# Predicting Battery Levels of Sensor Nodes Using Reinforcement Learning in Harsh Underground Mining Environments

Manish Anand Yadav
Computer Science Department
Missouri University of Science and Technology
Rolla, MO, USA
myzq8@mst.edu

Mohamed Elmahallawy
School of Engineering and Applied Science
Washington State University
Richland, WA, USA
mohamed.elmahallawy@wsu.edu

Sanjay Madria
Computer Science Department
Missouri University of Science and Technology
Rolla, MO, USA
madrias@mst.edu

Samuel Frimpong
Explosive & Mining Engineering Department
Missouri University of Science and Technology
Rolla, MO, USA
frimpong@mst.edu

## Abstract

Underground mining is a hazardous environment, with frequent accidents leading to significant loss of life each year. To enhance safety, sensor nodes monitor key environmental factors such as temperature, toxic gases, and miners' locations, as well as transmit critical messages. Miners interact with these sensors, which track their movements, enabling their location to be determined even without GPS signals. Therefore, predicting the battery life of these sensors is essential for: (i) rerouting miners during emergencies, (ii) ensuring timely maintenance, and most importantly (iii) identifying sensors that need energy harvesting to maintain vital communication within the mine. In this work, we propose a deep reinforcement learning (DRL) approach, *Proximal Policy Optimization-Long Short-Term Memory (PPO-LSTM)*, specifically tailored for the mining environment. This approach considers miners' movements and communication dynamics to predict sensor battery levels, facilitating timely "energy harvesting" for sensors nearing depletion at critical locations within the mine. Our PPO-LSTM framework integrates LSTM networks with PPO to leverage temporal data correlations, enabling better decision-making for energy management. Our extensive simulations demonstrate that the PPO-LSTM framework significantly outperforms current state-of-the-art methods, including *Deep Deterministic Policy Gradient (DDPG)* and *Soft Actor-Critic (SAC)*. Specifically, it achieves improvements of approximately 4%, 1.07%, and 5-10% in Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), respectively.

## Keywords

Wireless sensor networks, underground mines, reinforcement learning, actor-critic methods, proximal policy optimization


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

**ACM Reference Format:**
Manish Anand Yadav, Mohamed Elmahallawy, Sanjay Madria, and Samuel Frimpong. 2018. Predicting Battery Levels of Sensor Nodes Using Reinforcement Learning in Harsh Underground Mining Environments. In . ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

Underground mines have some of the most hazardous working environments; they consist of vast labyrinths spanning many square kilometers, where numerous workers (miners) constantly move for various purposes [1]. Monitoring and ensuring miners' safety face significant challenges due to the harsh operating conditions and communication difficulties in underground mining environments, especially in more complex settings such as deep mining pits or confined underground areas. In mines, sensors collaborate to gather crucial environmental data–alerts about *toxic gases, air quality changes, extreme temperature shifts, and structural instabilities*. Furthermore, the sensors enable precise location tracking [2] of miners to identify their exact positions within the complex tunnel systems without GPS signals. They also facilitate rapid emergency responses during critical incidents like fires or explosions. Altogether, these capabilities provide a robust solution for monitoring and safeguarding miners in the challenging and often hazardous environments of underground mines.

For accurate location prediction, miners need to collect mobility information about both their own movements and those of other miners relative to the pillars [1]. Devices attached to the pillars regularly broadcast their IDs and exchange mobility information, enabling effective tracking within the mine[3]. The miners' locations are then predicted with respect to pillars using these mobility data [4]. Since pillar devices are resource-constrained like Raspberry Pi 5, it is important to estimate their energy consumption and minimize their usage based on the trajectory of miners. The energy consumption of each sensor's battery is influenced by miners' movement within the mine, which is determined by the nature of their tasks. For instance, sensors on the trajectory followed by many miners consume more battery power due to (a) increased broadcast of environmental sensing data, (b) increased number of exchanged messages between them when the miners offload mobility data are passed to these miners (for location prediction task),

[1] In case of disasters, the only way to know miners' location is with respect to pillar ids [3]

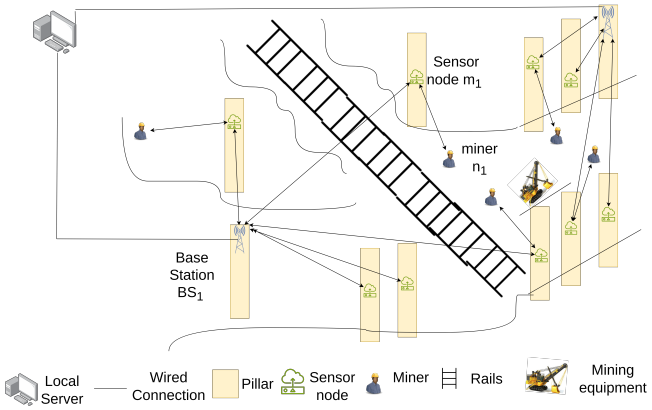



**Figure 1: An illustration depicting the architecture of an underground mine.**

being a part of opportunistic networks [3], while others might be on the trajectory taken by a few and thus consume relatively less energy. Thus, the limited battery capacity of these sensors needs to be predicted in a timely fashion, which helps to reroute miners (in case of a disaster or otherwise) to prolong the life of the sensing nodes and their periodic maintenance, such as replacing batteries, and also in deciding duty-cycles of sensing nodes.

A typical scenario of sensor deployment in underground mining, highlighting various components operating within the mine, is illustrated in Fig. 1. This figure shows multiple miners connecting to a sensor node where they exchange mobility information (direct communication) for location prediction [3] and obtained environmental data like toxic gas concentration, etc. (via periodic broadcast or on-demand) for avoiding dangerous areas within the mine. Each base station (BS) gathers data from nearby sensors using one-hop communication, aggregates this data, and then transmits it to the local server for further analysis, including the prediction of future battery levels of various sensors. This predictive analysis enables more informed decision-making, including *miners' rerouting, sensor node replacement, and identifying key nodes in need of harvested energy*–especially those at critical locations that serve multiple miners.

Unlike outdoor environments, energy harvesting inside underground mines is challenging due to the scarcity of available energy harvesting sources. The few sources from which energy can be harvested—such as light, thermal energy, and radio frequency (RF) signals—are both limited and sporadic [5]. Thus, prioritizing nodes for energy harvesting based on battery-level predictions ensures efficient operations. The work in [6] presents online energy harvesting prediction in WSN to predict future energy availability; this forecasting, in turn, can help exploit the available energy efficiently and make appropriate decisions regarding sensor management. In our work, we focus on battery-level predictions of the nodes for efficient management of harvestable energy. Since energy cannot be harvested for all the sensing devices, it is critical to predict the energy of the critical nodes which serve many miners and are at some critical positions in the mine which helps in case of emergency.

Machine learning algorithms have shown good performance in predicting battery charge in Internet of Things (IoT) devices [7]. However, to predict future battery levels based on spatio-temporal miners' movement inside the mine, we need a method that would allow efficient management of dynamic environments. Reinforcement Learning(RL) gives us that very capability [8]. In [7], RL approaches have been used for joint access control and battery prediction problems in an IoT system. However, traditional RL algorithms, such as Q-learning, excel primarily in small state spaces, where the input information is limited. In contrast, they struggle to converge to an optimal policy in larger state spaces due to the curse of dimensionality. As the state space grows, the number of state-action pairs increases exponentially, making it computationally expensive and time-consuming for these algorithms to explore and learn effectively. To solve this issue, deep reinforcement learning (DRL) was introduced by [9] to leverage neural networks for value function approximation, leading to significant improvements in performance. However, DQN cannot manage continuous action spaces (continuous values between 0 and 100 in our case), and applying it directly to underground mining environments–where precise prediction values of sensors' batteries are critical–is challenging and requires redesign. This is due to its inability to handle continuous action spaces effectively and achieve convergence to an optimal policy.

