

Interpretable Deep Reinforcement Learning for Dynamic Truck Dispatching in Open-Pit Mining

Francisco ROSALES*

ESAN Graduate School Of Business, Lima, Peru
ORCID ID: 0000-0003-2347-632X

*Corresponding author, frosales@esan.edu.pe

Angelo DIAZ

IMCA, Lima, Peru
ORCID ID: 0009-0003-8219-2391
Angelo.diaz@imca.edu.pe

Abstract. Dynamic resource allocation under uncertainty is a core challenge in operations research, particularly in capital-intensive sectors such as open-pit mining, where suboptimal dispatching decisions can lead to substantial productivity losses and increased operational costs. This paper proposes a Deep Reinforcement Learning framework to address the dynamic truck assignment problem in open-pit mining, with a dual focus on performance optimization and policy interpretability. The problem is formulated as a Markov Decision Process (MDP) and implemented within a discrete-event simulation environment that replicates the stochastic dynamics of truck-shovel operations. A Deep Q-Learning approach with neural network function approximation is employed to learn adaptive dispatching policies directly from simulated experience. To enhance transparency, we integrate Explainable Artificial Intelligence (XAI) methods—specifically, Local Interpretable Model-Agnostic Explanations (LIME)—to interpret and validate the learned decision strategies. Experimental results show that the proposed framework significantly outperforms heuristic and random dispatching baselines in terms of resource utilization and operational efficiency. Moreover, the incorporation of explainability provides actionable insights into policy behavior, bridging the gap between autonomous optimization and human-centered decision support in complex industrial systems.

Keywords: Deep Q-Learning, Explainable AI (XAI), Truck Dispatching, Markov Decision Process, Policy Learning, Interpretable Reinforcement Learning.

How to cite: Rosales Marticorena, L. F. Interpretable Deep Reinforcement Learning for Dynamic Truck Dispatching in Open-Pit Mining . *International Conference of Management and Industrial Engineering*, 12. <https://doi.org/10.56177/12icmie2025.202> .

1. Introduction

The assignment of resources to tasks under uncertainty is a central problem in operations research, with broad applications across logistics, transportation, and manufacturing. Efficient allocation decisions can significantly affect system performance, particularly in capital-intensive industries where operational inefficiencies translate directly into financial losses. Among these contexts, open-pit mining stands out as a domain where dynamic resource allocation is both crucial and challenging. The continuous dispatching of haul trucks to shovels must balance multiple, often conflicting, objectives—such as maximizing production throughput, minimizing idle times, and ensuring equipment reliability—while operating under substantial stochastic variability in travel times, loading rates, and machine availability.

This paper addresses the dynamic truck assignment problem in open-pit mining through the lens of Deep Reinforcement Learning (DRL), an approach that has recently shown great promise for sequential decision-making in complex, uncertain environments. Traditional optimization and rule-based dispatching systems, though widely used in practice, often struggle to adapt to real-time fluctuations and nonstationary conditions. In contrast, DRL provides a data-driven framework capable of learning adaptive policies directly from experience, enabling continuous improvement and robust decision-making without explicit modeling of all system dynamics.

We formulate the truck-shovel dispatching task as a Markov Decision Process (MDP) and construct a discrete-event simulation environment that accurately captures the operational dynamics of open-pit mining. Our proposed methodology employs Deep Q-Learning with neural network function approximation to derive optimal dispatching policies from simulated interactions. A key contribution of this work is the integration of Explainable Artificial Intelligence (XAI) methods—specifically, Local Interpretable Model-Agnostic Explanations (LIME)—to interpret and validate the learned policies. This combination allows us to bridge the

gap between highperforming but opaque DRL systems and the practical need for transparency and trust in autonomous decision-making tools used by mine operators.

The application of reinforcement learning to industrial operations builds on a rich body of literature. The foundational work of Bellman [1] established the dynamic programming principles underlying sequential decision-making, while Watkins and Dayan [2] introduced the Q-learning algorithm as a model-free solution to MDPs.

