accuracy of IR dynamic simulation models to predict EC. [24] provided a new simulation tool for interfacing with existing robots. This tool uses offline programming tools in the industry to compute energy-optimal operating parameters automatically and decrease robot EC while maintaining productivity and manufacturing quality. The resulting energy-efficient approach is then transformed into specific robot commands that an IR can execute. This systematic approach was validated by conducting a typical pick-and-place operation on an ABB IRB1600 IR. The simulation results demonstrate considerable time and energy savings when compared with most trajectories produced using the ABB RobotStudio tool. The simulation findings showed a significant time and energy improvement up to 5% [21].

Finally, we can conclude that the total motor power and total motor energy signals were used to estimate the amount of power and energy used by the robots. These signals can be used to identify power usage peaks in virtual robots, allowing the robot programmer to modify the robot program to reduce power consumption. In the case of real robots, the signals can be considered to compare the power consumption of different IRs performing the same robot program and to check if any robot differs considerably from the others. The robot may require maintenance if there is any deviation from this nature.

2.3. Optimizing industrial robot process sequence

Scheduling the IR operation is another technique for optimizing the robot process sequence that leads to energy optimization. This method can be initiated to optimize EC by reducing the operating and idle times and optimizing the IR operation sequence of the subtask [25,26]. This approach is easy and efficient for adaptation to an IR operator when other approaches are not feasible. The author of [27] claimed that using this method, the estimated yearly energy-saving potential of IR is approximately 31% in the automation industry.

In [28,29], the authors suggested the main goal of the sequence-planning approach to minimize the EC of individual and interacting robot manipulators at a workstation while following the actual trajectories and overall cycle duration. This optimization approach does not depend on dynamic modeling; therefore, it does not require system identification or confidential IR data. Rather, it is based on the preliminary sampling of IR trajectories (i.e., displacement, velocity, and acceleration) and torque carried out throughout the execution of a movement in real time. From the logged data, the user can manually optimize the motion trajectory by eliminating idle and waiting times.

2.4. Data-driven based optimization method using intelligent model

Recently, deep-learning-powered data-driven modeling methods have gained popularity for constructing EC models for IRs owing to their simple modeling approach to avoid complicated dynamic modeling processes. Intelligent algorithms are useful for exploring deeper relationships between IR operating parameters and power consumption to build robust power consumption feedback models. [7] proposed a data-driven technique for realistic EC optimization. Researchers have used a back-propagation neural network (BPNN) to identify the complex relationships between operational parameters and energy utilization accurately. The proposed method employs a genetic algorithm to optimize the variable operating parameters to optimize EC. Experiments were planned and performed with the assistance of an Epson C4 6-DOF IR. The data show that the proposed approach is advantageous for operational parameter selection and energy savings. Researchers have observed that the data-driven optimization technique has various limitations regarding the prediction scope, modeling accuracy, and necessary sample size. [8] provided a transfer-learning-based EC modeling approach to construct a rapid EC feedback model for any IR manipulator. [29] presented a novel hybrid strategy for identifying the kinematic parameters of IRs with an improved convergence response by combining BPNN and particle swarm optimization (PSO) algorithms. According to the results, the suggested parameter identification method based on the BPNN and PSO requires fewer iterations and has a faster convergence time than the standard PSO algorithm.

[10] proposed three steps for modeling methodology processes: motion planning, dynamic modeling, and EC modeling. They gathered approximately 55,000 data points per instance using various cobot models (UR3e and UR10e) and then trained the proposed model to determine the unknown parameters of the model. However, they did not perform validation and reliability assessment tests of the EC prediction model. [9] proposed a new 1D-ResNet14 model to build a power estimation model using the same UR dataset to avoid dynamic modeling complexity, as introduced in [10]. The deep learning approach generates an exact nonlinear relationship between the operational parameters and power of the IR manipulator. Then, using a transfer learning strategy, some layers of the well-trained model were transferred to the Siasun SR10C IR manipulator to construct a rapid power model with a smaller number of IR datasets. The proposed IR power modeling technique produced a power model with a 21.83% smaller error in the validation dataset compared with [10].

Thus, several methods have been suggested to build a reliable power-optimized model, such as a direct-inverse dynamic mathematical model, simulation model, machine learning model, and deep learning model. Owing to the complex relationship between the operating parameters and EC, there are no commercially available products in the market that can automatically optimize the EC before the manipulator executes an unknown task. In this study, we also attempted to build a reliable and comprehensive energy model using a deep-learning-powered data-driven method that can predict EC with a remarkable degree of accuracy compared with other existing approaches.

3. Adopted energy consumption prediction model

The EC of an IR is highly dependent on its operating conditions. Therefore, we considered this aspect in the EC modeling approaches. Using this method, the adopted model can accurately estimate the power flow and EC of an IR manipulator. In this section, we emphasize the following steps, which are illustrated in Fig. 1.

3.1. Electrical energy consumption calculation

The total EC of IR can be represented as the integral of power flow in the time dimension. This can be computed as follows:

$$E = \int_{t_i}^{t_f} W.dt \tag{1}$$

where E represents the energy consumption across the $[t_i, t_f]$ period, t_i and t_f represent the beginning and end times of the instruction, respectively, and W represents the total power flow of the IR. The AC power formula can be expressed as:

$$W = V_{rms}.I_{rms}.PF. \tag{2}$$

The AC power formula is based on the RMS (root mean squared) values of the voltage (V) and current (I) and power factor(PF).

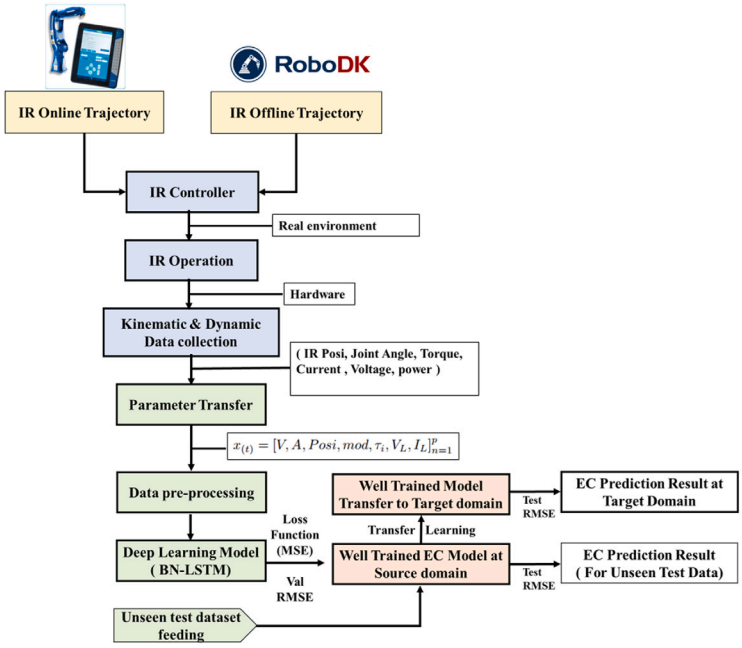


Fig. 1. Overview of the research architecture.

3.2. Input parameters preparation for the adopted BN-LSTM model

Owing to the complex nonlinear relationship between the operating parameters of IR and the EC, constructing an accurate EC model is challenging. The experiment revealed that the EC varies significantly for non-continuous PtP motion because the speed of the end effector changes quickly. To address this challenge, we considered specific kinematic and dynamic parameters when training the model to improve the accuracy of our EC prediction model. We considered 18 input features of the polishing dataset from Yaskawa GP7 IR, which comprises velocity (V), acceleration (A), payload, mod (MOVL, MOVJ, and MOVC), posi, joint torque (τ), robot current (I_L), and voltage (V_L), to train the network, whereas the UR datasets comprised 30 input features. The proposed dataset from the experiment on Yaskawa GP7 shows greater performance in building power and EC models with less complexity than the dataset proposed in [9,10].