In this work, we propose a proximal policy optimization (PPO)-based approach that overcomes the above challenges by employing the actor-critic network, where continuous action spaces are handled effectively. Compared to other RL algorithms, the PPO algorithm offers sample efficiency and stable learning, making it more robust for underground mines. Further, we integrate the PPO with a long short-term memory (LSTM) architecture to exploit the temporal correlation in the miners' movement data. Additionally, the clipping mechanism (see Section 4) helps prevent taking vigorous actions and enables converging to the optimal policy. These capabilities equip our framework to effectively address (a) the requirement of precise battery power prediction values, (b) the need for handling large state space and continuous action space, and (c) the requirement of capturing the highly dynamic movement of miners presented by an underground mining environment. In summary, this paper offers the following:

- We formulate a battery-level prediction problem for sensor nodes operating in the underground mining environment as an RL problem. The goal is to enable efficient energy harvesting, facilitate the rerouting of miners, and make timely node replacement decisions for sensors nearing depletion. To the best of our knowledge, this work is the first of its kind to predict the battery levels of sensors in harsh communication and uncertain environments, such as underground mining, based on *miners' movements and data transfer between miners and each sensor.*
- To enhance the reliability and sustainability of sensors in underground mining environments, we (i) employ a centralized agent PPO algorithm. This agent learns patterns across multiple sensors, each with unique miner movement patterns, rather than using multiple agents, reducing the computational costs and improving efficiency; (ii) incorporate an LSTM network architecture to leverage the temporal dynamics of miners' movement data and efficiently learn an optimal policy. This integrated approach enables precise battery level predictions and makes important decisions regarding sensors' replacement, energy-harvesting, and miners' rerouting.
- We conducted a comprehensive evaluation of the performance of our proposed PPO-LSTM approach, achieving a mean absolute

percentage error (MAPE) of less than 10% in environments with 80 sensors. This represents a significant improvement over existing state-of-the-art methods, such as deep deterministic policy gradient (DDPG) and soft actor-critic (SAC) methods, which have a MAPE of up to 14.67%. These results highlight the effectiveness of our approach in accurately predicting sensor battery levels.

## 2 Related Works

Machine learning algorithms have been extensively utilized in the fields of underground mining and wireless networks [10, 11]. In particular, reinforcement learning (RL) algorithms have demonstrated their effectiveness in addressing the challenges posed by highly dynamic wireless environments [12]. While RL algorithms for wireless sensors are common in control and decision-making tasks, their applications in regression/prediction tasks remain limited [13, 14]. Most of the works are focused on resource management; for example, RL algorithms are frequently employed to manage energy harvested from ambient sources to power sensor nodes [8, 15, 16, 17]. However, to the best of our knowledge, there are no works dedicated solely to utilizing RL for predicting battery levels for sensor nodes. For instance, in [8], the authors developed a custom-built energy harvesting wireless sensor network (EHWSN) platform that harvests energy from solar sources within a university building. The sensor nodes are configured using an RL algorithm. The Q-learning algorithm employed adapts the duty cycle of the sensor nodes to avoid node failure (zero-battery condition). Similarly, in [15], an actor-critic algorithm is used to set the packet rates of WSN nodes. Linear function approximation is employed instead of a neural network to make the DRL algorithm "lightweight" and suitable for application in WSNs. However, the authors do not quantitatively verify the computational requirements, and the training of the RL algorithm on the sensors depletes the battery levels at a high rate. In [16], the authors use the state–action–reward–state–action (SARSA) algorithm proposed in [18] to achieve energy-neutral operation (ENO) in solar EHWSNs. In [19], the authors use DQN to adjust the duty cycle of energy harvesting body sensor nodes. The work in [17] develops an RL-based throughput-on-demand (TOD) provisioning scheme for solar-based EHWSNs. The authors in [20] formulated their problem as a multi-objective Markov decision process (MOMDP) rather than an MDP, aiming to maximize multiple task objectives and achieve trade-offs between them. They evaluate their framework on single and dual-task EHWSNs. Another work in [21] proposed a DRL technique, PPO, for an energy management system, leveraging and training it with real-world historical data traces. Similarly, the authors in [22] developed an RL-based framework utilizing the PPO algorithm to manage energy for low-power IoT devices. To enhance its practicality, they optimized the framework for lightweight operation and successfully deployed the algorithm on a wearable IoT device using TensorFlow Lite.

In [23], the authors used the deep deterministic policy gradient (DDPG) algorithm to manage energy in energy-harvesting wireless networks, intending to maximize the average long-term bit rate. In [24], an RL algorithm is introduced for the charger path planning of a mobile charger in a wireless rechargeable sensor network (WRSN). Using the Q-learning algorithm, the authors aim to increase the WRSN's lifetime and improve the MC's efficiency. The work in [25] employs a DRL algorithm, specifically a policy-based algorithm, to determine the optimal charging path for the mobile charger in the WRSN, learning the highly dynamic environment using a combination of pointing and attention mechanisms. Similarly, in [26], the DDPG algorithm is employed to predict building energy consumption.

However, the aforementioned RL approaches cannot be directly applied to our problem due to the large state space, action space and the harsh communication conditions typical of underground mining, where factors such as miners' movements and varying conditions affect battery life. Therefore, a new approach is needed to address these challenges by providing a realistic framework for predicting battery levels and enhancing energy harvesting, which is crucial for broadcasting emergency messages during disasters and potentially saving miners' lives.

## 3 System model & Problem Formulation

### 3.1 System Model

Our system model comprises of four major components: miners' devices, sensor nodes, base station, and local server. A DRL approach is developed to predict sensors' future battery levels which, in turn, enables efficient miners' rerouting, nodes' replacement, and energy harvesting. Fig. 1 illustrates the overall system model of our work. We explain each of the components in detail below:

**Miners' devices:** Miners interact with sensor nodes during their routine tasks to request or obtain necessary information. Their devices, such as tablets, receive crucial safety data–like humidity levels, seismic activity, gas concentrations (e.g., $CO_2$, methane), and oxygen levels–that is broadcast periodically by the sensor nodes. Additionally, these devices can be used to request or exchange information related to miners' movements and other vital data as required.

**Sensor nodes:** In our system model, we consider a system of sensors operating in an underground mine. Each sensor node (like Raspberry Pi 5) is mounted on a pillar with a unique ID and has a communication unit (Bluetooth low energy 5 or LoRa).

**Base station (BS):** The BS (a device like Raspberry Pi-5 or Jetson Nano) facilitates the exchange of data between miners' devices and the local server. It periodically collects data, including the pillar ID, battery level, number of connected miners, and timestamps from each sensor. This information is transmitted to the local server via a wired connection (i.e., an optical fiber cable), which is used for the PPO-LSTM training process. We also incorporate multiple BSs to ensure the whole coverage of all sensors within the mine, thereby enhancing the network's reliability and communication efficiency.