Subsequent advances, notably the introduction of Deep Q-Networks by Mnih et al. [3], enabled reinforcement learning to scale to high-dimensional state spaces. In mining operations, DRL has been increasingly explored for its ability to optimize dispatching and production decisions. For example, Noriega et al. [4] applied Double Deep Q-Learning to improve haulage efficiency while meeting grade control targets, and Huo et al. [5] demonstrated that DRL-based policies can jointly enhance productivity and reduce emissions. Similarly, de Carvalho and Dimitrakopoulos [6], Chiarot et al. [7], and Mukundan et al. [8] have shown that DRL approaches can outperform heuristic dispatching systems under realistic operational constraints.

Despite these advances, a critical gap remains: the interpretability of learned policies in complex industrial settings. Most existing work prioritizes performance metrics, often at the expense of understanding the rationale behind model decisions. Our study explicitly addresses this limitation by embedding explainability into the DRL pipeline. The proposed framework not only learns effective dispatching strategies but also provides interpretable explanations of the underlying decision process, thereby enhancing human comprehension, operational validation, and trust in AI-driven dispatching systems.

The remainder of this paper is structured as follows. Section 2 formalizes the dynamic truck dispatching problem as a Markov Decision Process and details the operational framework of the open-pit mining simulation. Section 3 elaborates on the Deep Q-Learning methodology, including the state and action space design, reward function, and the integration of the LIME framework for policy interpretability. Section 4 presents the discrete-event simulation environment, the stochastic model of ore discovery, and the experimental setup. Section 5 evaluates the results, analyzing the learning process, the emergent dispatching behaviors, and the insights provided by the explainability analysis. Finally, Section 6 concludes by summarizing the key findings and proposing directions for future research, such as incorporating equipment heterogeneity and multi-agent dynamics.

2. Operational Framework

In this section, the truck dispatching problem in open-pit mining is formalized as a sequential decision-making task under uncertainty. The objective is to define the environment, agents, actions, and objective function within a reinforcement learning framework. Of particular importance is the specification of the reward structure, which implicitly defines the agent's goal. Inadequate or misaligned reward formulations may lead the agent to optimize behaviors that diverge from operational objectives—a manifestation of the so-called *alignment problem* in artificial intelligence [9]. Thus, careful problem formulation is essential to ensure that learned dispatching policies are both efficient and operationally meaningful.

Loading and Haulage Processes

In open-pit mining operations, the loading and haulage process constitutes one of the most critical and cost-intensive components, as it directly affects production rates and overall system efficiency. This process involves dynamically assigning haul trucks to loading shovels to transport extracted material—either ore or waste—to various destinations such as crushers, processing plants, or waste dumps.

Each operational cycle consists of four primary stages: (i) *loading at the shovel*, where material is deposited into the truck; (ii) *hauling to the dump or crusher*; (iii) *unloading the material*; and (iv) *returning to a shovel for the next loading*. The dispatching decision—selecting which shovel an available truck should be assigned to after unloading—represents the central optimization problem in this study.

For modeling purposes, all trucks and shovels are assumed to have homogeneous operational characteristics, including identical load capacities and normally distributed loading and unloading times. Although real-world fleets exhibit heterogeneity in size, speed, and cycle durations, this abstraction allows for a controlled analysis of the agent's learning behavior without introducing additional stochastic confounders.

Open-Pit Mining Environment

The open-pit mining environment is modeled as a two-dimensional spatial system with fixed locations for shovels and dumps. Trucks traverse predefined routes between these points, and travel times are determined by distance and velocity. Loading and unloading operations are stochastic, resulting in dynamic queue lengths that evolve over time. Consequently, dispatching decisions exert a direct and nonlinear influence on system performance: inefficient assignments may generate excessive waiting times, congestion, and reduced throughput.