The input sequence to the BN-LSTM network can be represented as:

$$x_{(t)} = [V, A, W, Posi, mod, \tau_i, V_L, I_L]_{n=1}^{p} \tag{3}$$

where V is the velocity, A is the acceleration, W is the payload, the robot position is $Posi$ (6 axis robot), mod is known as the move type, the robot joint torque is (τ_i), the supply voltage is (V_L), and the line current is (I_L). The complete dataset on the source domain robot can be represented as:

$$D_s = [x_{(t)}, E]_{n=1}^{p} \tag{4}$$

where D_s is defined as the source domain dataset, x_t represents the input parameter set, and E is the EC.

3.3. Data normalization

To deal with the data noise issue of the large IR dataset, we processed the input feature data to build an accurate data-driven EC prediction model. In this experiment, we used a long short-term memory (LSTM) recurrent neural network architecture with BN for the input-to-hidden states of each LSTM cell to enhance the training speed and accuracy of the model by recentering and rescaling.

The ranges of the values for the various IR operating parameters differ significantly; they are not equivalent. First, the BN layer normalizes the activation function of each channel by taking the mean of the minibatches and dividing it by the standard deviation of the minibatch. This makes the training more effective. Second, the learnable offset β of the BN layer moves the input and scales the learnable scale factor γ, known as learnable parameters, which are updated during network training.

The BN operation aids in the normalization of the input elements x_t by first calculating the mean μ_β and variance σ_β^2 over the spatial, time, and observation dimensions separately for each channel. The standard batch-normalized transformation can be defined as follows [30]:

$$\hat{x_{(t)}} = \frac{x_{(t)} - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}} \tag{5}$$

$$BN(x_{(t)}) = \beta + \gamma \odot \hat{x_{(t)}} \tag{6}$$

where ϵ is a regularization hyperparameter for smoothing, which prevents σ_β^2 from being zero.

3.4. Adopted deep learning model structure

The adopted LSTM model for sequence-to-sequence regression using BN is used to improve the training speed and accuracy of the model by recentering and rescaling. The implemented BN operation for the input-to-hidden state is highlighted with a red rectangle in Fig. 2.

The BN-LSTM network has an input sequence $x_{(t)}$ of feature data. $h_{(t-1)}$ and $c_{(t-1)}$ are the inputs from the previous time-step of the LSTM cell. $c_{(t)}$ and $h_{(t)}$ are the long-term and short-term states, respectively [30]. The mathematical formulas of the BN-LSTM cell can be represented as follows:

$$i_{(t)} = BN(W_{xi} \cdot x_{(t)}) + BN(W_{hi} \cdot h_{(t-1)}) + b_i \tag{7}$$

$$f_{(t)} = BN(W_{xf} \cdot x_{(t)}) + BN(W_{hf} \cdot h_{(t-1)}) + b_f \tag{8}$$

$$o_{(t)} = BN(W_{xo} \cdot x_{(t)}) + BN(W_{ho} \cdot h_{(t-1)}) + b_o \tag{9}$$

$$g_{(t)} = BN(W_{xg} \cdot x_{(t)}) + BN(W_{hg} \cdot h_{(t-1)}) + b_g \tag{10}$$

$$c_{(t)} = \sigma(f_{(t)}) \odot c_{(t-1)} + \sigma(i_t) \odot tanh(g_{(t)}) \tag{11}$$

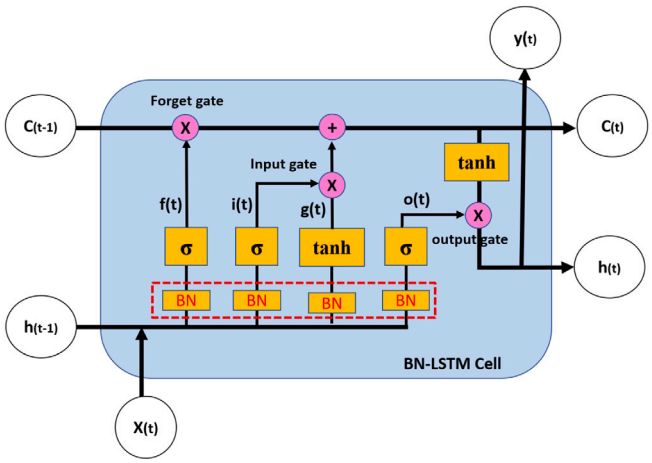


Fig. 2. BN-LSTM cell: The applied BN to the input-to-hidden state is highlighted with a red rectangle box.

$$y_{(t)} = h_{(t)} = \sigma(o_t) \odot tanh(c_{(t)}) \tag{12}$$

where $f_{(t)}$, $i_{(t)}$, $o_{(t)}$, and $g_{(t)}$ represent the forget, input, output, and candidate state gates, respectively. W_{xi}, W_{xf}, W_{xo}, and W_{xg} are the weight matrices of each of the four layers for their connection to input vector $x_{(t-1)}$. W_{hi}, W_{hf}, W_{ho}, and W_{hg} are the weight matrices of each of the four layers for their connection to the previous short-term state $h_{(t)}$. b_i, b_f, b_o, and b_g are the bias terms for each of the four layers. σ refers to the standard deviation of the batch normalization operation applied to the LSTM (Long Short-Term Memory) layer.

4. Result and discussion

In this section, we have designed two stages of experiments to investigate our approaches. Firstly, we investigate the dynamic input constraints that might affect the energy consumption of the robot. It covers different speeds, operating modes, acceleration, and payload. Then, we also discussed the performance of our adopted EC model based on BN-LSTM and compared it with previous models [9,10]. The remainder of this section will deal with the verification of the effectiveness of the adopted method using the UR dataset, as in [9,10]. Finally, the adopted BN-LSTM model is validated using the unseen Yaskawa polishing motion dataset.

4.1. Experimental setup and data collection

The experiment was conducted on a 6-DoF IR system (Yaskawa GP7 IR with a maximum payload capacity of 7 kg and a motion range of 927 mm) to validate the adopted methodologies. The experimental setup is shown in Fig. 3, where an external three-phase energy meter and a current transformer are connected to the main supply line of the robot controller to supply power. In this study, we measured the total current flow to the robot, instead of measuring each servo motor current. Most IR manufacturers do not provide facilities to access each joint current. Therefore, we preferred to measure the total current (I_L) flow to avoid the complexity of computing the electrical losses involved in the robotic operation, such as iron losses, windage, frictional losses, and stator and rotor losses. The additional electrical AC parameters of the IR system, such as voltage, power factor, and active power flow data, were measured using a three-phase smart energy meter (Acrel ADL400). We developed computational equations for the EC and overall carbon emissions in Python programming because the meter has the limitation of measuring a small amount of energy data for PtP motion.

Table 1
Table of operating param.

Operating mode	Range value
Payload	0–4 kg
MOVL	0–1500 mm/s
MOVJ	0–100%
MOVC	0–1500 mm/s
Acceleration	0–100%

The electrical data acquisition used the RS485 serial communication protocol. The kinematic and dynamic data of the Yaskawa controller were gathered using socket programming in Python. Table 1 shows the experimental stage investigated the influence of the operating parameters, such as speed ($Vmax$), acceleration ($Amax$), movement mode namely Linear Motion (MOVL), Joint Movement (MOVJ), and Circular Movement (MOVC). The variation of payload has been included to understand the energy consumption characteristic.