**Local Server:** The local server, typically equipped with hardware capable of training an RL model using data gathered from BSs, is located outside the mine. Once trained, the RL model can predict future battery levels, supporting crucial decisions such as rerouting miners, replacing sensor nodes with low battery levels, and targeting nodes for energy harvesting. Given the challenges of wireless communication–such as signal attenuation, multi-path propagation, interference, and power constraints–a wired connection is used to link the BS with the local server, ensuring reliable data transmission.

### 3.2 Problem Formulation

For any RL algorithm to learn effectively, a careful formulation of the Markov Decision Process (MDP) is essential. This involves defining the states, actions, rewards, and state transitions to accurately capture the core dynamics of the problem [8]. In this study,

we explored various MDP designs through empirical testing before selecting the most suitable design. Our final MDP structure effectively represents the dynamic interactions between miners and sensor nodes, as well as the corresponding changes in battery levels. To solve this MDP, we employ our proposed PPO-LSTM approach, integrated with an LSTM model to improve prediction accuracy and efficiency.

Firstly, we define some notations for the components of our system model. Specifically, we consider a scenario in which there are $N$ miners, $M$ sensor nodes and $N_{\text{BS}}$ base stations. Also, the number of sensor nodes connected to a base station is $M_{bs}$. We denote the set of miners as $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$, the set of sensor nodes as $\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$, and the set of BSs as $\mathcal{BS} = \{bs_1, bs_2, \ldots, bs_{N_{\text{BS}}}\}$.

We then formulate an energy consumption model for each sensor $m_i \in \mathcal{M}$. Using the first-order radio model, we calculate the total energy consumed by a sensor, $E_t(k, d)$, to send an information packet of $k - bit$, expressed as follows:

$$E_t(k,d) = \begin{cases} k(\alpha_{\text{elec}} + \alpha_{\text{short}} d^2), & \text{if } d \le d_{\text{Th}} \le d_{\text{range}} \\ k(\alpha_{\text{elec}} + \alpha_{\text{long}} d^4), & \text{if } d_{\text{Th}} \le d \le d_{\text{range}} \\ \infty, & d_{\text{range}} < d \end{cases} \quad (1)$$

where $d$ is the communication distance, $\alpha_{\text{elec}}$ is the total energy spent to power the transmitter circuit; $\alpha_{\text{short}}$ and $\alpha_{\text{long}}$ are the energy consumption values to transmit 1-bit of data with an acceptable bit error rate (BER) for short and long distances, respectively. The varying characteristics of signal propagation have been considered in this model. Specifically, this energy model uses different amplification energy constants depending on whether the communication is in the short-range (free space) or long-range (multipath) regime. Amplification constant $\alpha_{long}$ can account for the usual phenomenon of reflection, refraction, and diffraction inside the mines. Likewise, the total energy consumed by a receiver, $E_r(k, d)$, to receive a $k$-bit packet is given by:

$$E_r(k,d) = k\alpha_{elec} \quad (2)$$

The Euclidean distance $d_{mn}$, between the sensor node $m$ and miner $n$, can be given as

$$d_{mn} = \sqrt{(X_m - X_n)^2 + (Y_m - Y_n)^2} \quad (3)$$

where $(X_m, Y_m)$ and $(X_n, Y_n)$ are the Cartesian location of sensor node $m$ (identified by the pillar ID) and miner $n$, respectively. Similarly, the Euclidean distance between the sensor node $m$ and a BS $bs$ can be given as:

$$d_{ms} = \sqrt{(X_m - X_{bs})^2 + (Y_m - Y_{bs})^2} \quad (4)$$

where $(X_{bs}, Y_{bs})$ is the Cartesian co-ordinates of a BS $bs$. The distance $d_{\text{Th}} = \sqrt{\frac{\alpha_{\text{short}}}{\alpha_{\text{long}}}}$ is the threshold distance for switching between the short-distance and long-distance energy consumption models, and $d_{\text{range}}$ is the communication range of the sensor node.

For a particular node, the total energy consumption arises from three types of communication: (i) broadcasting messages, (ii) communicating with miners' devices individually, or (iii) communicating with a BS.

Once the communication occurs, the battery levels of the sensor nodes are updated based on their consumed energy. The battery level of each sensor node for the next time-step can be expressed

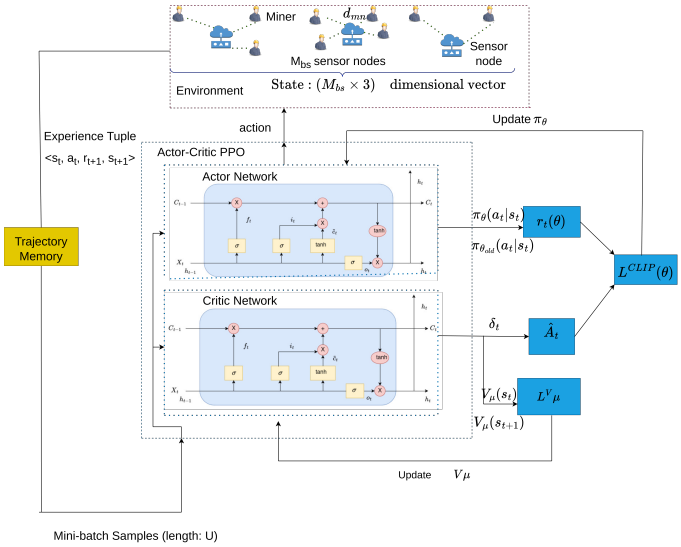



**Figure 2: An illustration of our proposed framework.**

in terms of the energy consumed as:

$$B_m^{t+1} = B_m^t - \frac{E_m^t(k,d) + E_m^r(k,d)}{B^N} \times 100 \quad (5)$$

where $B_m^t$, $E_m^t(k, d)$, and $E_m^r(k, d)$ are the battery level, energy consumed for transmission, energy consumed for reception of a sensor node $m$ at time-step $t$ respectively. $B^N$ is the battery capacity of this sensor node.

Now, we formulate our problem of predicting battery levels of sensors' as an MDP, which consists of five main components, detailed as follows:

- **State space:** For any time step $t$, we formulate the state space as
$$S_t = \{(B_1^a, n_1, \Delta_1)|_t, (B_2^a, n_2, \Delta_2)|_t, \cdots, (B_{M_{bs}}^a, n_{M_{bs}}, \Delta_{M_{bs}})|_t\} \quad (6)$$
where $B_m^a$ denotes the actual battery level of a sensor node $m$, $n_m$ denotes the number of miners communicated with a sensor node $m$, and $\Delta_m$ is the difference between the actual and the predicted battery level values for a sensor node $m$ at time step $t$.
- **Action space:** The action space includes the predicted battery level of all $m$ sensor nodes for the next time-step. Mathematically, the action space can be expressed as
$$A_t = \{(B_1^p|_{t+1}, B_2^p|_{t+1}, \cdots, B_{M_{bs}}^p|_{t+1})\} \quad (7)$$
where $B_m^p$ denotes the predicted battery-level of the sensor node $m$ for time-step $t + 1$.
- **Reward design:** We design a reward function where the RL agent's objective is to reduce the difference between the actual sensor battery level and the predicted value as closely as possible. Therefore, we develop a reward function $R_{t+1}$ based on the absolute difference between the actual battery level $B_m^a$ and the predicted battery level $B_m^p$ of each sensor node $m$, which can be expressed as
$$R_t = \exp\left(-\left(\sum_{m=1}^{M_{bs}} | B_m^a - B_m^p|_t\right)/C\right) \quad (8)$$

**Algorithm 1:** Proposed PPO-LSTM Scheme

**Input:** Initialize policy network $\pi$ with random weights $\theta$, value network $V$ with random weights $\mu$;
**Output:** Policy $\pi_\theta$

```
Initialize a buffer to store trajectories D
for t = 1 to T do
    Collect experiences for a trajectory
    for i = 1 to N do
        Select an action with action noise
        Collect a batch of experiences using current π_θ
        Compute A(s_t, a_t) using value function estimates
        for j = 1 to K do
            Update policy parameters θ using mini-batch
            Compute loss L^CLIP(θ) according to (12)
            Update actor LSTM network parameters θ as
            θ ← θ + α∇_θ L^CLIP(θ)
            Compute the loss of critic network using (15)
            Update critic LSTM network parameters μ as
            μ ← μ − η∇_μ L_μ
        end
        Update action noise
        Update the policy network parameters θ to old
         policy network θ_old
    end
end
```

where $C$ is some positive constant.