Traditional approaches to this problem have relied on heuristic dispatching rules or mathematical programming formulations. However, such methods often fail to adapt effectively to the high-dimensional, stochastic, and non-stationary nature of real mining operations. To assess the potential of adaptive, data-driven strategies, our simulation environment includes three shovels, four dumps, and ten trucks. At initialization, trucks are evenly distributed across shovels to ensure a balanced starting configuration.

A simplifying yet operationally meaningful assumption is that each shovel follows a predefined material dispatching rule: extracted ore is directed to crushers, waste is sent to waste dumps, and low-grade ore is delivered to stockpile dumps. When 4 multiple crushers are available, ore is routed to the crusher with the lowest cumulative utilization over the current operational period, thereby promoting balanced processing across facilities.

Truck Dispatching Variables

The environment is characterized numerically through feature vectors that encode both system-level and agent-specific information. These representations form the input to a neural network that approximates the action-value function $q(s, a)$.

State Representation

The state space is defined as a composite feature vector integrating both global and agent-level components. The global component includes the queue lengths at each shovel and dump, as well as the normalized remaining time in the simulation, which together describe the system's load and temporal evolution. The agent-specific component encodes, for each truck, its current location (represented as a one-hot vector), remaining travel time, and residual operation time. All continuous variables are normalized by their respective maxima to ensure numerical stability and comparability across features.

Action Space

At each decision epoch, the agent selects an action corresponding to the assignment of an available truck—that is, a truck that has just completed unloading—to one of the shovels that have not yet reached their extraction threshold. This formulation captures the dynamic and constrained nature of the dispatching process while preserving operational feasibility.

Reward Function

The reward structure is designed to promote efficient material transport while penalizing inefficiencies such as excessive queuing and underutilization of resources. Depending on the destination type, the immediate reward at each decision step is defined as:

$$\text{Reward} = \begin{cases} - \text{Queue time,} & \text{for waste dumps,} \\ \text{Low-grade ore} - \text{Trucks in queues,} & \text{for stockpiles,} \\ \text{Pure ore} - \text{Queue time,} & \text{for crushers.} \end{cases} \quad \#(1)$$

This formulation encourages the agent to identify dispatching strategies that maximize overall throughput while maintaining low waiting times and balanced utilization across the system.

3. Deep Reinforcement Learning

Reinforcement Learning (RL) provides a principled framework for optimizing sequential decision-making in dynamic and uncertain environments. In the context of open-pit mining, truck dispatching decisions have both immediate and long-term consequences: each assignment affects not only instantaneous productivity but also future queue lengths and resource availability. This section formalizes the dispatching task as a Markov Decision Process (MDP), introduces the Deep Q-Learning (DQL) algorithm used for policy optimization, and details the architecture and training setup of the neural network that approximates the action-value function.

Truck Dispatching as a Markov Decision Process

Truck dispatching can be modeled as a stochastic sequential decision-making problem, where an agent must select the optimal truck-to-shovel assignment at each decision point based on the current system state. Each decision influences both immediate outcomes—such as waiting time and material throughput—and future system states, including queue lengths and equipment availability. This temporal dependency makes reinforcement learning (RL) a natural framework for optimizing dispatching policies.

Let $\{S_t, A_t\}_{t \in \mathbb{N}_0}$ denote a stochastic process that satisfies the Markov property:

$$\mathbb{P}(S_{t+1} = s_{t+1} \mid S_t = s_t, A_t = a_t, \dots, S_0 = s_0, A_0 = a_0) = \mathbb{P}(S_{t+1} = s_{t+1} \mid S_t = s_t, A_t = a_t), \#(2)$$

where the state and action spaces, S and A , are assumed to be finite. At each time step t , the agent observes the current state $S_t \in S$, selects an action $A_t \in A$, receives a scalar reward R_{t+1} , and transitions to the next state S_{t+1} . The cumulative discounted return is defined as