4.2. Effects of dynamic input constraints on the power consumption

In this first experiment, we only investigate dynamic input's effects on power consumption. The robot's motion is mainly Point-to-Point (PtP), as we want to understand its characteristics. Fig. 4 shows the illustration of experiments. This experiment mainly operates MOVL and MOVJ as robot movement as it represent Point-to-Point (PtP) in the pick-and-place application. The payloads of the experimental robot ranged from 0 to 4 kg at different robot speeds: 25%, 50%, 75% and 100% of the maximum speed. In these situations, we examine how the operating settings of IRs affect the amount of power they use and how they change over time. Based on the experimental investigation, the following strategies can be enforced to improve the cost-effectiveness of IRs in the production system:

- The operating speeds of the robot manipulators significantly affect the power flow and EC of the IR, as shown in Fig. 5. Thus, operating IRs at medium speeds is suggested based on the type of robot movement.
- The IR joint acceleration has a notable effect on the EC, as depicted in Fig. 6. In most cases, the relationship is linear with EC, but it also depends on the type of joint movement. The user must know the optimal joint acceleration for a particular set of instructions.
- Smoothing of the IR motion can be practiced to reduce EC. IR motor-drive systems require greater power to accelerate while commencing discontinuous robot motions.
- The payload significantly affects the IR power requirements by changing the axis torque. Consequently, the EC of the IR is linear with respect to that of the robot. Adding 1 kg of payload increases the energy by approximately 4%. Therefore, it is advisable to reduce the weight of the tools that the IR uses to decrease the EC of IRs. This technique can be implemented using lightweight materials for tooling system parts. In Fig. 7, the impact of the payload on the EC of the robot at 75% of its operating speed is demonstrated.
- We also observed that altering the payload at the end effector affects the torque at the joints of the manipulator. At 75% of the speed test, it has been observed that the torque of the active joints significantly affects power consumption. To save energy, it is advised that the operator must pay attention to the velocity, acceleration, and jerk when dealing with heavy task handling operations.

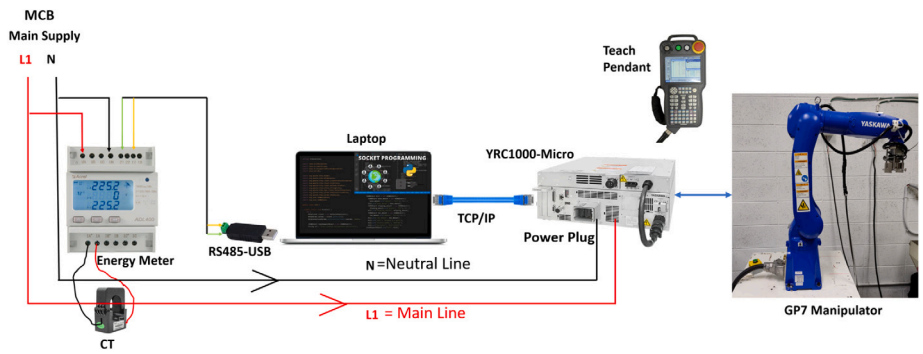


Fig. 3. Experimental setup and Data collection.

Table 2
Experimental table on EC for MOVJ and MOVL PtP instruction.

Mod	Dis (mm)	Vmax%	Amax%	Exe. time (s)	EC (Wh)	EC saved by MOVJ (%)
MOVJ	1365.7	25	25	2.74	0.447	46.77
MOVL	1365.7	25	25	4.71	0.839	
MOVJ	1365.7	50	50	1.12	0.370	29.64
MOVL	1365.7	50	50	2.50	0.526	
MOVJ	1365.7	75	75	0.76	0.362	31.242
MOVL	1365.7	75	75	1.72	0.526	
MOVJ	1365.7	100	100	1.31	0.422	5.115
MOVL	1365.7	100	100	1.42	0.445	

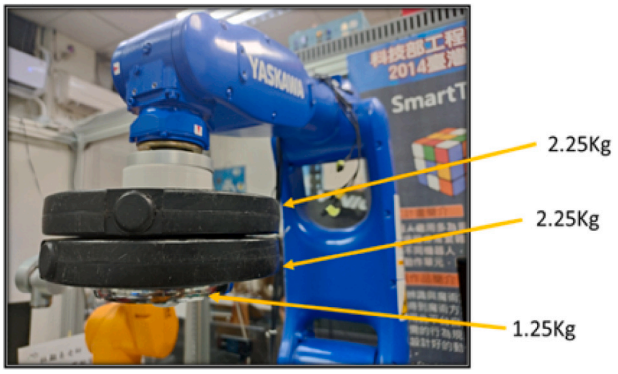


Fig. 4. Experiment with dynamic input constraint - Payload.

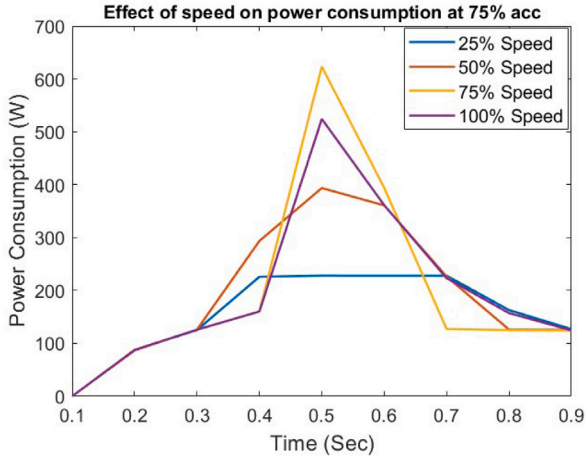


Fig. 5. Effect of variable speed on power consumption at 75% robot acceleration.

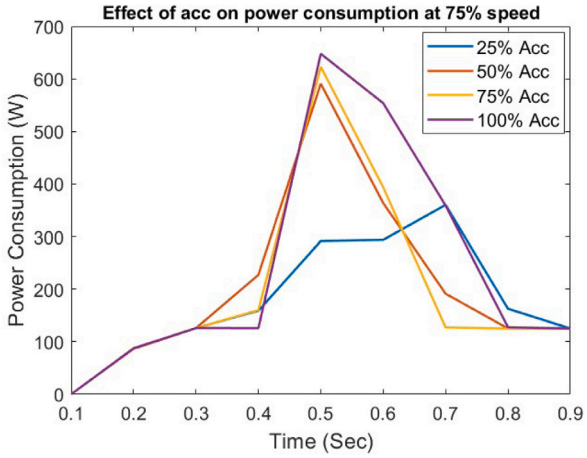


Fig. 6. Effect of variable acceleration on power consumption at 75% robot speed.

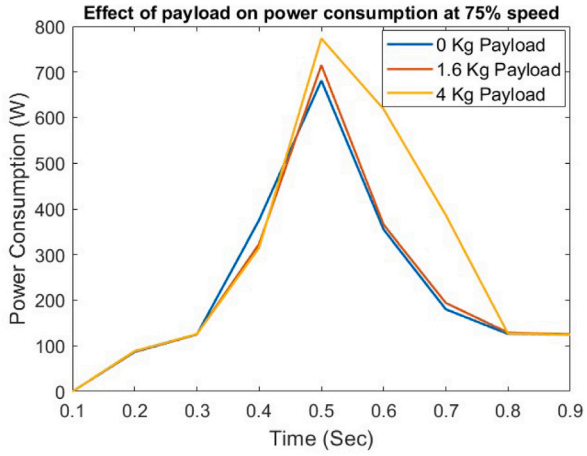


Fig. 7. Effect of payload on power consumption at 75% speed.