- **Transition probability:** We define a matrix $\mathbf{P}$ below that represents the transition probabilities for the $m$-state Markov chain as:

$$\mathbf{P} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0(m-1)} \\ p_{10} & p_{11} & \cdots & p_{1(m-1)} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(m-1)0} & p_{(m-1)1} & \cdots & p_{(m-1)(m-1)} \end{bmatrix} \tag{9}$$

where $m$ represents the number of states of the Markov chain.

- **Discount factor($\gamma$):** To bound the reward, we use a discount factor between 0 and 1, applied at each time step to prevent the accumulated reward from growing infinitely.

## 4 Proposed PPO-LSTM Framework

In this section, we present our proposed PPO-LSTM framework, which incorporates an LSTM model, one of the well-known RNN models, for accurate battery-level prediction for sensor nodes within the challenging environment of underground mining. Our framework operates as follows (Fig. 2 provides a graphical representation of the proposed framework):

1. At the beginning of the training, i.e., at $t = 0$, since the agent has not taken any action yet, the predicted battery level values for each sensor $m$ will be selected randomly to start with. For the subsequent time steps, the action taken by the agent (predicted battery level values for the next time step) will be based on the observed state information, which includes miners' movements and the battery usage of each sensor. Please note that the miners' locations are updated every time step, where a fixed number ($l$) of miners change their position and move from the vicinity of one sensor node to another node (a variable we set in our simulation).
2. Then, the BSs receive information regarding the current battery levels of all sensors $M_{bs}$, along with the number of miners communicating with each sensor $m$. The BSs then transfer this information to the local server.
3. The local server has an RL agent for each BS. Thus, it constructs a state $s_t \in S_t$ of the MDP as provided in Eq.(6) for each BS. This state includes the received readings and the error, which is the difference between the actual and the predicted battery levels for each sensor node.
4. After that, the actor-network of our PPO-LSTM framework, *an LSTM network*, takes the current state information—which is temporally correlated—and outputs a probability distribution. Then, the agent (i.e., the local server) takes an action $a_t \in A_t$ as described in Eq.(7) by sampling from this distribution. This action represents the predicted battery levels of each sensor node for the next time step.
5. After the agent selects an action $a_t$, it receives a scalar reward $R_t$ as specified in Eq. (8), and the environment transitions to the next state $s_{t+1}$. Note, the sample $(s_t, a_t, R_t, s_{t+1})$ is stored in the proximal policy optimization (PPO) trajectory memory.
6. Finally, a mini-batch of samples is used to update the actor and critic network (LSTM network) by minimizing their respective loss functions.

Algorithm 1 provides a formal representation of our proposed framework.

### 4.1 Predicting sensor's Battery Levels Using PPO-LSTM

Given the large size of our state space, traditional RL methods like Q-learning are impractical due to the complexity of updating Q-values in large Q-tables. Instead, we turn to DRL, specifically PPO, for its capability to handle such large state spaces effectively. Our approach focuses on a single-agent PPO-LSTM framework to simplify operations, avoiding the complexities inherent in multi-agent setups, especially the complication in learning caused by the interactions and non-stationarity due to multiple agents [27]. Specifically, each sensor node's state space comprises three key components: current battery level (continuous, ranging from 0 to 100 in our simulation), the number of connected miners with each sensor $m$ (discrete, from 0 to $N$ in our simulation), and $\Delta_m$ (continuous, also ranging from 0 to 100). The continuous nature of these components results in an infinite state space. Similarly, our action space is continuous, predicting battery levels from 0 to 100, which offers enhanced granularity in predictions. Unlike scenarios with known environmental models, our approach is model-free. Therefore, we integrate valuable features from various DRL algorithms: the model-free nature of Q-learning, the ability of DQN to manage large state spaces, and the capability of policy gradient and actor-critic methods to handle continuous action spaces. Specifically, our method employs the PPO algorithm, an on-policy actor-critic DRL approach.

The PPO algorithm addresses a critical issue in actor-critic RL algorithms by mitigating sensitivity to parameter perturbations. Unlike most other actor-critic methods, where small changes in

neural network weights can lead to significant policy alterations, PPO constrains policy updates, ensuring stability. By integrating aspects from various DRL techniques and improving upon trust region methods like trust region policy optimization (TRPO) [28], PPO simplifies implementation and hyperparameter tuning while incorporating a clipping mechanism to prevent overly drastic policy updates. This prevents the agent from converging to sub-optimal policies irreversibly. PPO also enhances sample efficiency through multiple epochs of mini-batch updates.

In our work, we adopt PPO in an actor-critic framework, as outlined in Algorithm 1. The actor network, parameterized by $\theta$, selects optimal action based on current state observation. Actions are sampled from the actor network's probability distribution. Afterward, the agent receives a scalar reward based on the selected action. The critic network, parameterized by $\mu$, evaluates the actions selected by the actor by leveraging obtained rewards. The feedback provided by the critic network is used to optimize the actor network's parameters.

During training, PPO learns from mini-batches sampled from trajectory memory. Basically, multiple iterative updates of the policy and value function parameters are done using gradient descent. Using the current policy $\pi_\theta$, a batch of experiences is collected for each training time-steps. For each epoch, the "advantage function" is calculated to assess the state goodness and informs decision-making, which is given as

$$\hat{A}_t = \sum_{l=0}^{\infty} (\lambda\gamma)^l \delta_{t+l} \tag{10}$$

where $\lambda$ is the generalized advantage estimation parameter, balancing bias and variance in advantage estimation $0 \leq \lambda \leq 1$, and $\delta_t$ is a temporal difference error, which can be given as

$$\delta_t = \gamma V_\mu(s_{t+1}) - V_\mu(s_t) + R_t \tag{11}$$

where $V_\mu(s_t)$ and $V_\mu(s_{t+1})$ represent the state-value functions for achieving the optimal policy in the current state and the next state, respectively. After that, the policy parameter $\theta$ is updated using the clipped objective function as

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[ \min \left( h_t(\theta)\hat{A}_t, \text{clip}(h_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{12}$$

where $h_t(\theta)$ is the ratio of the new policy to the old policy, which can be given as

$$h_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{13}$$

where $\theta_{old}$ is the policy parameters prior to the update. Therefore, the parameter $\theta$ will be updated by gradient ascent as

$$\theta \leftarrow \theta + \alpha \nabla_\theta L^{CLIP}(\theta) \tag{14}$$

Likewise, the critic network loss function $L_V(\mu)$ can be expressed as

$$L_V(\mu) = \mathbb{E}_t[L_t^V(\mu)] = \mathbb{E}[|\hat{V}_\mu^{\text{target}}(s_t) - V_\mu(s_t)|] \tag{15}$$

where $\hat{V}_\mu^{\text{target}}$ is the temporal-difference error, which can be expressed as

$$\hat{V}_\mu^{\text{target}}(s_t) = R_{t+1} + \gamma V_\mu(s_{t+1}) \tag{16}$$

Then, the parameters of critic network $\mu$ are updated by gradient descent as:

$$\mu \leftarrow \mu - \alpha \nabla_\mu L^V(\mu) \tag{17}$$

## 4.2 Utilizing LSTM for Enhancing Battery Level Prediction

Given the temporal structure of miners' movements data, we propose utilizing the LSTM model architecture to capture this information over extended periods. Specifically, while PPO algorithm can learn temporal information by stacking historical state data, determining the optimal number of states to stack depends on the RL environment's dynamics. In our scenario, sensor node behavior changes dynamically with each time-step, impacting the entire environment. This instability over short durations limits available training data, potentially hindering efficient learning. To mitigate this challenge, we employ LSTM networks as both *actor and critic networks* in our approach.