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \#(3)$$

where $\gamma \in [0,1]$ denotes the discount factor that regulates the trade-off between immediate and future rewards. The agent's behavior is characterized by a policy $\pi(a \mid s)$, which specifies the probability of selecting action a when the system is in state s , i.e.,

$$\pi(a \mid s) = \mathbb{P}(A_t = a \mid S_t = s), \#(4)$$

The objective of the agent is to learn an optimal policy π^* that maximizes the expected return from every state. When the transition dynamics of the environment are known, they can be represented by the conditional probabilities

$$\mathcal{P}_{ss'}^a = \mathbb{P}(S_{t+1} = s' \mid S_t = s, A_t = a), \#(5)$$

which describe the likelihood of transitioning from state s to state s' under action a . Given these dynamics, the state-action value function (or *Q-function*) associated with a policy π is defined as

$$q_{\pi}(s, a) = \mathbb{E}_{\pi}[G_t \mid S_t = s, A_t = a], \#(6)$$

and the expected immediate reward function is expressed as

$$\mathcal{R}_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a], \#(7)$$

The tuple $\langle S, A, \mathcal{P}, \mathcal{R}, \gamma \rangle$ fully characterizes a Markov Decision Process (MDP). In many practical applications, however, the transition probabilities $\mathcal{P}_{ss'}^a$ are unknown. In such cases, value functions can be estimated empirically through simulationbased methods, such as Monte Carlo (MC) estimation, which approximate expected returns from repeated episodes of experience. Regardless of the learning approach, a policy π^* is said to be *optimal* if it satisfies

$$q_{\pi^*}(s, a) \geq q_{\pi}(s, a), \forall s \in S, a \in A, \#(8)$$

for all admissible policies π .

Reinforcement Learning via Deep Q-Learning

In high-dimensional or continuous state spaces, directly representing the action-value function $q(s, a)$ becomes computationally intractable. Deep Q-Learning (DQL) addresses this limitation by using a neural network to approximate the Q-function, enabling the agent to learn effective policies in complex and dynamic environments. The neural network (known as the main or online network) maps state-action pairs to their estimated Q-values and is trained to minimize the discrepancy between predicted and target Q-values using gradient-based optimization.

Formally, the DQL algorithm approximates the value function $q(s, a; w)$ with parameters w , which are iteratively updated using the temporal-difference (TD) learning rule:

$$w_{t+1} = w_t + \alpha(q_s^a - q(s, a; w_t))\nabla_{w_t} q(s, a; w_t), \#(9)$$

where α denotes the learning rate. The TD target q_s^a is defined as

$$q_s^a = \begin{cases} R_{t+1}, & \text{if the episode terminates at } t + 1, \\ R_{t+1} + \gamma q(s', \arg \max_{a' \in A} q(s', a'; w), w'), & \text{otherwise.} \end{cases} \#(10)$$

During training, the agent balances exploration and exploitation through an ε -greedy policy. With probability ε , a random action is selected to encourage exploration of the state-action space; otherwise, the agent exploits its current knowledge by selecting the action that maximizes the estimated Q-value. This mechanism allows the agent to progressively refine its understanding of the environment while avoiding premature convergence to suboptimal policies.

In open-pit mining operations, the environment is inherently stochastic, driven by uncertain loading times, equipment failures, weather conditions, and traffic congestion. Constructing an exact transition model under such uncertainty is impractical. Consequently, a model-free approach such as DQL is particularly suitable: it learns directly from simulated interactions, generalizes across continuous and noisy state representations, and adapts dynamically as operating conditions evolve.

4. Open-Pit Mine Simulation

The open-pit mine simulation serves as a virtual testbed for evaluating reinforcement learning-based dispatching strategies under realistic operational dynamics. It integrates a discrete-event representation of truck-shovel interactions, a stochastic model of ore type discovery reflecting geological uncertainty, and a controlled learning environment for agent training and policy evaluation. This modular structure allows systematic experimentation with varying degrees of system complexity, providing both analytical tractability and the ability to progressively incorporate real-world mining conditions.