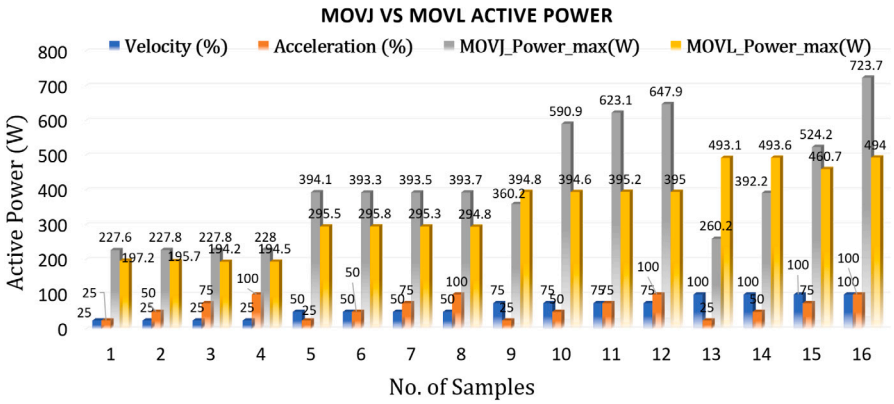


Fig. 8. Max rated active power for two distinct move types in a PtP instruction.

- Multiple PtP motion trajectories were performed by considering two modes: MOVJ and MOVL. The linear motion is smoother and has a lower max-rated active power than the joint motion at the same speed and acceleration, as shown in Fig. 8. However, the MOVJ instruction of an IR is even faster than the MOVL; therefore, there is less execution time for traveling the same distance at the same speed % and acceleration %. In Table 2, the experimental data table shows that MOVJ PtP instruction can save up to 46.77% EC at a lower speed than MOVL and up to 5.1% EC at a higher speed for GP7 IR.

4.3. UR3e and UR10e cobots dataset representation

In [9,10], scholars considered five sets (5 × 6) of input operating parameters to construct the IR power model, such as joint position q, joint speed $\dot{q}$, joint motor current I_q, end-effector position, and joint torque τ. In this study, we used a similar UR dataset to compare the effectiveness of the adopted power model with that of the method proposed in [9]. Table 3 presents the datasets for both UR3e and UR10e cobots. The dataset had three different payload conditions, including no-payload conditions, and the power model was trained using the no-payload condition dataset.

4.4. Yaskawa polishing motion dataset

The dataset contains information on the kinematics and dynamics of Yaskawa GP7 for the data-driven modeling of IR power and EC. The dataset includes the speed, acceleration, payload, type of robot movement (MOVL, MOVJ, MOVC), end-effector position, joint torque, voltage, and current of the robot. The angular velocity ($\dot{q}_i$), angular acceleration ($\ddot{q}_i$), and distance traveled (mm) values were obtained from the measured data and three different payload conditions (including no-load conditions) were considered. However, a no-payload condition dataset was used to train the power and EC models. We collected 64,600 sample datasets (a data sampling rate of 3 Hz) with 18 features from polishing instructions by varying the operating parameters. The total data samples were then split into training and validation groups in a ratio of 80:20, as shown in Table 4.

Automatic code generation was performed to collect the Yaskawa Polishing Motion Datasets. Robots can be programmed as multi-axis machines that can save manual teaching time, increase trajectory smoothness, and be used for a wide range of manufacturing applications, such as polishing, drilling, welding, trimming, 3D printing, and robot machining. Therefore, implementing this method can reduce the EC at the beginning of the production phase. The implemented wood-polishing task using the GP7 IR is shown in Fig. 9.

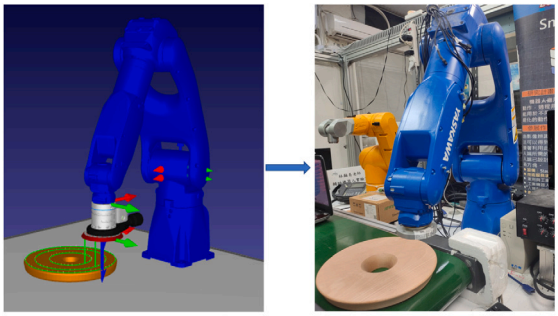


Fig. 9. (a) Automatic trajectory generation in RoboDK; (b) Deploying the auto-generated code into physical IR controller.

4.5. Loss function and model evaluation

The stochastic gradient descent (SGD) with the mean squared error (MSE) in Eq. (13) is used to compute the loss function.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(E - \hat{E})^2 \tag{13}$$

In this study, we used the RMSE to evaluate our proposed method, which is expressed as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(E - \hat{E})^2} \tag{14}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{(E - \hat{E})}{E}\right| \tag{15}$$

where E denotes the measured energy value, $\hat{E}$ denotes the predicted energy by the model, and m defines as the number of times the summation iteration happens. Also, we use MAPE metrics to emphasize more on the relative error between predicted and actual values on the percentage unit.

4.6. BN-LSTM-based power modelling of UR3e cobot

We used the UR dataset from IEEE DataPort [10] to train the adopted BN-LSTM-based power model. After cleaning the null values from the UR3e cobot dataset, the total data samples were divided into training and validation sets in a ratio of 8:2. The implementation of the BN-LSTM uses an Intel® Core™ i7- 12700 CPU @ 2.11 GHz and an NVIDIA GeForce GPU computer with 32 GB RAM in MATLAB software. Table 5 shows the tuning parameters of the proposed model. The efficacy of an optimizer is generally determined by the convergence

Table 3
UR3e and UR10e no-payload dataset.

Type	File number	Train data shape	File number	Test data shape
UR3e	[4, 5, …, 23]	(72679,30)	[24,25,26]	(5000,30)
UR10e	[1, 2, …, 10]	(12607,30)	[11]	(1455,30)

Table 4
Yaskawa GP7 no-payload dataset.

Robot type	Train input	Train target	Val. input	Val. target	Test input	Test target
GP7	(18×51680)	(1×51680)	(18×12920)	(1×12920)	(18×3000)	(1×3000)

Table 5
Parameters of UR3e and UR10e power model.

Parameters	UR3e power model	UR10e power model
No. of input Features	30	30
No. of Hidden Unit	256	256
No. Responses	1	1
Epochs	300	300
Min BatchSize	32	32
LRDrop Period	200	200
Learning Rate	0.1	0.01
LRDrop Factor	0.1	0.1
Val Frequency	20	20
Solver	sgdm	sgdm

speed (the process of reaching a global optimum for gradient descent) and generalization (the performance of the model on new data). Two optimization solvers were used in this study to train the model: sgdm and Adam. Based on the results, the sgdm solver had a better convergence speed and performance than the Adam solver. The sgdm optimizer solver model can accurately learn the rules in the training data, create a precise power-prediction model, and produce positive validation results after training. Fig. 10 shows that the model can accurately predict the power data for unknown test data and produce a satisfactory fitting result.

After training the adopted power model, the best RMS error value of the validation dataset was 2.50 W, and the RMS error value of the unseen test dataset was 2.82 W, which was 56.74% lower than the error in [9]. Consequently, the BN-LSTM power model can achieve up to 99% accuracy in predicting unseen test datasets. Therefore, the adopted model is better at making predictions than the other methods [9,10], as shown in Table 6.