The training process of LSTMs involves backpropagation through time to handle sequential data structures. This method is complex and computationally intensive, potentially resulting in suboptimal performance. However, LSTMs offer significant advantages when data exhibits temporal correlations. Given that miners' movement patterns, device usage, and sensor node battery levels correlate over time, we integrate an LSTM architecture for policy and value function approximation. This approach allows us to capture and utilize these temporal dependencies effectively. The PPO-LSTM architecture is illustrated in Fig. 2.

## 4.3 Complexity Analysis

The computational complexity of the proposed PPO-LSTM algorithm is determined by its nested loops and the operations performed within each iteration. Let $T$ denote the total number of episodes, $N$ the number of trajectories collected per episode, $S$ the average trajectory size, $K$ the number of policy optimization epochs, $M$ the mini-batch size, and $H_\pi$, $H_V$ the hidden state sizes of the actor and critic LSTM networks, respectively.

During each episode, $N$ trajectories are collected, each of length $S$. The computational cost of collecting trajectories includes action selection and value estimation, both of which depend on the LSTM network sizes, resulting in a complexity of:

$$O(N \cdot S \cdot (H_\pi + H_V)).$$

Additionally, policy optimization is performed for $K$ epochs, with a mini-batch size of $M$, leading to a complexity of:

$$O(K \cdot M \cdot (H_\pi + H_V)).$$

Thus, the total computational complexity per episode is:

$$O(N \cdot S \cdot (H_\pi + H_V)) + O(K \cdot M \cdot (H_\pi + H_V)).$$

For $T$ episodes, the overall complexity of the algorithm becomes:

$$O(T \cdot (N \cdot S + K \cdot M) \cdot (H_\pi + H_V)).$$

In practical scenarios, the dominant term depends on the trajectory length $S$, the number of trajectories $N$, and the number of optimization epochs $K$.

# 5 Performance Evaluation

## 5.1 Experimental Setup

We simulate a mine with 84 pillars (with 80 sensor nodes and 4 BSs), each equipped with a sensor node, similar to the setup in [3]. To represent the mine's geometry, we model it as a rectangular grid, distributing sensor nodes and miners randomly at the start of the training. This random placement reflects the uneven spacing

**Table 1: Simulation Parameters**

| Hyperparameter | Symbol | Value |
|---|---|---|
| Total epochs | - | 400 |
| Total time slots per epoch | - | 2048 |
| Number of base stations | - | 4 |
| Mine area | - | 3Km× 3Km |
| Learning rate | $\alpha$ | 0.0003 |
| Discount rate | $\gamma$ | 0.99 |
| Clip range | $\epsilon$ | 0.2 |
| Batch size | - | 128 |
| Reward function constant | C | 100 |
| Number of Sensor nodes | - | 80 |
| Number of Miners | - | 135 |
| Battery Capacity | $B^N$ | 10 J |
| Number of mobile miners | $l$ | 40 |
| Energy constant for transmitter circuit | $\alpha_{elec}$ | 50 $nJ/bit$ |
| Amplification constant for short distance | $\alpha_{short}$ | 10 $pJ/bit/m^2$ |
| Amplification constant for long-distance | $\alpha_{long}$ | 0.0013 $pJ/bit/m^2$ |
| Broadcasting distance | - | 100 m |
| Message size for broadcasting | k1 | 800Kb |
| Message size for communicating with BS | k2 | 100Kb |
| Message size for direct communication | k3 | 300kb |

of pillars typically found in real mines and accounts for variations in layout across different mines. While the sensor nodes remain fixed at the pillars, miners move around and connect to the nearest sensor node. Each node begins with varying initial battery levels.

We also account for scenarios where a miner is within range of multiple nodes and connects to the closest one, optimizing for energy efficiency by reducing communication distance. The miners' movements are modeled based on a fixed number ($l$) moving from one node's vicinity to another at each time step, similar to the approach in [3], and they connect to the nearest node upon arrival. At each time step, a sensor's energy consumption results from three types of communication: broadcasting messages over a certain distance, one-to-one communication with a miner's device, and communication between the node and a BS. We employ four BSs to cover the entire mine; for each BS there is a PPO agent to be trained. Each BS collects data from 20 sensor nodes. To test the robustness of our algorithm, we simulate diverse miner mobility patterns across different BSs.

**Environment and Simulation Parameter.** We implemented our proposed approach, PPO-LSTM, and baselines using Python 3.10.12 and OpenAI Gym 0.25.2, leveraging Google Colab's GPU (NVIDIA Tesla K80 with 12GB of VRAM) for the training process. To ensure efficient learning, we normalized the values of both states and actions. To evaluate the performance of our approach, we analyzed the data generated from agent-environment interactions after the model converged, specifically focusing on time steps between 700,000 and 800,000. The remaining simulation parameters are summarized in Table 1.

**Evaluation Metric.** To evaluate the algorithm's performance, we employ the mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE). We explain them briefly below.

The MAE for each sensor node $m$ is calculated as follows:

$$\text{MAE}_m = \frac{1}{N_s}\sum_{n=1}^{N_s} |B_m^a(t) - B_m^p(t)|, \tag{18}$$

The MAPE of each sensor node $m$ is given as:

$$\text{MAPE}_m = \frac{1}{N_s}\sum_{n=1}^{N_s} \left|\frac{B_m^a(t) - B_m^p(t)}{B_m^a(t)}\right| \times 100\%, \tag{19}$$

The RMSE of each sensor node $m$ is given as:

$$\text{RMSE}_m = \sqrt{\frac{1}{N_s}\sum_{n=1}^{N_s} \left(B_m^a(t) - B_m^p(t)\right)^2} \tag{20}$$

where, in Equations 18, 19 and 20, $N_s$ is the total number of data samples considered for the evaluation, and $B_m^a(t)$ and $B_m^p(t)$ denote the actual ($a$) and the predicated ($p$) battery level of sensor node $m$ at time $t$, respectively.