Discrete-Event Simulation Framework

The open-pit mine operation is modeled using a discrete-event simulation (DES) framework, where each event represents a temporal transition within the truck loading and haulage process. Events are managed in a chronologically ordered priority queue and processed sequentially, allowing for an efficient and high-fidelity representation of the system's stochastic and dynamic behavior. This design captures the causal dependencies among truck movements, shovel queues, and dump operations, forming a foundation upon which the reinforcement learning (RL) agent interacts with the environment.

Each event is represented as a tuple (time, truck, event_type, position, index), whose components are described in Table 1. This structured formulation ensures that all simulation updates—truck arrivals, departures, and queue transitions—are executed deterministically with respect to simulation time.

Table 1: Attributes defining an event in the discrete-event simulation.

Attribute	Type / Values	Description
time	Real number	Global simulation time at which the event occurs.
truck event_type	Integer identifier ARRIVAL / DEPARTURE	Index of the truck associated with the event. Indicates whether the event corresponds to a truck arrival or departure.
position index	0 = shovel, 1 = dump Integer	Facility type associated with the event. Identifier of the specific shovel or dump (e.g., shovel 0-2, dump 0-3).

Source: Authors' own research

Within this framework, the environment is formalized as a Markov Decision Process (MDP), represented by the tuple (S, A, R, P) . Each element captures a distinct component of the learning problem and defines how the agent perceives and interacts with the simulated mining operation, as summarized in Table 2.

Table 2: Markov Decision Process (MDP) formulation of the truck dispatching environment.

Component	Notation	Description
State space	S	Each state is encoded as a normalized feature vector describing the global system configuration at a given time step, including queue lengths, remaining operation times, and truck positions.
Action space	A	The agent acts only upon processing a DEPARTURE event at a dump, selecting the next shovel destination for the truck. Decisions at shovels follow

		fixed operational policies defined by the mine.
<i>Reward function</i>	<i>R</i>	The immediate reward is defined as the difference between the tons of ore delivered to crushers and a penalty proportional to truck congestion in queues. This structure promotes high throughput while discouraging excessive waiting times.
<i>Transition dynamics</i>	<i>P</i>	The transition kernel is governed by the event simulator, which updates system queues, schedules future events, and advances the simulation clock according to the event sequence.

Source: Authors' own research

The event-driven modeling approach offers both computational efficiency and behavioral realism, enabling detailed evaluation of how dispatching policies influence system-wide performance. Moreover, the modular architecture of the simulation facilitates future extensions—such as incorporating equipment breakdowns, maintenance interventions, or stochastic routing delays—thereby supporting progressive integration of realism without compromising computational tractability.

Stochastic Modeling of Ore Type Discovery

In open-pit mining operations, the composition of the extracted material varies stochastically as excavation progresses through the deposit. Although the general geological structure is characterized through exploration drilling and modeled in the mining plan, the exact sequence and spatial distribution of ore, waste, and low-grade material remain uncertain. This phenomenon, often referred to as the law of the territory, implies that the true mineral content is progressively revealed only as extraction takes place.

To represent this uncertainty in the simulation environment, the material discovery process was modeled as a stochastic process constrained by known proportions derived from the mine plan. Each shovel operates within a designated extraction zone associated with predefined probabilistic distributions of material types—ore, waste, and low-grade ore. At each loading event, the material type is sampled according to these distributions, ensuring that while the overall proportions align with the mine plan, the local extraction sequence remains random.

Formally, for shovel i , the probability of extracting material of type $k \in \{ \text{ore, waste, low-grade} \}$ is given by

$$p_{i,k} = \frac{n_{i,k}}{\sum_{k'} n_{i,k'}}, \quad (11)$$

where $n_{i,k}$ denotes the nominal number of expected extractions of type k assigned to shovel i by the mine plan. This representation captures the dual nature of the mineral discovery process—structured yet uncertain—and allows for variability in operational outcomes across simulation runs.

Table 3 summarizes the assumed extraction probabilities for each shovel. These values can be adjusted to reflect different geological scenarios or planning assumptions.