4.7. Comparison of adopted BN-LSTM model over UR3e dataset with other baseline methods

We also investigated several machine learning and deep learning models, in addition to the adopted BN-LSTM model, to find an optimal power and EC prediction model for IRs. First, we implemented traditional machine learning models, such as RTs, ETs, LR, and SVR, and the SVR model outperformed the other machine learning models. For instance, as shown in Table 7, the Coarse Gaussian SVR model shows an impressive prediction performance with a validation RMS error of 2.844 W and a test RMS error of 2.92 W. Furthermore, to investigate the computation time, we performed specific test for best two model, namely Coarse Gaussian SVR and BN-LSTM. Here, the Coarse Gaussian SVR performed in 21 min and 45 s, whereas the BN-LSTM performed in 34 min. Fig. 11 shows the loss and epoch of gaussian coarse and BN-LSTM. For the current states of research, we do not consider the computation time to determine the best model prediction model.

Second, we investigated several deep learning models and observed that their performance is better and more consistent than that of traditional machine learning when they deal with a more complex dataset. We adopted single- and multi-layered feedforward neural network, hybrid CNN-GRU, and BN-LSTM network models. The BN-LSTM model achieved a validation RMS error, test RMS error, and test mean absolute error of 2.50 W, 2.82 W, and 2.10 W, respectively. Thus, the BN-LSTM model achieves better prediction accuracy compared with the deep learning and machine learning models using the UR3e dataset, as shown in Table 7.

4.8. Transfer learning based power modeling of UR10e cobot

The well-trained UR3e power model was tuned using the UR10e training dataset based on the transfer learning approach. The UR10e cobot dataset was randomly divided into training and validation sets in a ratio of 8:2, as shown in Table 3. The final tuning parameters and comparison results are listed in Tables 5 and 8, respectively. The results showed that the convergence speed of the UR10e power model improved significantly, and the power model could accurately predict unfamiliar test data as depicts in Fig. 12. We observed that the training and validation loss stabilized smoothly, and Fig. 13 shows the best-fitting curve for the validation, training, and test datasets. Finally, when using the transfer learning approach on an unseen test dataset, the prediction error of the proposed power model for the UR10e cobot was 3.67 W, which is 23.22% smaller than the error in the [9]. Therefore, the BN-LSTM-based power model outperformed the other methods in [9,10].

4.9. Validation of the adopted model using Yaskawa GP7 polishing motion dataset

We also tested the adopted model using the dataset from Yaskawa GP7 IR. Table 9 presents the tuning parameters of the proposed model. Two optimization solvers were used to train the model: sgdm and Adam. Based on the results, the sgdm solver had a better convergence speed and performance than the Adam solver. The sgdm optimizer solver model can accurately learn the rules in the training data, create a precise power and energy prediction model, and produce positive validation results after the training. As shown in Fig. 14, the model could precisely forecast the power and energy data for the validation data and had a satisfactory fitting outcome. After the adopted power flow and EC model were trained, the RMS error values for the validation dataset were 1.27 W and 0.134 mWh, respectively, and a prediction accuracy up to 99% could be achieved on unseen test datasets. Using Eq. (15), we also can evaluate its relative error, which is less than 1%. Tables 10 and 11 compare the performances of the two optimization solvers over the Yaskawa GP7 dataset.

4.10. Power and EC prediction results in specific polishing task

We considered a well-trained model to predict unknown polishing instructions. Subsequently, we fed the trial test dataset to obtain the desired power and EC prediction values for the specific polishing task. We collected 3000 data samples from the polishing instructions by varying the operating parameters. Finally, we conducted a test to compare the prediction accuracy of the adopted model between the measured and predicted values in the test dataset. Fig. 15(a) and (b) show the power and EC fitting plot results for the predicted and measured values, respectively. Yaskawa GP7 IR demonstrated an accurate prediction of

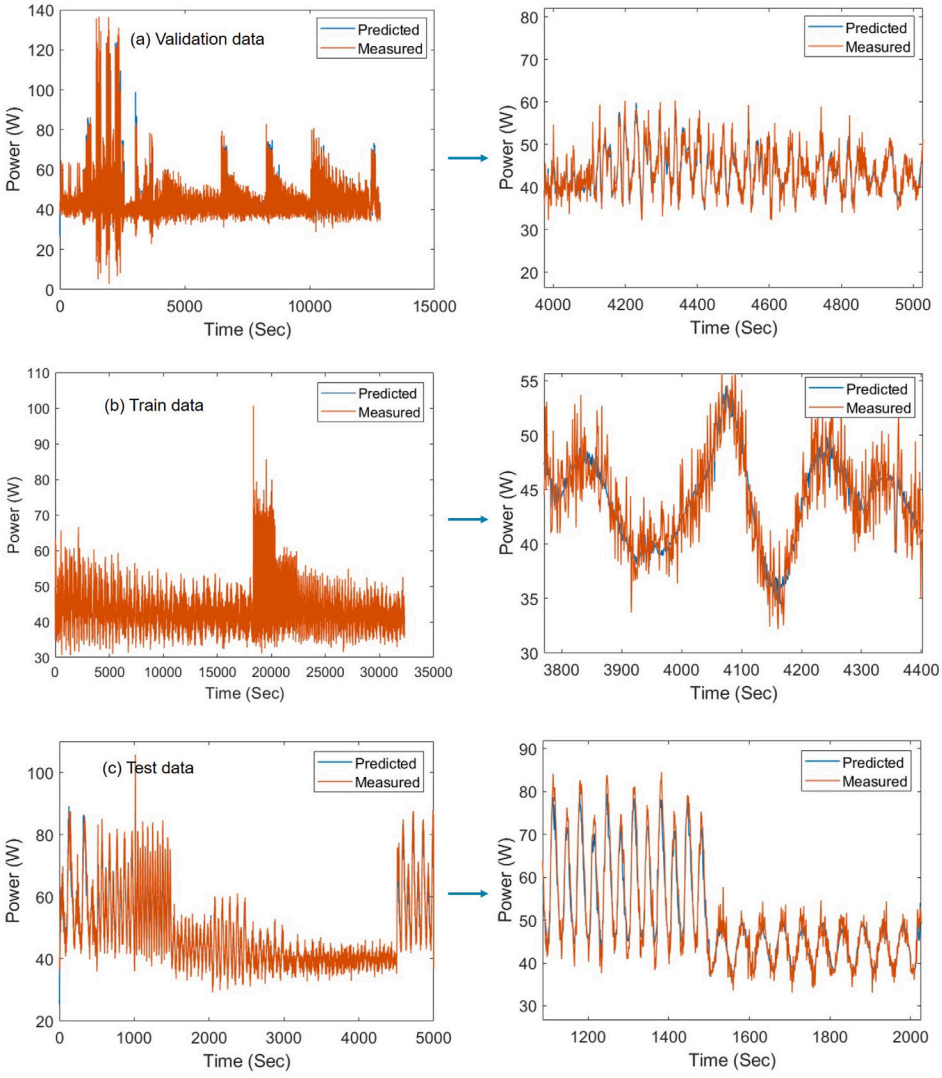


Fig. 10. UR3e power fitting results (a) Validation dataset; (b) Train dataset (File: 15,16,17); (c) Test dataset (File: 24, 25, 26).

Table 6
Comparison of UR3e power model results.

	RMSE [W]		
Dataset	Proposed method	Method in [9]	Method in [10]
Validation dataset	2.50	2.22	2.84
Train dataset (15, 16, 17)	2.22	2.28	1.93
Test dataset (24, 25, 26)	**2.82**	6.52	–

Table 7
Performance of different machine learning and deep learning models over UR3e dataset.

