IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received April 20, 2020, accepted April 28, 2020, date of publication May 6, 2020, date of current version June 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992853

Energy-Efficient IoT Sensor Calibration With Deep Reinforcement Learning

AKM ASHIQUZZAMAN[1], HYUNMIN LEE[2], TAI-WON UM[3], AND JINSUL KIM[1]
[1]School of Electronics and Computer Engineering, Chonnam National University, Gwangju 59626, South Korea
[2]Human IT Convergence Research Center, Korea Electronics Technology Institute, Seoul 02792, South Korea
[3]Department of Cyber Security, College of Science and Technology, Duksung Women's University, Seoul 132-714, South Korea

Corresponding authors: Tai-Won Um (twum@duksung.ac.kr) and Jinsul Kim (jsworld@jnu.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant 2018R1A2B2003774, and in part by the Information Technology Research Center (ITRC) support program, supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) under Grant IITP-2020-2016-0-00314.

ABSTRACT The modern development of ultra-durable and energy-efficient IoT based communication sensors has much application in modern telecommunication and networking sectors. Sensor calibration to reduce power usage is beneficial to minimizing energy consumption in sensors as well as improve the efficiency of devices. Reinforcement learning (RL) has been received much attention from researchers and now widely applied in many study fields to achieve intelligent automation. Though various types of sensors have been widely used in the field of IoT, rare researches were conducted in resource optimizing. In this novel research, a new style of power conservation has been explored with the help of RL to make a new generation of IoT devices with calibrated power sources to maximize resource utilization. A closed grid multiple power source based control for sensor resource utilization has been introduced. Our proposed model using Deep Q learning (DQN) enables IoT sensors to maximize its resource utilization. This research focuses solely on the energy-efficient sensor calibration and simulation results show promising performance of the proposed method.

INDEX TERMS Algorithm design and analysis, optimization, computational and artificial intelligence, battery management systems, simulation, electronic design automation and methodology, deeplearning, reinforcement learning.

I. INTRODUCTION

Internet of Things (IoT) has wide usage and is crucial to our daily life. Application of real-time monitoring and activity detection has gained a significant role in modern-day IoT sensors. IoT sensor networks are distributed within which every sensor will its own tasks to sense and acquire info from the setting [1]. However, sensors have restricted functions like short in battery period of time, low vary communication which makes them very problematic to deploy in remote areas with limited power sources. Rapid growth in the demand for energy in every sector such as communication, industrial production, domestic usage, etc. makes now the efficient usage of energy resources a very demanding research interest. IoT focused on small-scale, power-resource computers. One of the top priorities is the concept of saving energy during the operation of the IoT tools. The energy usage of underused services, particularly in a remote IoT-based testbed, measures for a significant amount of actual energy consumption. So, as to utilize this usage will both make the testbed more robust and very longer lasting than usual. IoT devices are expected to be low-powered devices but the sheer number of IoT devices that Cisco predicts will be 50 billion by 2020, is an order of magnitude larger than the number of smartphones and tablets in use today [2]. Mohammadi *et al.* [3] predicted an annual 2.7 trillion-dollar estimated market growth. This also makes an economic impact of IoT Devices far more significant to research and develop. Rise of IoT usage trending will eventually render into many sectors of industries to adapt to such a system, resulting in more energy surge in the next decade. As a result, the research into making more energy effective solutions are on the rise [4].

Reinforcement Learning (RL) algorithm is not a modern-day invention. The whole idea of these algorithms

The associate editor coordinating the review of this manuscript and approving it for publication was Honghao Gao.

hovers around natural human learning mechanisms. The main idea is to encourage positive action through some positive value, or reward and vice versa. This gives the agent or the actor/model to adapt to the real-life model scenario easily by learning rapidly. Previously, the RL algorithms were notoriously slow and not easily adapted to the environment. However, after the rise of Deep Learning Neural Networks (DNN) and back-propagation gives the DNN based agent to perform well RL based learning. The agent, which essentially, a DNN with learning capacity gets weight update based on the RL policy can rapidly learn the mechanism defined in the environment and suppress any human in the same situation. Google Inc. developed the deep learning-based Reinforcement model which mastered the ancient game of Alphago and beat the human professional player [5]. This paves a way to apply RL in many sectors of the problem that cannot be optimized only by the classical machine learning models. Although much research has been focused to solve complex problems with reinforcement learning, the energy-saving mechanism of the IoT devices with the reinforcement model has not been properly explored. The idea of resource utilization in IoT devices is crucial for any real-time sensor field. The resource scarcity will make this reinforcement learning very effective to the IoT based model to adapt to longer-lasting extreme scenarios.

Deep reinforcement learning (deep RL) has emerged as an effective solution for learning how to mimic decision flows of very complex real models [6]. Most of the applications of such models are mainly focused on game-based problem-solving. The energy utilizing a section for IoT sensors with deep reinforcement learning is a promising sector for such a domain to apply reinforcement learning solutions. Although much research focused on visual puzzle solving, using the deep reinforcement learning in the IoT sensors to utilize the energy resources is not properly explored. Although the intelligent control for systems with RL is broadly excepted in machine learning applications for faster accuracy and precision, not a lot of applications were developed focused on RL control. This is mainly because of the lack of infrastructure to test and verify RL algorithms. Nowadays many simulations and testing have been developed to test and benchmark various RL and machine learning algorithms to test and benchmark its performance on simulation. The simulations are also becoming very standardized, making it very close to the real-life application scenario. This makes the RL trained in the simulation easier to deploy in real-life settings after training in the simulations.

Sensor control with deep learning was not a new concept. Much deep learning algorithm has yielded higher accuracy and proper outcome with machine learning with both real-life and synthetic data. Control systems deep learning perform spectacularly in almost every aspect of technology. But the traditional deep learning method has its limitations. Deep learning learns the patterns with data and sometimes the data needed to achieve higher accuracy. This leads to a shortcoming on deep learning-based control management as sometimes it is difficult to collect large amounts of data in real-life to build such collection. Simulation for data control also sometimes is not designed conveniently to apply in such a manner. The most important shortcomings of deep learning module are that it is pruned to bad predictions and learning with volatile and rapidly changing data. So applying deep learning in such control systems often yields bad results. In this research, we have studied the technology for sensor data control with RL algorithms. The main limitation of such control energy-efficient models is the lack of proper simulation models to test the algorithm performance and accuracy. So, in this research, we have purposed some control based simulation for energy-efficient resource management to train with both deep learning algorithms and RL algorithms and compared the result.