Table 3: Probabilistic material distributions assigned to each shovel.

Shovel	Ore	Waste	Low-grade Ore
Shovel 1	0.50	0.30	0.20
Shovel 2	0.40	0.40	0.20
Shovel 3	0.35	0.45	0.20

Source: Authors' own research

Learning Process Implementation

To maintain analytical tractability and ensure the stability of the learning process, several simplifying assumptions are introduced in the simulation environment. These assumptions allow the focus to remain on the core aspects of the reinforcement learning framework—namely, the agent's ability to learn adaptive dispatching strategies and the interpretability of its decision-making behavior.

Table 4: Simplifying assumptions adopted in the simulation environment.

Category	Assumption Description
<i>Equipment homogeneity</i>	All trucks and shovels are assumed to have identical loading capacities, speeds, and productivity rates, eliminating heterogeneity effects in performance.
<i>Spatial configuration</i>	The positions of shovels and dumps are fixed in a twodimensional layout, and all routes between facilities are predefined and deterministic.
<i>Operational continuity</i>	Mechanical failures, maintenance downtime, and operatorrelated interruptions are neglected, implying uninterrupted operation during the simulation horizon.
<i>Environmental stability</i>	Exogenous factors such as weather, material properties, and demand fluctuations are considered constant over time.
<i>Perfect information</i>	The agent is assumed to have complete and accurate observations of the system state at each decision epoch.

Source: Authors' own research

While these assumptions substantially reduce system complexity, they also limit the generalizability of the results to real mining operations, where heterogeneity, uncertainty, and partial observability play significant roles. Future extensions of this work will incorporate heterogeneous equipment fleets, stochastic equipment failures, three-dimensional terrain modeling, and non-stationary operational conditions. Integrating these factors will provide a more realistic testbed for evaluating the robustness and scalability of reinforcement learning-based dispatching policies.

Technical Considerations

The Deep Q-Learning agent employs a feedforward neural network to approximate the action-value function $Q(s, a; \theta)$. The architecture comprises an input layer, two hidden layers, and an output layer. The input layer encodes the system state, including queue lengths, remaining operation times, and positional and operational data for each truck.

Two fully connected hidden layers, each with 24 neurons and ReLU activation functions, capture the nonlinear dependencies between system states and their associated Q-values. The output layer contains one neuron per possible action and uses a linear activation function to generate continuous Q-value estimates for each action. The network is trained using the mean squared error (MSE) loss function and optimized via the Adam algorithm, which ensures stable convergence in stochastic environments. The learning rate α is tuned experimentally to balance convergence speed and training stability.

The hyperparameters used during training are summarized in Table 5. These values control key aspects of the learning process, including discounting, exploration, and synchronization between networks, and were selected based on established practices in deep reinforcement learning and sensitivity analyses conducted during preliminary experiments.

Table 5: Summary of DQL training hyperparameters.

Parameter	Symbol / Value	Description
Learning rate	$\alpha = 0.001$	Step size for weight updates during gradient descent.
Discount factor	$\gamma \in [0.90, 0.99]$	Determines the relative importance of future rewards.
Experience replay buffer size	—	Maximum number of stored transitions used for sampling training experiences (no minibatching).
Target network update frequency	Every 1,000 updates	Frequency at which the target network parameters are synchronized with the online network.
Exploration strategy	ε -greedy, $\varepsilon \in [1.0, 0.05]$	Probability of random action selection decays from 1.0 to 0.05 over time.

5. Evaluation and Results

The reinforcement learning framework was empirically evaluated through a series of simulation experiments designed to assess convergence behavior, policy performance, and interpretability of the trained agent. The Deep Q-Learning (DQN) agent exhibited stable convergence throughout training, progressively refining its dispatching strategy and achieving higher cumulative rewards across episodes.