	UR3e power model [W]		
Mothod	Validation RMSE	Test RMSE	Test MAE
Regression Trees	22.14	5.42	3.99
Ensembles Trees	21.08	4.168	3.05
Linear Regression	9.57	12.62	9.70
Fine Gaussian SVR	4.20	9.58	7.09
Coarse Gaussian SVR	2.84	2.92	2.29
Narrow FFNN	5.18	3.49	2.71
Bilayered FFNN	4.02	3.19	2.32
CNN-GRU [31]	9.72	9.15	4.43
1D-ResNet14 [9]	**2.22**	6.52	–
BN-LSTM	2.50	**2.82**	**2.10**

the power profile with an RMS error of 1.37 W and an energy model with an RMS error of 0.123 mWh. Both optimization solvers could achieve less than ±1% of the average error between the measured and predicted values, and could predict the target values with up to 99% accuracy on unfamiliar test datasets. However, the sgdm solver outperformed the Adam solver in terms of the overall performance.

4.11. Discussion

In view of the increasing energy prices and growing environmental awareness, engineers and researchers are seeking new solutions to reduce EC in manufacturing. This is the case for IR and automatic systems, for which the reduction in energy demand was not previously considered a design goal. An excellent opportunity to reduce the amount of energy consumed by a robotic system is provided by the efficient design and implementation of several methods. Therefore, in this

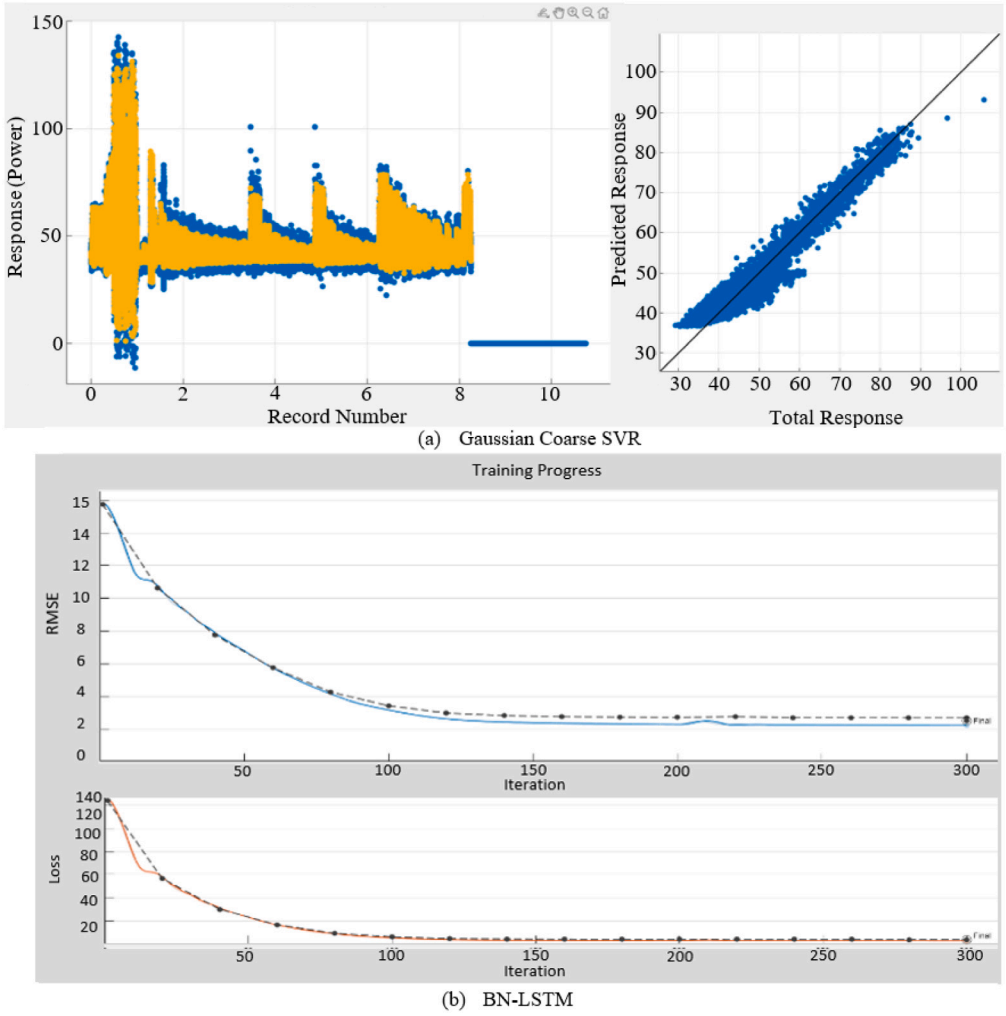


Fig. 11. Gaussian Coarse and BN-LSTM.

Table 8
Comparison of transfer learning based UR10e power model results.

	RMSE [W]		
Dataset	Adopted method	Method in [9]	Method in [10]
Validation Dataset	2.44	2.54	3.87
Train Dataset[3, 4]	2.25	2.30	5.5
Test Dataset [11]	**3.67**	4.78	–

Table 9
Parameters of the LSTM model.

Parameters	GP7 power model	GP7 EC model
No. of input Features	18	18
No. of Hidden Unit	256	256
No. Responses	1	1
Epochs	300	300
Min BatchSize	32	32
LRDrop Period	200	200
Learning Rate	0.1	0.1
LRDrop Factor	0.1	0.1
Validation Frequency	20	20
Solver	sdgm,adam	sdgm,adam

Table 10
Results comparison on GP7 Power model.

Solver	Validation RMSE	Test RMSE	Test MAE
sgdm	**1.27**	**1.37**	**0.642**
adam	1.32	2.17	1.898

Table 11
Results comparison on GP7 EC model.

Solver	Validation RMSE	Test RMSE	Test MAE
sgdm	**0.134**	**0.123**	**0.0618**
adam	0.336	0.5663	0.5310

study, we attempted to classify and analyze a data-driven EC modeling approach compared with existing methods, such as dynamic modeling, simulation, and intelligent-model-based methods, for enhancing the energy performance of IRs. We adopted the BN-LSTM model to construct a robust EC model over the existing data-driven-based model, which outperformed the baseline methods we examined.

Thus, the core concept is to propose a model that can produce accurate predictions, regardless of the complexity of the dataset. We can avoid implementing more complex models unnecessarily, based

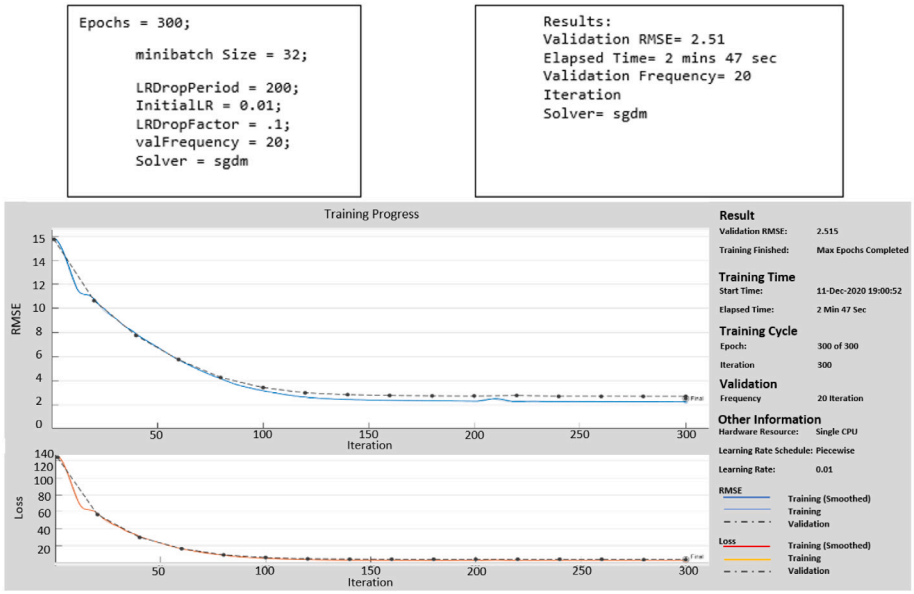


Fig. 12. Convergence time on Transfer learning.