Power consumption minimization and intelligent resource management had seen much research. Most of the modern research in this field specifically focused on resource utilization with most various machine learning techniques. The research mostly focused on the implementation of workload in various sectors such as data center load distribution to reduce workloads, electric power station intelligent switching to reduce power consumption and energy conservation. To our best knowledge, the novel research for utilizing IoT sensors and power conservation with reinforcement learning has not yet been thoroughly explored.

In summary, the contribution of this research can be summarized as below:

- The study has proposed a new framework for simulating intelligent control and designed new simulation for testing energy-efficient models, especially for IoT sensor-based scenarios.
- Although there have been several studies followed by the development of Long-short Term Memory (LSTM)in the deep learning research, the ensemble of LSTM in reinforcement learning has not been introduced or explored properly. In this research, we have made an LSTM based deep learning neural network agent that learns the resource utilization and achieves a better result than the general deep learning agent counterpart in shorter training time-span.
- In the simulation and experiment part, we have proposed and developed a rel-life based scenario and environment based on real battery and solar resources, paving the way to the practical application of the conceptual model. The result demonstrated the state of the art improvement on closed-grid standalone sensor kit systems.

The remainder of the article is structured as follows. The related work is discussed in Section II. Section III provides the structure of our work and discusses the proposed systems. The findings of the experiments are summarized in the IV section V ends the paper and addresses possible research.

II. RELATED WORKS

Intelligent decisions based on the data have introduced a revolutionary change in the field of machine learning including classification, clustering, and regression [7], [8]. The application and implementation of deep learning were not that old. The basics of deep learning neural networks is quite new compared to the history of neural networks. The first neural network was evolved from the idea of perception. Perceptron based learning was first devised in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt [9]. It was originally described as a machine-based application rather than a program. Basically, the perceptron program is a threshold-based logical output machine based on the common idea of the inputs are being given some variance based on the importance of influence it has on the final output [10]. The final output gets through a nonlinearity or activation to provide final output. This type of multi-layered perceptron later become very useful for predicting and learning data patterns but soon fell out of favor as an effective machine learning algorithm due to the high computation power needed at that era of computing. After a decade later introduction to back propagation algorithm and faster computing power in 1989 made MLP potentially an effective algorithm to use in machine learning [11]. The later rise of GPU based parallel computing gave rise to a very deep learning neural network with many layers and various activation functions. Target specific objective learning with the big dataset is now a common use case for deep learning neural networks [12].

Idea use cases for discovering data patterns in big data sets virtually has limitless applications in all of the aspect of human knowledge discoveries [13]. But in this research for the sake of simplicity, related work for machine learning was only focused on resource utilization and sensor managements [14]. The application for energy saving IoT sensor-based deep learning model has a huge impact on Mobile Edge Computing (MEC) and Cloud Computing based infrastructures [15]. Cloud Computing & virtual platform-based research for the MEC (Mobile Edge Computing) has seen a lot of state of the arty research these days [16]. Application of MEC and cloud environment has proven to have faster-processing speed and lower latency in communication [17]. MEC environments give the opportunity to develop faster Intelligent applications which are very essential for quick service providence and recommendation based services [18]. The IoT based application for deep learning is heavily researched for better Quality of Service (QoS) and resource optimization [19]. However, This research is heavily focused on the application of only deep learning algorithms. Also, the research focused on various detection schemes and network environments rather than energy saving resource utilization [20], [21].

The main idea of reinforcement learning first coined in computer science by Sutton in 1998 [22]. It is separated into some sectors of definitions. RL can be properly explained using the concepts of agents, environments, states, actions, and rewards. The main idea of deep learning-based reinforcement learning is to incorporate the agent with deep learning neural network to achieve the highest accuracy with the agent‘s ability to master finding patterns in the state and thus giving proper action return to maximize reward [6]. This objective lead to make deepening based agent in reinforcement learning very effective in control based simulations [23].

Power consumption minimization and intelligent resource management had seen much research. Most of the modern research in this field specifically focused on resource utilization with most various machine learning techniques. The research mostly focused on the implementation of workload in various sectors such as data center load distribution to reduce workloads, electric power station intelligent switching to reduce power consumption and energy conservation. To our best knowledge, the novel research for utilizing IoT sensors and power conservation with reinforcement learning has not yet been thoroughly explored.

According to Shroud *et al.* [24], energy costs are the leading causation of total mass generation costs in the factories. In other terms, this research defined energy usage as the main factor of production costing in modern factory based production. This enables the decision-maker to approach this problem with at-most importance. The research proposes a mathematical model to minimize energy consumption by using machine learning methods to intelligently ‘‘Turn on’’ or ‘‘Turn off’’ the machine. This method will eventually reduce the machine energy consumption drastically. This research also pointed out that significant reductions in energy costs can be maintained by shifting the production time into the low priced sessions. This minimization process also has a positive environmental effect by reducing energy consumption during peak periods. This increases the chance of a reduced CO_2 emissions from the energy generators.

Mocanu *et al.* [25] proposed a new reinforcement learning, especially Deep Q learning-based online building energy optimization technique which energy scheduling strategies could be used to provide real-time feedback to consumers to encourage more efficient use of electricity. However, this research completely focused on the learning mechanisms of the energy optimization of the electric grids and do not give any insights on battery-based energy optimization.

All the research currently going on the field of energy conversion, and IoT based models energy optimization and implementation of the energy-saving method is currently not properly explored. The IoT based models need to be highly optimized for minimum energy consumption and the proposed method for this power consumption minimization will surely give a new frontier in the IoT based device power optimization with reinforcement learning. Because of a lack of infrastructure to test the reinforcement learning algorithms in real test ted scenarios, the majority of the research took the step of simulation to get the results. However, sometimes simulation lack rel life inconsistency and moreover complexity. This eventually results in a bad learning strategy.

However, our extended analysis of current literature gives the insight to make an efficient simulation for energy-efficient models with deep leaning. this is due to the fact, the complex nature of the deep learning reinforcement algorithm gives versatility and scope to real-life excellent performance due to adaptability. In this research, we have made some traditional switching algorithms to save maximum power by learning to get optimal power saving to turn off the machine in real-time using deep learning and deep reinforcement algorithms.