When compared against benchmark policies, including random, greedy, and heuristic strategies based on distance or waiting time, the learned policy consistently demonstrated superior performance in both efficiency and adaptability. These findings indicate that the reinforcement learning agent successfully captured the underlying dynamics of the dispatching environment and was able to derive operationally meaningful strategies without explicit rule encoding. The following subsections present a detailed analysis of the learning dynamics, emergent behaviors, and interpretability of the resulting policy.

Analysis of the Learning Process

The evolution of the agent's performance during training was evaluated through key learning indicators, summarized in Figure 1. These metrics provide quantitative evidence of convergence, policy improvement, and training stability over time.

The first metric, *cumulative reward per episode*, exhibits a consistent upward trend, demonstrating that the agent progressively refines its dispatching policy and achieves higher operational efficiency as learning proceeds. The *loss function* shows a gradual and stable decline, indicating successful optimization of the neural network parameters and improved approximation of the action-value function $Q(s, a; \theta)$. Finally, the *standard deviation of episode rewards* decreases as training advances, suggesting that the learned policy yields increasingly consistent results across different stochastic realizations of the simulation.

Together, these indicators confirm that the reinforcement learning agent attains stable convergence and robust decision-making behavior, effectively capturing the underlying structure of the dispatching problem within the stochastic open-pit mining environment.

Behaviors Learned by the Agent

The performance of the reinforcement learning agent was evaluated through both quantitative metrics and qualitative analysis of its emergent decision-making patterns. Two principal indicators were used to assess the quality of the learned dispatching policy. First, the agent achieved improved compliance with ore and stockpile production targets, reflecting a greater proportion of productive trips and enhanced consistency in material quality. Second, the learned policy yielded a higher average cumulative reward per episode relative to baseline strategies, confirming that the agent effectively optimized long-term operational performance rather than focusing on short-term outcomes.

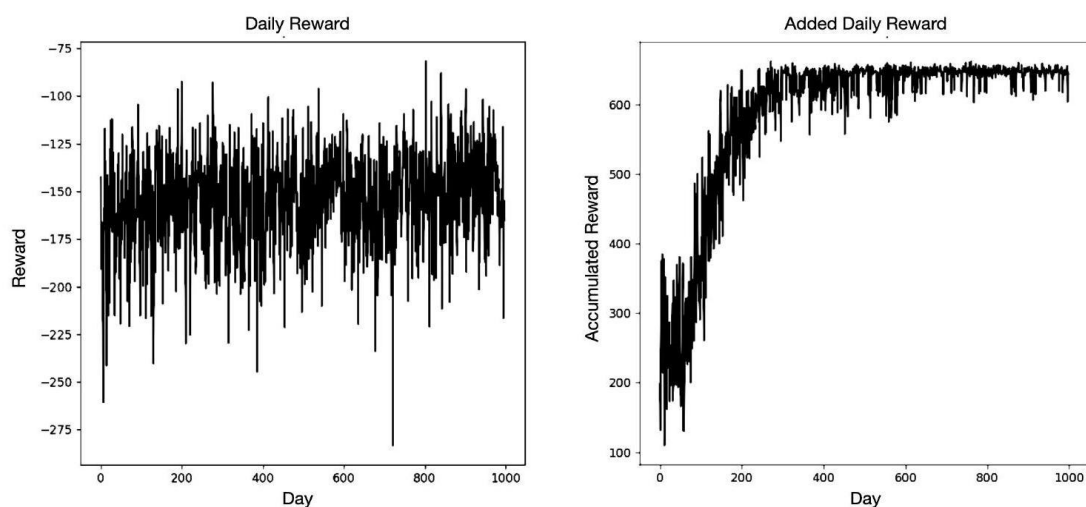


Figure 1: Evolution of the cumulative reward during training. The upward trend indicates progressive improvement in the learned policy.

Source: Author's own contribution

A closer examination of the simulation results reveals several emergent behaviors that characterize the learned dispatching strategy. The agent systematically favored the allocation of trucks to shovels with shorter queues, thereby balancing the operational load and reducing overall idle times. This dynamic load-balancing behavior contrasts with static dispatching rules, where trucks often accumulate at specific loading points, creating inefficiencies. Moreover, the learned policy displayed anticipatory behavior, adjusting decisions in advance to prevent congestion at dumps and along haul routes. This adaptive response maintained operational continuity and improved system throughput, particularly under fluctuating traffic conditions.