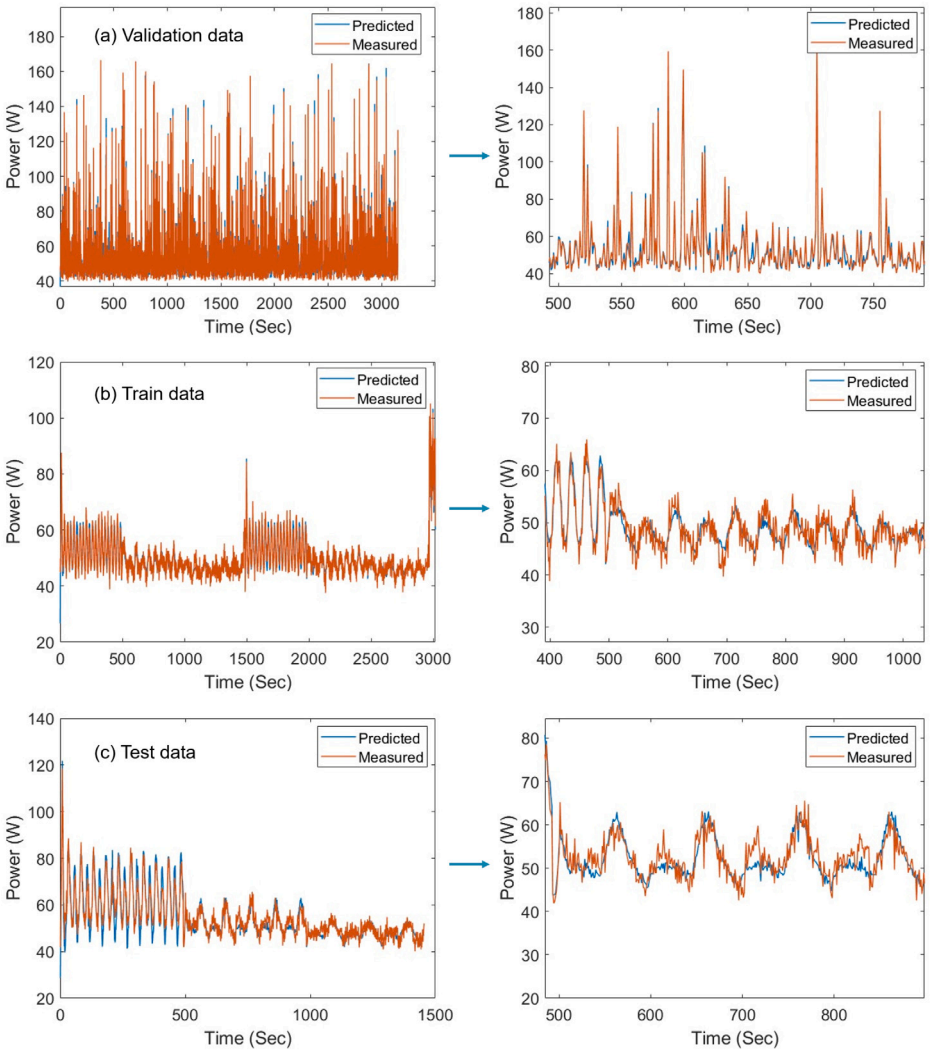


Fig. 13. UR10e power fitting results (a) Validation Dataset; (b) Train Dataset (File: 3, 4); (c) Test Dataset (File: 11).

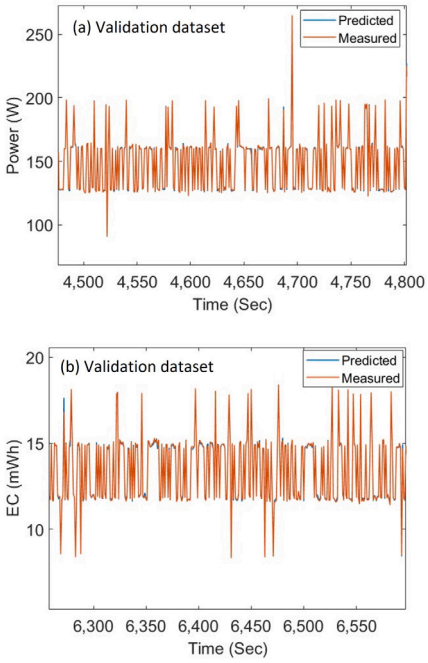


Fig. 14. sgdm solver: (a) Power validation dataset fitting plot; (b) EC validation dataset fitting plot.

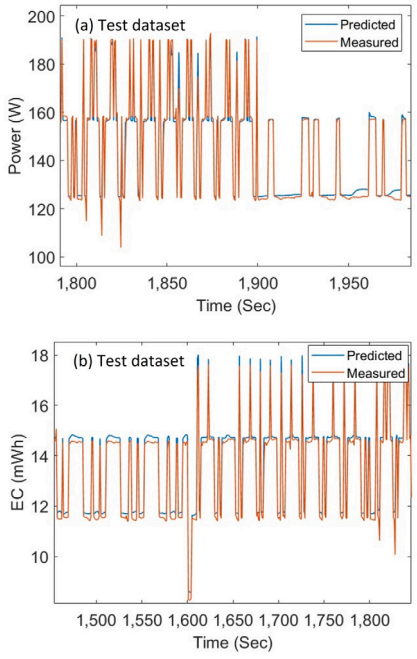


Fig. 15. Unseen test data from polishing instruction (MOVL, MOVJ, and MOVC) (a) Power prediction in Watt and (b) Energy prediction in mWh.

on the dataset and purpose. From the extensive investigation of the different models, it was observed that the BN-LSTM, MLP, and SVR models show a better performance than other machine learning and deep learning models, such as LR, DTs, ETs, 1D-ResNet14, and CNN-GRU, with complex datasets, although machine learning models are more efficient with less complex sequence datasets than deep learning models in terms of a faster convergence rate. When building an IR EC model, the BN-LSTM-based deep learning model demonstrated impressive test accuracy (98%–99%) and consistency, regardless of the complexity of the dataset.

5. Conclusion and future work

This study provided an in-depth look at the research and real-world methods used to predict the power profile and EC of IRs before they are used in a production system. We adopted a BN-LSTM-based data-driven modeling method to predict the power and EC of IRs.

The adopted LSTM-based model could accurately predict the power of the UR3e cobot with 56.74% smaller error than the method in [9]. Furthermore, the UR10e power model could be rapidly constructed while the total amount of data was reduced by approximately 80% using the transfer learning approach. Moreover, the test prediction error of the UR10e power model was reduced by 23.22% compared with that in [9]. In addition, we investigated the performance of the adopted BN-LSTM model over several machine learning and deep learning models using the UR3e datasets. The adopted model outperformed the other models that we examined.

The adopted BN-LSTM model was successfully validated using the Yaskawa polishing motion dataset with 3,000 unseen test data samples and 18 features. The proposed Yaskawa dataset provides less complexity than the UR3e and UR10e datasets.