**Baselines.** We evaluate the PPO-LSTM approach against state-of-the-art methods, including DDPG, Soft Actor-Critic (SAC), and PPO (without the LSTM network). A detailed discussion of the first two approaches is presented below:

- **Deep deterministic policy gradient (DDPG) [23].** The DDPG algorithm is an actor-critic algorithm that uses a *replay buffer* to store past experiences (state, action, reward, next state). It updates the actor (which selects actions) and critic (which evaluates actions) networks by sampling from this buffer, breaking correlations between experiences for more stable training. Target networks for both actor and critic are used to further stabilize the learning process.
- **Soft actor-Critic (SAC) [29].** In contrast to DDPG, the SAC algorithm includes an *additional entropy term* in its optimization objective, alongside the rewards. This entropy term encourages exploration by adding stochasticity to the policy, allowing the agent to explore a wider range of actions. As a result, SAC strikes a balance between maximizing expected rewards and promoting diverse behavior, leading to more effective learning in complex environments.

## 5.2 Results and Discussions

**Comparing with baselines.** We compare our PPO-LSTM approach with our baselines for predicting sensors' battery levels against the actual values. In Table 2, the results show that, on average, our PPO-LSTM approach predicts battery levels more accurately than the DDPG and SAC algorithms. In terms of the MAE, our PPO-LSTM achieves lower values compared to DDPG and SAC among all the BSs, with an *average reduction of five times*.

Moving to the MAPE, PPO-LSTM still achieves the lowest MAPE, as small as 4.39%, compared to a maximum of 14.67% achieved by DDPG and 13.61% of SAC for BS #3, demonstrating the effectiveness of our approach, thanks to the integration of LSTM with PPO. This indicates that our proposed algorithm predicts better when the battery-level values are very small too–the MAPE values provide more weight to the percentage of error, which is magnified when the actual values are small. This is crucial in underground mining scenarios, as timely decisions can be made when the battery level values become small.

**Table 2: Comparison of our proposed PPO-LSTM approach vs. DDPG and SAC and PPO in an 80-Node, 4-BS scenario with respect to MAE, MAPE, and RMSE.**

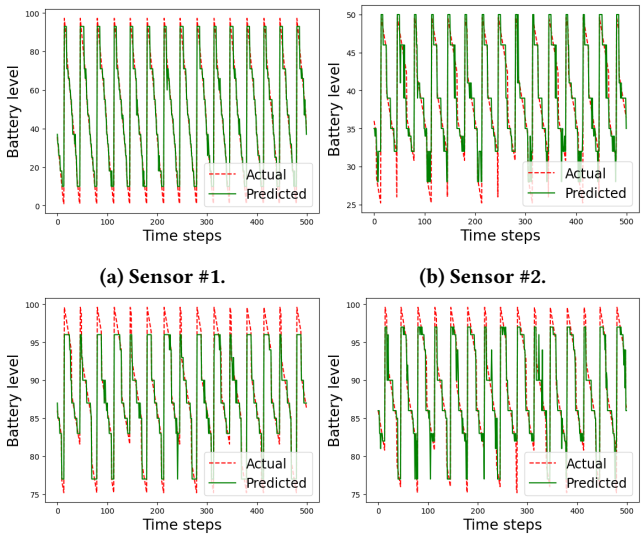
| Base Station# | MAE | | | | MAPE (%) | | | | RMSE | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DDPG | SAC | PPO | **PPO-LSTM** | DDPG | SAC | PPO | **PPO-LSTM** | DDPG | SAC | PPO | **PPO-LSTM** |
| BS #1 | 5.31 | 4.80 | 4.61 | **2.25** | 14.25 | 11.72 | 15.89 | **5.99** | 9.97 | 9.69 | 5.82 | **2.96** |
| BS #2 | 5.30 | 4.82 | 4.30 | **3.78** | 13.15 | 14.06 | 11.79 | **9.02** | 9.93 | 9.68 | 5.42 | **5.68** |
| BS #3 | 5.30 | 4.88 | 3.86 | **1.73** | 14.67 | 13.61 | 10.77 | **4.39** | 9.90 | 9.78 | 4.94 | **2.29** |
| BS #4 | 5.36 | 4.85 | 4.10 | **1.07** | 11.57 | 9.94 | 9.60 | **3.66** | 11.57 | 9.92 | 5.36 | **1.35** |

The PPO algorithm–intergated with LSTM–offers more stable learning than DDPG and SAC, as it is an on-policy algorithm, ensuring that the agent has learned good policies for all ranges of values. In PPO, the policy is directly optimized based on the recent trajectories, compared to DDPG and SAC, which rely on replay buffers and have the possibility of the policy being updated based on outdated data. Thus, our proposed approach to predict granular changes in the battery-level values makes it practical and beneficial for underground mining scenarios.

Lastly, the RMSE results demonstrate that our PPO-LSTM approach outperforms the baseline methods, achieving a significantly lower RMSE of 1.35 compared to 11.57 for DDPG and 9.92 for SAC–an improvement of approximately 11-fold. The low RMSE highlights PPO-LSTM's robustness to large deviations in prediction values, as RMSE penalizes larger deviations more heavily. Across all three evaluation metrics, *PPO-LSTM consistently outperforms the PPO algorithm using fully connected neural networks, further validating its effectiveness.*

**In-depth Evaluation.** In this subsection, we first present additional results evaluating our PPO-LSTM against the baselines in terms of the predicted battery levels versus the actual battery levels across all sensor networks. For simplicity, in Fig. 3, Fig. 4, Fig. 5 and Fig. 6, we display only a subset of sensor nodes (as other nodes exhibit similar trends) and show data for the last 500 time steps. Specifically, in Fig. 3, our proposed PPO-LSTM approach predicts the battery levels close to the actual values in most cases, deviating only during sharp changes (at the pointy ends). In contrast, the DDPG algorithm and SAC algorithm, as illustrated in Fig. 4 and Fig. 5, deviate from the actual values at various points and provide noisy predictions. Finally, as illustrated in Fig. 6, the PPO algorithm without LSTM exhibits significantly greater deviations compared to our PPO-LSTM approach, highlighting the effectiveness and importance of the integration of LSTM network.

The superior performance of our PPO-LSTM framework can be attributed to the clipping mechanism used in PPO (explained in (12)). This clipping mechanism, applied to the probability ratio between the new and old policies, prevents vigorous actions by the agent, thus avoiding convergence to a sub-optimal policy. Furthermore, the incorporation of LSTM for both value function approximation and policy function approximation leverages the temporal nature of the data, contributing to the achievement of a better policy. Specifically, the long-term temporal pattern is preserved by the LSTM (using the memory cell), which helps in taking into account the data of long temporal ranges while generating the output. While DDPG and SAC–both off-policy algorithms–utilize a replay buffer to store and learn from past experiences, PPO, as an on-policy algorithm, does not retain long-term past experiences. As a result, *integrating LSTM with PPO helps leverage long-term temporal patterns, enhancing its ability to learn from sequential data.*



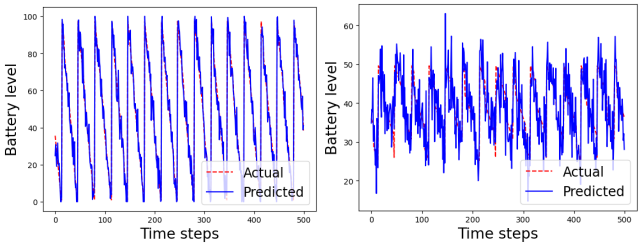

**(a) Sensor #1.** **(b) Sensor #2.**

**(c) Sensor #3.** **(d) Sensor #4.**

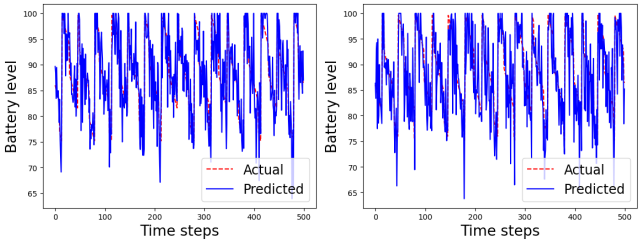
**Figure 3: Predicted battery levels v. actual levels using our PPO-LSTM algorithm.**

Second, we evaluate how average reward values evolve with increasing time steps, along with the rewards achieved by each baseline upon convergence across various BSs. As shown in Fig. 7, our proposed PPO-LSTM approach converges to the optimal policy within 700,000 to 800,000 time steps, attaining average rewards between approximately 0.7-0.8. In contrast, the DDPG algorithm converges much more quickly, within just a few thousand time steps, but results in a sub-optimal policy with an average reward below 0.4. The SAC algorithm, on the other hand, achieves average rewards around 0.5 and requires about 700,000 time steps to converge. Notably, these trends are consistent across all BSs.