III. PROPOSED METHODS

In this research, we have made some traditional switching algorithms to save maximum power by learning to get optimal power saving to turn off the machine in real-time using deep learning and deep reinforcement algorithms. Basically, the deep learning algorithms and the reinforcement learning algorithms are a more generalized algorithm that can be specified and train on a subset of problem-solving by specif denotation of the problem itself. A lot of the problem-solving in real life can be generalized and thus cluster together for applying Reinforcement learning to learn the best outcome suitable for this particular problem set. The agent who learns here can be a neural network that learns from the state and observations from the environments. In this section, we describe the details of this algorithm and specific modifications done in this research to accommodate the energy-specific savings needed to achieve the defined goal. The basis of deep learning neural network, reinforcement learning and finally the proposed model for out research has been described.

A. DEEP LEARNING NEURAL NETWORK

A multi-layered perceptron (MLP) or Deep learning Neural Network DNN with multiple hidden layers can be represented graphically as the main 3 layered functioned model. In Fig. 1 shown the simplified model architecture for the DNN. The hidden layer shown in the middle can be expandable into several layers. so the sake of computational simplicity, in this explanation, we calculated the whole neural network considering a 1 hidden layer with input and output layers.

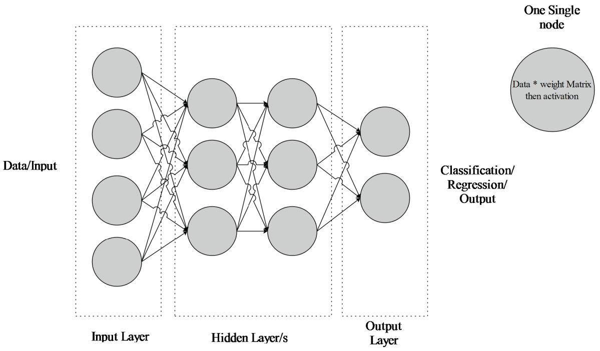

One Single node
Data * weight Matrix then activation
Data/Input
Classification/ Regression/ Output
Input Layer
Hidden Layer/s
Output Layer

FIGURE 1. Simplified architecture of deeplearning neural network.

Formally, a one-hidden-layer MLP is a function $f : R^X \to R^Y$, where X is the size of input vector x and Y is the size of the output vector $f(x)$, such that, in matrix notation:

$$f(x) = A(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x))) \tag{1}$$

with bias vectors $b^{(1)}, b^{(2)}$ weight matrices $W^{(1)}, W^{(2)}$ and activation functions A and s. The vector $h(x) = \Phi(x) = s(b^{(1)} + W^{(1)}x)$ constitutes the hidden layer. $W^{(1)} \in R^{X \times X_h}$ is the weight matrix connecting the input vector to the hidden layer. Each $column W_{\cdot i}^{(1)}$represents the weights from the input units to the i-th hidden unit. Typical choices for s include $tanh$, with $tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$, or the logistic sigmoid function, with $\sigma(x) = 1/(1 + e^{-x})$. In the newest system for the DNN activation, (Rectified Linear Unit) or Relu has been useful for non linearity [26]. In this Research, Relu has been used exclusively for this reason.

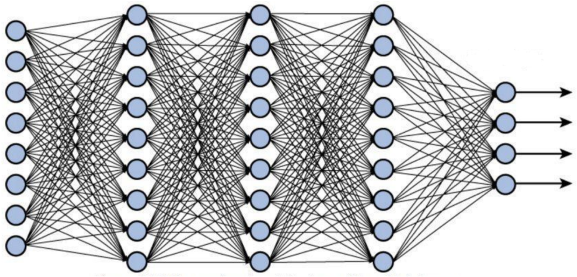


FIGURE 2. Fully connected deeplearning neural network architecture.

The design of the neural network is not a complicated matter. but the architecture of DNN results in the final outcome of the objectives. Thus, choosing a proper hyper-parameter for the DNN is crucial for DNN. These hyper-parameters heavily rely on the data input and output dimensions. In a traditional neural network, the data is often high dimensional and needs very deep architecture to classify the hidden pattern. A classical fully connected deep learning neural network with a deep hidden layer is shown in Fig. 2. Thus previously in DNN based recognition system, deep architectures have been dominated. However, in the rise of the reinforcement learning algorithms, the environment often gives out a finite state and action which makes the input and output dimension of the neural network significantly smaller.

The main shortcomings of such supervised model is that these model does not take the previous data as a whole. on the other sense, the time series concept data is not explored in this concept. Long Short Term Memory (LSTM) is the modified concept of the deepening neural network that takes the time series based values in the action. It was first proposed by Hochreiter et al [27]. LSTM is specialized RNN with a memory cell, making it particularly useful to learn from very long sequences [28]. This long sequence learning process with extended memory cell gives LSTM advantages to get long sequences. Unlike RNN, LSTM has a ''memory cell'', which gives it the holding power of a long sequence, thus making it particularity suitable for long sequences. The equations below describe how a layer of memory cells is updated at every time steps t. with the following assumptions. x_t is the input to the memory cell layer at time t.

$W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o$ & V_o are weight matrices b_i, b_f, b_c & b_o are bias vectors. Values for i_t, the input gate, $\widetilde{C}_t$ the candidate value for the states of the memory cells [29].

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$\widetilde{C}_t = ReLu(W_c x_t + U_c h_{t-1} + b_c) \tag{3}$$

Now, Forget gate activation f_t at time t will be

$$f_t = ReLu(W_f x_t + U_f h_{t-1} + b_f) \tag{4}$$

So, now the C_t the memory cells new state at time t will be,

$$C_t = i_t * \widetilde{C}_t + f_t * C_{t-1} \tag{5}$$

following by the new outputs o_t & h_t,

$$o_t = Relu(W_o x_t + U_o h_{t-1} + V_o C_t + b_o) \tag{6}$$

$$h_t = o_t * Relu(C_t) \tag{7}$$

LSTM networks are completely immune from both the vanishing or exploding gradient problem [30]. This also makes it a little tricky to train, computation wise. But the proper weight initialization and proper activation function choice can resolve this issue. In our proposed method, we have explored the LSTM model with the Q learning agent as the main learning mechanism. This along with the deep neural network architecture provides a huge boost up in the model to learn from the simulation. the detailed of the model and ensemble strategy is discussed detailed in the deep q learning subsection.