In the future, we will use a different payload scenario. The main concern is to increase productivity and achieve cost-effective robotic operation, as the data-driven deep learning model and transfer learning method have opened the door to a wide range of applications to construct automatic commercial products that can optimize the EC of IRs. We will also investigate the direct carbon footprint of any IR operation to propose an optimal solution for reducing carbon emissions.

CRediT authorship contribution statement

Hsien-I Lin: Conceptualization, Methodology, Formal analysis, Supervision, Funding acquisition, Writing – review & editing. **Raja Mandal:** Investigation, Software, Data curation, Writing – original draft, Visualization, Validation, Writing – review & editing. **Fauzy Satrio Wibowo:** Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hsien-I Lin reports financial support was provided by National Science and Technology Council.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Science and Technology Council under NSTC 111-2221-E-A49-192-MY3.

References

[1] J. Oláh, N. Aburumman, J. Popp, M.A. Khan, H. Haddad, N. Kitukutha, Impact of Industry 4.0 on environmental sustainability, Sustainability 12 (11) (2020) 4674.
[2] Y. Li, J. Dai, L. Cui, The impact of digital technologies on economic and environmental performance in the context of industry 4.0: A moderated mediation model, Int. J. Prod. Econ. 229 (2020) 107777.
[3] C. Schmidt, W. Li, S. Thiede, S. Kara, C. Herrmann, A methodology for customized prediction of energy consumption in manufacturing industries, Int. J. Precis. Eng. Manuf.-Green Technol. 2 (2015) 163–172.
[4] G. Carabin, E. Wehrle, R. Vidoni, A review on energy-saving optimization methods for robotic and automatic systems, Robotics 6 (4) (2017) 39.

[5] D. Richiedei, A. Trevisani, Analytical computation of the energy-efficient optimal planning in rest-to-rest motion of constant inertia systems, Mechatronics 39 (2016) 147–159.
[6] M. Brossog, M. Bornschlegl, J. Franke, Reducing the energy consumption of industrial robots in manufacturing systems, Int. J. Adv. Manuf. Technol. 78 (2015) 1315–1328.
[7] M. Zhang, J. Yan, A data-driven method for optimizing the energy consumption of industrial robots, J. Clean. Prod. 285 (2021) 124862.
[8] J. Yan, M. Zhang, A transfer-learning based energy consumption modelling method for industrial robots, J. Clean. Prod. 325 (2021) 129299.
[9] M. Yao, Q. Zhao, Z. Shao, Y. Zhao, Research on power modelling of the industrial robot based on ResNet, in: 2022 7th International Conference on Automation, Control and Robotics Engineering, CACRE, IEEE, 2022, pp. 87–92.
[10] J. Heredia, C. Schlette, M.B. Kjærgaard, Data-driven energy estimation of individual instructions in user-defined robot programs for collaborative robots, IEEE Robot. Autom. Lett. 6 (4) (2021) 6836–6843.
[11] F.J. Abu-Dakka, F. Rubio, F. Valero, V. Mata, Evolutionary indirect approach to solving trajectory planning problem for industrial robots operating in workspaces with obstacles, Eur. J. Mech. A Solids 42 (2013) 210–218.
[12] A. Abe, Minimum energy trajectory planning method for robot manipulator mounted on flexible base, in: 2013 9th Asian Control Conference, ASCC, IEEE, 2013, pp. 1–7.
[13] W.P. Bailon, E.B. Cardiel, I.J. Campos, A.R. Paz, Mechanical energy optimization in trajectory planning for six DOF robot manipulators based on eighth-degree polynomial functions and a genetic algorithm, in: 2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control, IEEE, 2010, pp. 446–451.
[14] G. Field, Y. Stepanenko, Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators, in: Proceedings of IEEE International Conference on Robotics and Automation. Vol. 3, IEEE, 1996, pp. 2755–2760.
[15] C. Hansen, J. Öltjen, D. Meike, T. Ortmaier, Enhanced approach for energy-efficient trajectory generation of industrial robots, in: 2012 IEEE International Conference on Automation Science and Engineering, CASE, IEEE, 2012, pp. 1–7.
[16] D.G. Wilson, R.D. Robinett, G. Eisler, Discrete dynamic programming for optimized path planning of flexible robots, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vol. 3, IROS IEEE Cat. No. 04CH37566, IEEE, 2004, pp. 2918–2923.
[17] P. Boscariol, G. Carabin, A. Gasparetto, N. Lever, R. Vidoni, Energy-efficient point-to-point trajectory generation for industrial robotic machines, in: Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics 2015, Barcelona, June 29-July 2, 2015, European Community on Computational Methods in Applied Sciences, 2015, pp. 1425–1433.
[18] E.S. Sergaki, G.S. Stavrakakis, A.D. Pouliezos, Optimal robot speed trajectory by minimization of the actuator motor electromechanical losses, J. Intell. Robot. Syst. 33 (2002) 187–207.
[19] Y. Wang, Y. Zhao, S.A. Bortoff, K. Ueda, A real-time energy-optimal trajectory generation method for a servomotor system, IEEE Trans. Ind. Electron. 62 (2) (2014) 1175–1188.
[20] J.S. Park, Motion profile planning of repetitive point-to-point control for maximum energy conversion efficiency under acceleration conditions, Mechatronics 6 (6) (1996) 649–663.
[21] K. Paes, W. Dewulf, K.Vander. Elst, K. Kellens, P. Slaets, Energy efficient trajectories for an industrial ABB robot, Procedia Cirp 15 (2014) 105–110.
[22] A.R. Hirakawa, A. Kawamura, Trajectory planning of redundant manipulators for minimum energy consumption without matrix inversion, in: Proceedings of International Conference on Robotics and Automation. Vol. 3, IEEE, 1997, pp. 2415–2420.
[23] A. Mohammed, B. Schmidt, L. Wang, L. Gao, Minimizing energy consumption for robot arm movement, Procedia Cirp 25 (2014) 400–405.
[24] M. Gadaleta, M. Pellicciari, G. Berselli, Optimization of the energy consumption of industrial robots for automatic code generation, Robot. Comput.-Integr. Manuf. 57 (2019) 452–464.
[25] O. Wigstrom, B. Lennartson, A. Vergnano, C. Breitholtz, High-level scheduling of energy optimal trajectories, IEEE Trans. Autom. Sci. Eng. 10 (1) (2012) 57–64.
[26] S. Alatartsev, V. Mersheeva, M. Augustine, F. Ortmeier, On optimizing a sequence of robotic tasks, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE., 2013, pp. 217–223.
[27] D. Meike, L. Ribickis, Energy efficient use of robotics in the automobile industry, in: 2011 15th International Conference on Advanced Robotics, ICAR, IEEE., 2011, pp. 507–511.
[28] S. Riazi, K. Bengtsson, O. Wigström, E. Vidarsson, B. Lennartson, Energy optimization of multi-robot systems, in: 2015 IEEE International Conference on Automation Science and Engineering, case, IEEE., 2015, pp. 1345–1350.
[29] G. Gao, F. Liu, H. San, X. Wu, W. Wang, Hybrid optimal kinematic parameter identification for an industrial robot based on BPNN-PSO, Complexity 2018 (2018).
[30] T. Gedeon, K.W. Wong, M. Lee (Eds.), Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December (2019) 12–15, Proceedings, Part III. Vol. 11955, Springer Nature, 2019.
[31] M. Sajjad, Z.A. Khan, A. Ullah, T. Hussain, W. Ullah, M.Y. Lee, S.W. Baik, A novel CNN-GRU-based hybrid approach for short-term residential load forecasting, IEEE Access 8 (2020) 143759–143768.