Lastly, in Fig. 8, we provide a detailed comparison of the actual battery values against the predicted values among our baseline algorithms. As shown in Fig. 8.c, the results indicate that the PPO-LSTM agent exhibits only slight deviations at most points. In contrast, the DDPG and SAC agents show significantly greater deviations, leading to noisy predictions. Furthermore, during sharp changes, the PPO-LSTM agent maintains a stable output, whereas the DDPG and SAC agents fluctuate, which contributes to a reduction in prediction error.

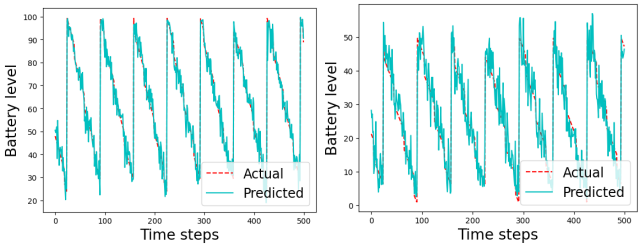



(a) Sensor #1. (b) Sensor #2.

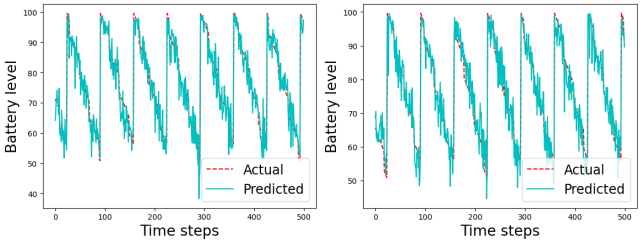



(c) Sensor #3. (d) Sensor #4.

**Figure 4: Predicted battery levels v. actual levels using DDPG algorithm.**

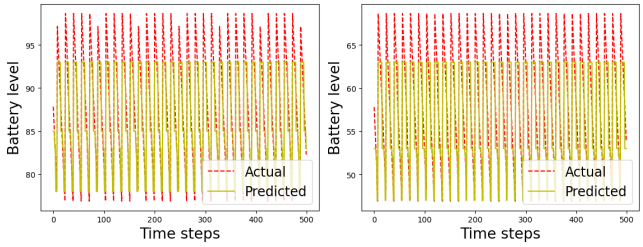



(a) Sensor #1. (b) Sensor #2.

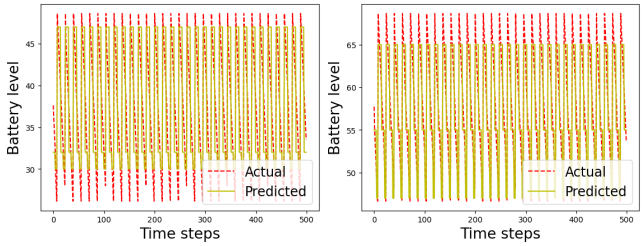



(c) Sensor #3. (d) Sensor #4.

**Figure 5: Predicted battery levels v. actual levels using SAC algorithm.**

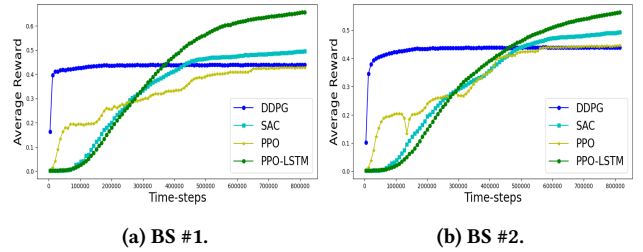



(a) Sensor #1. (b) Sensor #2.

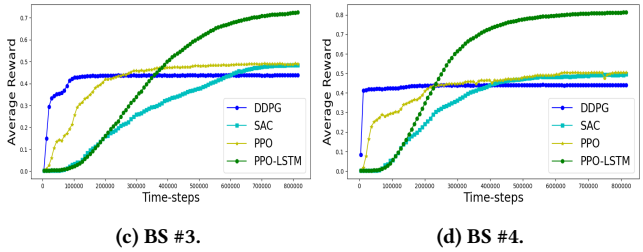



(c) Sensor #3. (d) Sensor #4.

**Figure 6: Predicted battery levels v. actual levels using PPO algorithm.**




(a) BS #1. (b) BS #2.

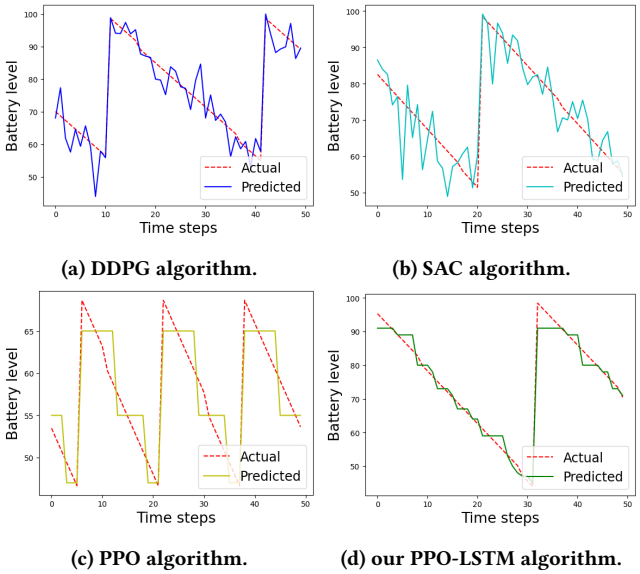



(c) BS #3. (d) BS #4.

**Figure 7: Average reward of our PPO-LSTM algorithm vs. baselines for different BSs in the 80-node scenario.**

## 6 Conclusion

This paper proposes a novel PPO-LSTM framework designed to predict future battery levels of sensors in the challenging underground mining environment by leveraging the movement trajectories of miners. To address computational complexity, we employed a single-agent RL algorithm to model interactions among multiple sensors and miners. Additionally, we integrate an LSTM network to capture dynamic relationships among sensors and miners, where miner movements significantly impact sensor battery levels. Predicting the battery levels supports timely decision-making for rerouting miners, replacing nodes, and implementing energy harvesting strategies. Through extensive simulations, our proposed PPO-LSTM approach demonstrates significant improvements over state-of-the-art methods. Specifically, PPO-LSTM achieves a Mean Absolute Error (MAE) of 1.07, representing a five-fold improvement over existing methods, a Mean Absolute Percentage Error (MAPE) as low as 3.66% in environments with 80 sensors, which is a remarkable four-fold enhancement, and a Root Mean Square Error (RMSE) of 1.35, showing an improvement by a factor of ten compared to




(a) DDPG algorithm. (b) SAC algorithm.