B. REINFORCEMENT LEARNING

The main idea of reinforcement learning first coined in computer science by Sutton in 1998 [22]. It is divided into certain traditional standards. Using agents, environments, states, actions, and rewards principles, RL can be adequately explained. An agent takes action. Some good examples can be a Remote controlled (RC) drone making a delivery or a player navigating in a video game. Action is the finite set of all possible moves of the agent. The action is almost self-explanatory, but it should be noted that agents choose among a list of possible actions. The world through which the agent moves is called the environment. The reward is the feedback by which we measure the success or failure of an agent's actions. The discount factor is multiplied by future rewards as discovered by the agent in order to dampen these rewards effect on the agent''s possible choice of action sets. It is designed to make future rewards worth less or more, depending on the settings, than immediate rewards. So basically, it enforces a kind of short-term hedonism or long term feedback or rewarding in the agents. The main interaction is done by the agent in the environment. Based on the agent's action it will receive some reward. This process can be described in a simple algorithm.

Algorithm 1 Basic Reinforcement Algorithm

```
Require:
  State $\mathcal{S} = \{1, \dots, n\}$
  Action $\mathcal{A} = \{1, \dots, n\}$
  Reward $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
  procedure Reinforcement($\mathcal{S}$)
    Start in state $s \in \mathcal{S}$
    while $s$ is not terminal do
      $r \leftarrow R(s, a)$
      $s' \leftarrow T(s, a)$
      $s \leftarrow s'$
    return $\mathcal{S}$,$R$
```

As shown in the Algorithm 1, The states of the reinforcement learning can be stated as $\mathcal{S} = \{1, \dots, n\}$ and the action as $\mathcal{A} = \{1, \dots, n\}$. All of these are finite sets with a discrete of continuous value that depends on the problem specification. Later, The reward policy of the function was specified and the main goal of the reinforcement learning agent is to maximize the reward based on the state and take actions. This algorithm is a self-containing loop that needs to be defined with proper exit condition. So the proper conditioning of reinforcement learning is crucial for the expected result. Often the simplification of states leads to a poorly designed condition of RL that eventually leads to poor results and often inability to not master any sort of positive action to get the reward.

C. DEEP Q LEARNING

Q-learning is a reinforcement learning algorithm in the unsupervised machine learning domain. The goal of Q-Learning is to compile the optimal policy for the agent to achieve the final goal with maximum reward. This does not require an atmosphere design and can manage issues with stochastic transformations and incentives without having to make changes. Q-learning seeks an optimal strategy for any finite Markov decision process (FMDP) in the context that this really maximizes the expected value of the maximum reward cumulative successive stages, originating from the current state. Q is the function that returns the reward which provides the reinforcement and can be shown to reflect the value of an action taken in a given state. The action-value function is defined by a function approximation, such as a neural network, in value-based model-free reinforcement learning methods. The action-value variable parameters are taught in one-step Q-learning by iterative decreasing a series of loss functions, which is basically the foundation of any DNN.

As stated from the algorithm in 2, this is an off-policy learning algorithm that searches and locates the highest possible result or movement to also be drawn given the state. It is regarded off-policy because the mechanism of q-learning understands from decisions outside of the existing policy, like taking small random actions and therefore the usual policy is not summoned on a daily basis. In addition, q-learning aims at implementing a strategy that maximizes its overall incentive. This feedback mechanism produced with this methodology often leads to greater precision in learning and optimize compensation on a series of actions. Here any type of an agent

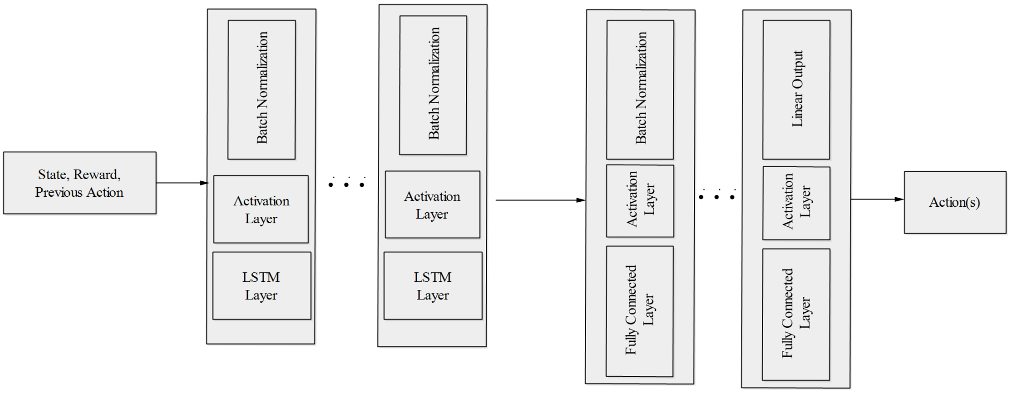


FIGURE 3. **Propsed LSTM based DQN agent for resource utlization.**

```
Algorithm 2 Simplified Q learning Algorithm
Require:
  Sate $\mathcal{S} = \{1, \dots, n_x\}$
  Action $\mathcal{A} = \{1, \dots, n_a\}$, as known    $A : \mathcal{S} \Rightarrow \mathcal{A}$
  Reward $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
  State transition $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$
  Learning rate $\alpha \in [0, 1]$, typically $\alpha = 0.1$
  Discounting factor $\gamma \in [0, 1]$
  procedure Q Learning($\mathcal{S}, A, R, T, \alpha, \gamma$)
    Initialize $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ arbitrarily
    while $Q$ is not converged do
      Start in state $s \in \mathcal{S}$
      while $s$ is not terminal do
        $\pi(x) \leftarrow \arg\max_a Q(x, a)$
        $a \leftarrow \pi(s)$
        $r \leftarrow R(s, a)$                    ▷ Receive the reward
        $s' \leftarrow T(s, a)$                   ▷ Receive the new state
        $Q(s', a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a'))$
        $s \leftarrow s'$

    return $Q$
```

with time series based data processing capability can classify the underline structure of the input and can provide the output in a shorter time. Our proposed deep learning LSTM agent ensemble can identify the action and state time series based element with the memory element in the deep learning neural network layer. The proposed model for deep Q learning is novel and has the potential to apply in various optimization problems. However, the resource utilization of the sensor in a closed smart sensor grid the main scope of the research. So the application was conceptualized to use in the above-mentioned application domain exclusively.

As shown in Fig. 3, the proposed DNN-v has the following structure. It will take the input of the previous action, state and the reward as input vectors. the later ensemble is the combination of various LSTM layers with the Batch normalization layer to reduce the overfitting and model. Later parts were fed into some classic deep learning layer, also known as the fully connected layer. The total layer and the node number vary with the actual number of inputs from the simulation or real-life circuit signal input. It can be easily chosen with a simple grid searching algorithm with sample data.

D. SIMULATION DESIGN

Designing simulation for a deep q learning environment in this study is vital. The main idea of deep q learning is that this gives the perfect accuracy for limited or finite-state environments. In the rise of Software-defined networking (SDN), intelligent control of virtual switching of the module can potentially be extremely power conservative and thus lead to a modern IoT based energy efficient module. However, the lack of a modern IoT based environment or simulator that is design to train the test just models if very limited or virtually close to none. This results to design a proper environment simulation for the DQN to learn. In other words, the proper design of an environment to fit into q learning state and action loop with proper reward feedback can result in tremendous learning and achieving high accuracy in a short time. For our study, the development or design of this simulation was fully inspired by the natural system. In other words, The simulation design of this study was based on the data collected from real-life machine and then based on the data all the state transfer were designed and the probability functions were sampled from the collected data-sets.

Battery or resource life analysis is vital for optimizing the power sources in the sensor module [31]. For certain wireless sensor networks of medium-range, while a device's average current consumption is small, the instantaneous current may be high. Also, when a battery is discharged at a high and constant rate, particularly though there are still active materials remaining in the battery, the battery reaches its end of existence. The real efficiency of a battery can be calculated experimentally for a particular scenario of usage. we have calculated the load simulation and the resource utilization in a very similar manner based on real-life Zigbee network power consumption analysis.

As followed by the research in [31], we have build up a use case scenario of the IoT sensor data transmission and receiving power consumption Profile. A typical 5G millimeter wave high power antenna module is considered for our research simulation. The mentioned 5G mm wave has 800 MHz bandwidth and 2 × 2 MIMO down link and up link module [32]. The power consumption model can be broken down into several parts. The main consumption of power is the sensor wake up and information processing. So, in average a sensor had a total energy consumption of 0.001 *mAH* (milli Ampere Hour). But this varies depending on the transmission receives an acknowledgment as these might take multiple times to execute depending on weather and other interference.

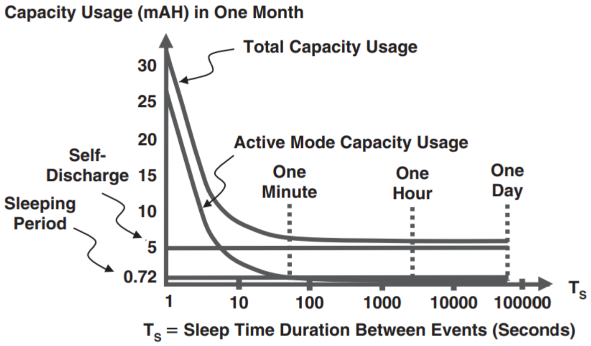


FIGURE 4. **A demo usage cycle for various capacity of a *Li*-ion battery.**

Lithium-ion (*Li*) batteries are now the most common form of batteries in the world. Even though the efficiency of these batteries is very high, the performance of the battery life might not be efficient for a specific use-case. for example, the nominal battery size for a given device is 300 mAH with an output of 60 percent, 180 *mAH* is the total power to be used in battery life measurements. As shown in Fig. 4, total capacity used in 1 month for different activities based on 1000 *mAH* nominal capacity, 50% battery efficiency, and 1% Self-discharge per month. The resource simulation was based on these Li batter activities. The simulation was made with the same battery profile mention in [31], [33]. In previous times, The size or energy capacity of the battery was increased in lieu of maintaining the demand. But this makes the operating temperature an issue in high performance. The most important factor here is the logistical maintenance. the whole system might fail in a rapid discharge scenario. These problem has made the single giant unit power resource very fragile and prone to poorer performance. Nowadays most of the power resources highly depend on these multi-pack *Li*-ion batteries [34].

Fig. 5 shows the proposed block grid diagram of the simulation that we have designed based on our studies in the battery resource. the main sensor is denoted as S. V is the external charging source. for simplification, we have considered this as a solar panel or low priority based grid electric supply source. The $F1$, $F2$ are the logic-based control gate that manages the circuit connection and the main ADC board C has all the energy info of all 6 batteries $B1$ to $B6$. This system can easily be considered a micro-grid. The functioning of the micro-grid is defined by means of state space $\mathcal{S}$, operational space $\mathcal{A}$, dynamic and cost/reward function $\mathcal{R}$ as the sequential decision-making problem. The evolution of the system is defined as a discrete, one-minute process over a finite time horizon.

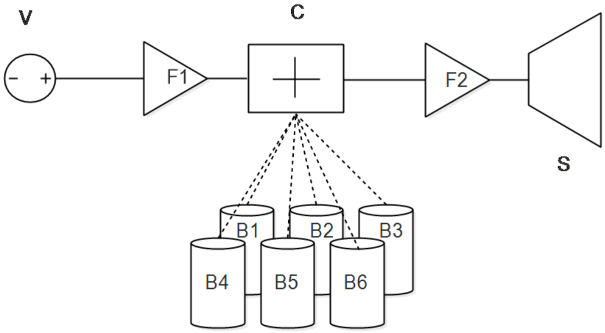


FIGURE 5. **Proposed multi-pack resource sensor block diagram.**

Let $s \in S$ is a time-dependent vector with time as t_d, t_m, t_s as the day, minutes and seconds and the V is the input charging mechanisms. The S_L is the sensor load and the B_{EV} is the energy level of all batteries. So, the observation space of the consisted of the state and following

$$\mathcal{S} = \{t_d, t_m, t_s, \ldots, \Sigma B_{EV}, V\} \tag{8}$$

$$\mathcal{A} = a^{crg} \cdot a^{dcrg} = 0 \tag{9}$$

The action space consists of charging a^{crg} and discharging a^{dcrg} actions of the battery, The actions are in discreet amounts of energy in [mAH]. The actions are constrained by (9) ensuring the battery is not charged and discharged at the same time. The battery capacity can be calculated as follows

$$E_{t+1} = E_t + \mu a^{crg} - \frac{a^{dcrg}}{\mu} \tag{10}$$

Here both E_{t+1} and E_t denotes the capacity of battery in 2 consecutive time states. The discharging is proportional to the sensor load which is calculated as avg of 0.001 mAH in per execution cycle. So the reward to calculated as they maximize the charge with usage but minimize the charge-discharge cycle. The maximize and minimize can be any arbitrary value. But for our simulation practically, we have decided the total reward should me 200 which is based on the discharge and charge values of real Batteries as shown in Fig. 6.