(c) PPO algorithm. (d) our PPO-LSTM algorithm.

**Figure 8: A closer look at the actual vs. predicted values for a sensor node across all models, including our PPO-LSTM algorithm.**

the baselines. These results underscore the effectiveness of our approach in accurately predicting sensor battery levels, thereby enhancing operational efficiency and safety in underground mining environments.

## Acknowledgments

We would like to thank CDC-NIOSH for their support of this research through a grant. Additionally, the authors are grateful to PhD student Abhay Goyal for his valuable contributions in developing ideas for this manuscript.

## References

[1] Fabián Seguel, Pablo Palacios-Játiva, Cesar A Azurdia-Meza, Nicolas Krommenacker, Patrick Charpentier, and Ismael Soto. 2021. Underground mine positioning: a review. *IEEE Sensors Journal*, 22, 6, 4755–4771.

[2] Lalatendu Muduli, Prasanta K Jana, and Devi Prasad Mishra. 2022. Wireless sensor network based miner localization in underground coal mines. In *Advances in Distributed Computing and Machine Learning: Proceedings of ICADCML 2022*. Springer, 121–131.

[3] Abhay Goyal, Sanjay Madria, and Samuel Frimpong. 2024. Minerrouter : effective message routing using contact-graphs and location prediction in underground mine. In *2024 25th IEEE International Conference on Mobile Data Management (MDM)*, 149–158.

[4] Abhay Goyal, Sanjay Madria, and Samuel Frimpong. 2022. Minerfinder: a gae-lstm method for predicting location of miners in underground mines. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 1–12.

[5] Damien Wohwe Sambo and Anna Förster. 2023. Wireless underground sensor networks: a comprehensive survey and tutorial. *ACM Computing Surveys*, 56, 4, 1–44.

[6] Alessandro Cammarano, Chiara Petrioli, and Dora Spenza. 2016. Online energy harvesting prediction in environmentally powered wireless sensor networks. *IEEE Sensors Journal*, 16, 17, 6793–6804.

[7] Man Chu, Hang Li, Xuewen Liao, and Shuguang Cui. 2018. Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in iot systems. *IEEE Internet of Things Journal*, 6, 2, 2009–2020.

[8] Francesco Fraternali, Bharathan Balaji, Yuvraj Agarwal, and Rajesh K Gupta. 2020. Aces: automatic configuration of energy harvesting sensors with reinforcement learning. *ACM Transactions on Sensor Networks (TOSN)*, 16, 4, 1–31.

[9] Volodymyr Mnih et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518, 7540, 529–533.

[10] Mizanur Rahman Jewel, Mohamed Elmahallawy, Sanjay Madria, and Samuel Frimpong. 2024. Dis-mine: instance segmentation for disaster-awareness in poor-light condition in underground mines. (2024). https://arxiv.org/abs/2411.13544 arXiv: 2411.13544 [cs.CV].

[11] Md Sazedur Rahman, Mohamed Elmahallawy, Sanjay Madria, and Samuel Frimpong. 2024. Cav-ad: a robust framework for detection of anomalous data and malicious sensors in cav networks. In *2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*, 330–338. DOI: 10.1109/MASS62177.2024.00051.

[12] Manish Anand Yadav, Yuhui Li, Guangjin Fang, and Bin Shen. 2022. Deep q-network based reinforcement learning for distributed dynamic spectrum access. In *2022 IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI)*, 1–6.

[13] Mohamed Said Frikha, Sonia Mettali Gammar, Abdelkader Lahmadi, and Laurent Andrey. 2021. Reinforcement and deep reinforcement learning for wireless internet of things: a survey. *Computer Communications*, 178, 98–113.

[14] Lei Lei, Yue Tan, Kan Zheng, Shiwen Liu, Kuan Zhang, and Xuemin Shen. 2020. Deep reinforcement learning for autonomous internet of things: model, applications and challenges. *IEEE Communications Surveys & Tutorials*, 22, 3, 1722–1760.

[15] Fayçal Ait Aoudia, Matthieu Gautier, and Olivier Berder. 2018. Rlman: an energy manager based on reinforcement learning for energy harvesting wireless sensor networks. *IEEE Transactions on Green Communications and Networking*, 2, 2, 408–417.

[16] Shaswot Shresthamali, Masaaki Kondo, and Hiroshi Nakamura. 2017. Adaptive power management in solar energy harvesting sensor node using reinforcement learning. *ACM Transactions on Embedded Computing Systems (TECS)*, 16, 5s, 1–21.

[17] Roy Chaoming Hsu, Cheng-Ting Liu, and Hao-Li Wang. 2014. A reinforcement learning-based tod provisioning dynamic power management for sustainable operation of energy harvesting wireless sensor node. *IEEE Transactions on Emerging Topics in Computing*, 2, 2, 181–191.

[18] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[19] Razieh Mohammadi and Zahra Shirmohammadi. 2023. Drdc: deep reinforcement learning based duty cycle for energy harvesting body sensor node. *Energy Reports*, 9, 1707–1719.

[20] Shaswot Shresthamali, Masaaki Kondo, and Hiroshi Nakamura. 2021. Multi-objective reinforcement learning for energy harvesting wireless sensor nodes. In *2021 IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*. IEEE, 98–105.

[21] Francesco Fraternali, Bharathan Balaji, Dhiman Sengupta, Dezhi Hong, and Rajesh K Gupta. 2020. Ember: energy management of batteryless event detection sensors with deep reinforcement learning. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 503–516.

[22] Toygun Basaklar, Yigit Tuncel, and Umit Y Ogras. 2022. Tinyman: lightweight energy manager using reinforcement learning for energy harvesting wearable iot devices.

[23] Chengrun Qiu, Yang Hu, Yan Chen, and Bing Zeng. 2019. Deep deterministic policy gradient (ddpg)-based energy harvesting wireless communications. *IEEE Internet of Things Journal*, 6, 5, 8577–8588.

[24] Zhenchun Wei, Fei Liu, Zengwei Lyu, Xu Ding, Lei Shi, and Chengkai Xia. 2018. Reinforcement learning for a novel mobile charging strategy in wireless rechargeable sensor networks. In *Wireless Algorithms, Systems, and Applications: 13th International Conference, WASA 2018, Tianjin, China, June 20-22, 2018, Proceedings 13*. Springer, 485–496.

[25] Ngoc Bui, Phi Le Nguyen, Viet Anh Nguyen, and Phan Thuan Do. 2022. A deep reinforcement learning-based adaptive charging policy for wrsns. In *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 661–667.

[26] Tao Liu, Chengliang Xu, Yabin Guo, and Huanxin Chen. 2019. A novel deep reinforcement learning based methodology for short-term hvac system energy consumption prediction. *International Journal of Refrigeration*, 107, 39–51.

[27] Donghwan Lee, Niao He, Parameswaran Kamalaruban, and Volkan Cevher. 2020. Optimization for reinforcement learning: from a single agent to cooperative agents. English (US). *IEEE Signal Processing Magazine*, 37, 3, (May 2020), 123–135. Publisher Copyright: © 1991-2012 IEEE.

[28] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[29] Abdullah Alajmi, Waleed Ahsan, Muhammad Fayaz, and Arumugam Nallanathan. 2023. Intelligent resource allocation in backscatter-noma networks: a soft actor critic framework. *IEEE Transactions on Vehicular Technology*, 72, 8, 10119–10132.