The experiments were setup in two different parts. Fist the assemble of the neural network is done. This is done by composing the LSTM based module to learn from the data that is generated the simulation with the right action taken by the human. In the Later part the "Learned Agent" is transferred in the simulation as an agent with the input as the observations and the output as the action taken by the agent for the previous observation. The simulation has the battery energy status with the previous step action and reward on the

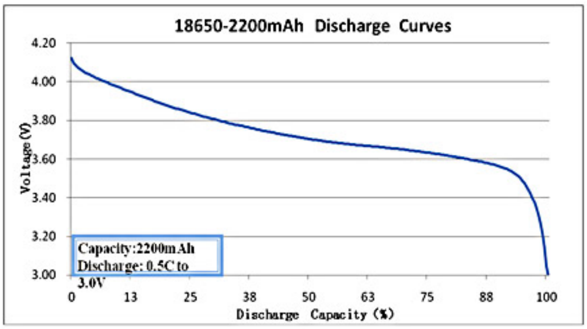


FIGURE 6. Discharge curve of a 2200 *mAH* battery.

observation input with the preferred action as the output. then the reward of the steps were calculated and then send as the same step observation to predict the next step output.

IV. EXPERIMENTS

Experiments are an essential part of any scientific exploration. For our proposed method building the proper simulation is extremely crucial. The simulation was build based on the theory that is described in Section III-D. The simulation was build in an ecological system that is easier to control by any deep learning neural network. In recent years, the reinforcement learning framework has been rapidly developed for fast prototyping. However, most of the simulations were exclusive to simple computer-generated graphics games and simple control systems that are heavily reliant on the human-based interfaces such as button and two-dimensional image output. But based on the mathematical equation that is developed and conspired in Section III-D, a simple simulation can be made in the python based system reinforcement learning platform Gym [35]. Fig. 7 shown a dynamic sensor energy demand generated in the simulation in each time step based on Equation 10.

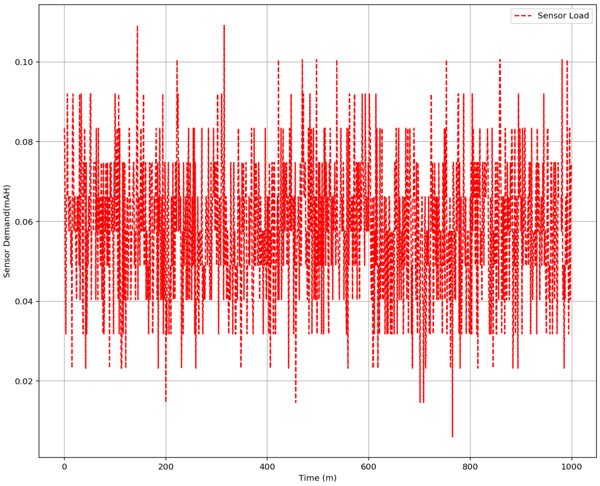


FIGURE 7. Simulation sensor energy load generation.

Firstly, we have made the LSTM based neural network agent that is described in Fig. 3. The neural network was designed by Keras open source library [36]. The first part of the experiment was done by collecting data that is designed for the neural network to learn and make a baseline to access the later DQN model. The model has the class 3 input as the current state reward and previous action, as it gives back the result of its expected output action. For collecting this data set, a human agent takes the proper action for 1 million steps and the data is then exported as a numerical output.

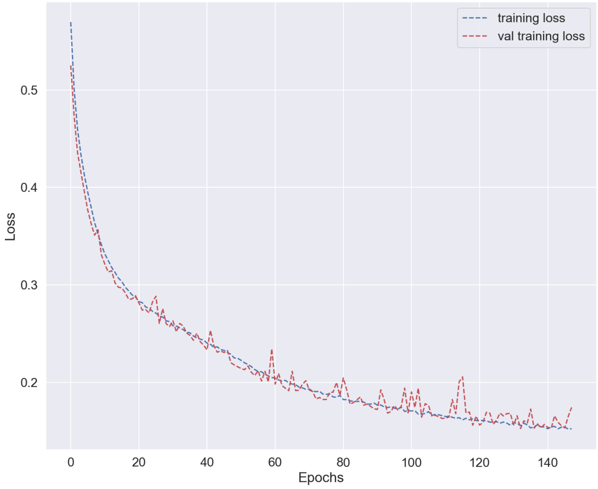


FIGURE 8. DNN loss in training and testing.

Fig. 8 shows the training loss or error minimizing of the neural network in classical training and testing and Fig. 9 shows the accuracy of the same process.

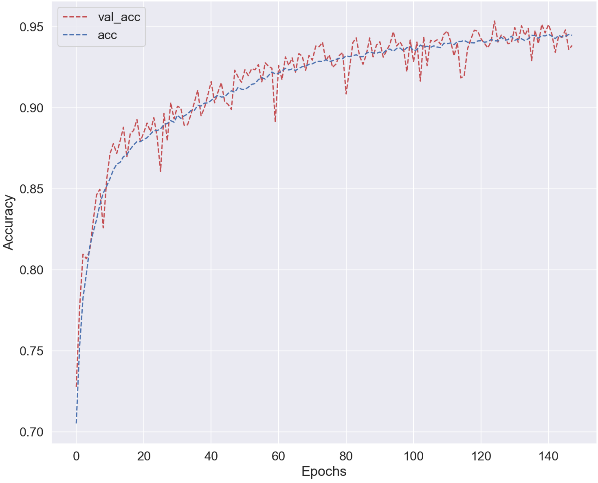


FIGURE 9. DNN accuracy in training and testing.

The overall accuracy of the neural network later achieves an astonishing 98% maximum over-training and testing scenarios. But the accuracy is not the main concern of these experiments as the data is very versatile and prone to change.

This is the main short-come of DNN. The training data variation to the slightest degree drop the accuracy to a shear 50% which is blind guessing and that is fundamental for DQN shortcoming in such rapid changing situation based simulations. However, this learned weight of the LSTM agent is important as the transfer learning-based approaches were taken later in the experiment.

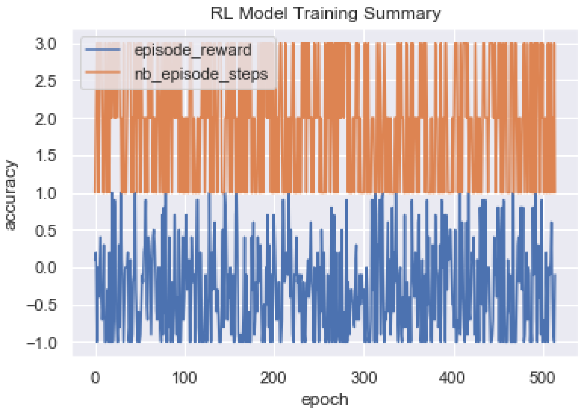


FIGURE 10. Episode steps during DQN training.

Fig. 10 shows the graphical representation of the data generated for the RL training in the simulator. The trained model was learn based on the reward policy generated in each episode of the models. The total steps for the simulation were set at $1, 000, 000$. The agent learning configuration was set as following, the discount factor or γ was set to 0.97 to look into a future reward for maximizing the conservation. Fig. 11 shows the total episode rewards in the total simulation run-time for an agent.

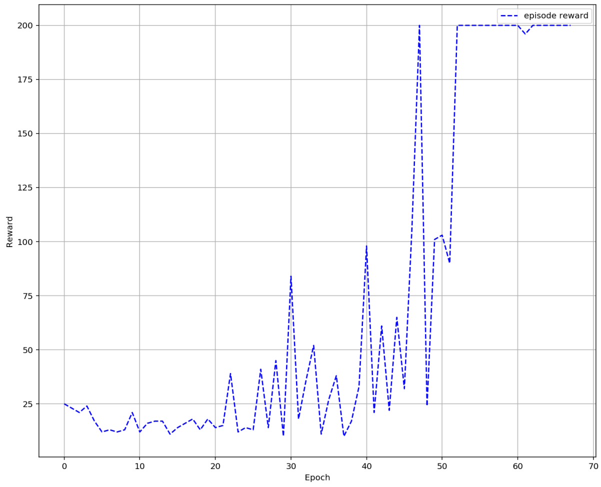


FIGURE 11. Episode based reward for the LSTM agent.

Based on the saved energy and utilization, the model receives a reward that is shown in Fig. 11 A real-world application of using reinforcement learning to control a battery would have to deal with both a variable price profile. The biggest advantage of (deep) reinforcement learning is that it could be exploited remotely in a simulated environment. After the model is fully trained in such an off-line platform, If this online environment has changed a little bit from the learning environment, the reinforcing learning paradigm will recognize such adjustments by default and adjust its actions dynamically to obtain the best outcome. This example is demonstrated in Fig. 12. The rapid change in the data in the first sectors of iterations eventually thrown the neural network actor into bad performance. But it soon learns from the chance of the data set in the end and corrects on the error thus the rise of the reward and the minimization of the error occurs. Basically, the ever-changing variables in the simulation and the real-life situation converge properly in this use case.

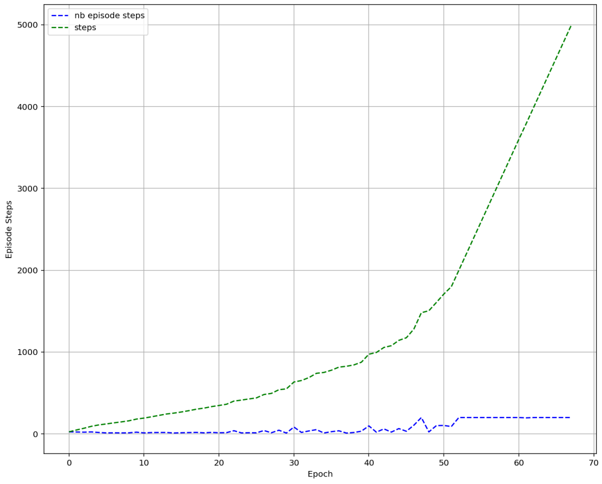


FIGURE 12. Episode steps taken in the DQN training.

A. DISCUSSION

Deep learning, as well as reinforcement learning, are both programs that operate independently. Even though deep learning neural networks had often shown remarkable accuracy in all of the practical learning problems, it has a major drawback in rapid deployment. Deep learning neural network needs to train in a specific manner in with a lot of data to learn a specific pattern. This makes a very high volume data necessity for proper DNN based classification or detection problem. Additionally, DNN also performs well while the training and test data are drawn from the same domain and then the same process is observed. The distinction between them is that deep learning is training from a data frame but then implementing that learning to a new data set while reinforcement learning is interactively learning through adjusting actions based on constant responses to optimize reward. So this reward maximizing process is constant and this creates eventually a buffer that always sets and corrects the data and the learning of the DNN is still possible. This type of implementation is particularly helpful in the rapidly changing environments and helps the system to outperform all other algorithms by acting as a whole module.

LSTM modules are the modified version of the DNN that utilize memory gate mechanism to consider the time series element of the data and gives the neural network the ability to consider the previous moves in the decision making. In general, using LSTM module in the agent improves the performance as the input observance is highly time sensitive. The idea of transfer learning form the learn general LSTM model to the DQN to boost up model performance is a new approach to reinforcement learning. This experiment paves a way to develop modern 5G standalone high-performance durable micro-grid based substation with multiple battery packs that can revolutionize the future telecommunication and internet.

V. CONCLUSION

The sensor used for energy preservation is also an important mission in ensuring dependability and accessibility in managing and monitoring a sensor network as they have the battery and computational storage constraints. Though the work on sensor data for the new sensor network generation is still cumbersome. Our proposed solution to use sensors using RL algorithms is, to the best of our knowledge, the first attempt to understand the usage of deep learning base reinforcement learning in conjunction with sensors based on energy conservation utility. In this research, we have generalized the estimation of an information utility for RL algorithms. Later we have designed a sensor based closed grid resource management system and proposed a new model to utilize the battery or power consumption of the IoT node sensors. The whole simulation was successful indicative of the real-life RL deployment in the real-time IoT based devices later to enable RL based optimization later. The promising result opens up a new research frontier for RL based IoT energy conservation and collaboration in real-time in the future.

REFERENCES

[1] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Netw.*, pp. 1–17, Nov. 2019.

[2] J. Dofe, J. Frey, and Q. Yu, "Hardware security assurance in emerging IoT applications," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2050–2053.

[3] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.

[4] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services," *IEEE Internet Things J.*, early access, Dec. 2, 2019, doi: 10.1109/JIOT.2019.2956827.

[5] E. Gibney, "Google AI algorithm masters ancient game of go," *Nature*, vol. 529, no. 7587, pp. 445–446, Jan. 2016.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[7] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[8] A. Pradhan, "Support vector machine-a survey," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 8, pp. 82–85, 2012.

[9] F. Rosenblatt, *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Buffalo, NY, USA: Cornell Aeronaut. Lab., 1957.

[10] G. Zini and G. d'Onofrio, "Neural network in hematopoietic malignancies," *Clinica Chim. Acta*, vol. 333, no. 2, pp. 195–201, Jul. 2003.

[11] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-propagation algorithm which varies the number of hidden units," *Neural Netw.*, vol. 4, no. 1, pp. 61–66, Jan. 1991.

[12] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10. Cambridge, MA, USA: MIT Press, 1995, p. 1995.

[13] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "IPrivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 5, pp. 1005–1016, May 2017.

[14] J. Yu, M. Tan, H. Zhang, D. Tao, and Y. Rui, "Hierarchical deep click feature prediction for fine-grained image recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 30, 2019, doi: 10.1109/TPAMI.2019.2932058.

[15] Y. Yin, F. Yu, Y. Xu, L. Yu, and J. Mu, "Network location-aware service recommendation with random walk in cyber-physical systems," *Sensors*, vol. 17, no. 9, p. 2059, 2017.

[16] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[17] Y. Yin, L. Chen, Y. Xu, J. Wan, H. Zhang, and Z. Mai, "QoS prediction for service recommendation with deep feature learning in edge computing environment," *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 391–401, Apr. 2020.

[18] Y. Yin, J. Xia, Y. Li, Y. Xu, W. Xu, and L. Yu, "Group-wise itinerary planning in temporary mobile social network," *IEEE Access*, vol. 7, pp. 83682–83693, 2019.

[19] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Oct. 15, 2019, doi: 10.1109/TCSVT.2019.2947482.

[20] L. Liu, Y. Cheng, L. Cai, S. Zhou, and Z. Niu, "Deep learning based optimization in wireless network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[21] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro, and F. Kawsar, "DeepX: A software accelerator for low-power deep learning inference on mobile devices," in *Proc. 15th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2016, p. 23.

[22] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 2, no. 4. Cambridge, MA, USA: MIT Press, 1998.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[24] F. Shrouf, J. Ordieres-Meré, A. García-Sánchez, and M. Ortega-Mier, "Optimizing the production scheduling of a single machine to minimize total energy consumption costs," *J. Cleaner Prod.*, vol. 67, pp. 197–207, Mar. 2014.

[25] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line building energy optimization using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3698–3708, Jul. 2019.

[26] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*. [Online]. Available: http://arxiv.org/abs/1710.05941

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[29] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 274, Dec. 2019.

[30] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.

[31] S. Farahani, "Chapter 6-battery life analysis," in *ZigBee Wireless Networks and Transceivers*. 2008, pp. 207–224.

[32] J. Gozalvez, "5G worldwide developments [mobile radio]," *IEEE Veh. Technol. Mag.*, vol. 12, no. 1, pp. 4–11, Mar. 2017.

[33] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," *J. Power Sources*, vol. 226, pp. 272–288, Mar. 2013.

[34] J. B. Straubel, D. Lyons, E. Berdichevsky, S. Kohn, and R. Teixeira, "Battery pack and method for protecting batteries," U.S. Patent 7 671 565, Mar. 2, 2010.

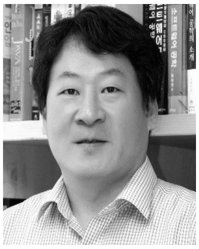
[35] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," 2016, *arXiv:1606.01540*. [Online]. Available: http://arxiv.org/abs/1606.01540

[36] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

AKM ASHIQUZZAMAN received the B.Sc. Diploma (Hons.) degree in computer science and engineering from the University of Asia Pacific, Dhaka, Bangladesh, in 2017. He is currently pursuing the master's degree with the Smart Mobile and Media Computing Laboratory, School of Electronics and Engineering, Chonnam National University, South Korea. His areas of expertise involve QoS / QoE, measuring / management, deep learning, cloud computing, wearable computer, signal processing, and their applications.

HYUNMIN LEE received the M.S. and Ph.D. degrees in electronic engineering from Cheongju University, Cheongju, South Korea, in 2008 and 2012, respectively. He is currently a Researcher with the Korea Electronics Technology Institute (KETI). His research interests include biomedical engineering, cloud computing, wearable device, signal processing, and their applications.

TAI-WON UM received the B.S. degree in electronic and electrical engineering from Hongik University, Seoul, South Korea, in 1999, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2000 and 2006, respectively. From 2006 to 2017, he was a Principal Researcher with the Electronics and Telecommunications Research Institute (ETRI), a leading government institute on information and communications technologies in South Korea. He is currently an Assistant Professor with Duksung Women's University, Seoul. He has been actively participating in standardization meetings, including ITU-T SG 13 (future networks, including mobile, cloud computing, and NGN).



JINSUL KIM received the B.S. degree in computer science from The University of Utah, Salt Lake City, UT, USA, in 2001, and the M.S. and Ph.D. degrees in digital media engineering from the Department of Information and Communications, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2005 and 2008, respectively. He worked as a Researcher with the IPTV Infrastructure Technology Research Laboratory, Broadcasting/Telecommunications Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, from 2005 to 2008. He worked as a Professor with the Korea Nazarene University, Chonan, South Korea, from 2009 to 2011. He is currently a Professor with Chonnam National University, Gwangju, South Korea. His research interests include cloud computing, smart factory/city based application development, and intelligent networking solutions.