Interestingly, the agent also developed implicit operational heuristics that resemble well-established human dispatching practices, despite not being explicitly encoded with such domain knowledge. This finding suggests that the reinforcement learning framework is capable of discovering and internalizing effective operational rules through experience alone. The emergence of these human-like behaviors highlights the potential of reinforcement learning not only to replicate, but in some cases to refine, traditional decision-making processes in open-pit mining operations.

Overall, the results demonstrate that the trained agent converged toward stable and efficient policies that adapt dynamically to system conditions. By learning from simulated experience, the agent was able to generalize across operational contexts, balancing productivity and congestion mitigation—two inherently conflicting objectives in truck dispatching systems.

Interpretable Model Explanations

As machine learning models become increasingly complex, their internal decision processes often lose transparency, making interpretability a central concern—particularly in high-stakes industrial settings where understanding individual predictions is essential for user trust and operational accountability. To address this challenge, we employed the Local Interpretable Model-Agnostic Explanations (LIME) framework, which generates local, human-understandable approximations of a model's behavior.

LIME operates by constructing a simple, interpretable surrogate model that mimics the behavior of the original model in the vicinity of a specific instance. Formally, given a black-box model $f: \mathbb{R}^d \rightarrow \mathbb{R}$ and an instance $x \in \mathbb{R}^d$, LIME seeks an interpretable model $g \in G$ that minimizes the following objective function:

$$\mathcal{L}(f, g, \pi_x) + \Omega(g), \quad (12)$$

where the first term measures the local fidelity of g to f , and the second term penalizes model complexity to ensure interpretability. Specifically,

$$\mathcal{L}(f, g, \pi_x) = \sum_{z_i \in Z} \pi_x(z_i) (f(z_i) - g(z_i))^2, \pi_x(z) = \exp\left(-\frac{D(x, z)^2}{\sigma^2}\right), \quad (13)$$

where $\pi_x(z)$ assigns higher weights to samples z that are closer to x in the feature space, as measured by the distance function $D(\cdot, \cdot)$. The term $\Omega(g)$ regularizes the surrogate model to preserve interpretability, typically favoring sparse linear models.

The resulting coefficients of g quantify the local importance of each feature, thereby providing an interpretable representation of how the deep Q-network (DQN) arrives at its decisions. This approach enables a transparent examination of the learned policy, highlighting the relative influence of environmental states and operational conditions on the agent's actions, and supporting the validation of the model's behavior against domain expertise.

6. Conclusions and Future Work

The results demonstrate that deep reinforcement learning, when integrated with a discrete-event simulation environment, can learn robust, efficient, and adaptive dispatching policies for open-pit mining operations. The agent successfully captured the stochastic and dynamic nature of the system, achieving stable convergence and improving both productivity and consistency compared to heuristic or rule-based strategies. Through experience-driven learning, the model developed behaviors that resemble human decision-making heuristics—such as anticipatory allocation and dynamic load balancing—without explicit programming. These findings highlight the capacity of reinforcement learning to uncover and formalize effective operational strategies in complex, uncertain environments.

Moreover, the interpretability analysis using LIME provided valuable insights into the internal decision mechanisms of the learned policy, enabling a transparent understanding of how environmental states influence the agent's actions. This interpretability is crucial for validating model behavior and fostering trust in industrial applications of AI-driven dispatching systems. In practical terms, the proposed framework can assist mine planners and dispatch engineers in making faster, data-driven allocation decisions, reducing reliance on manual

rules. By optimizing truck–shovel assignments in real time, our approach has the potential to improve fleet utilization and reduce idle times by 10–15%, translating into measurable gains in throughput and operational efficiency.

Future work will focus on extending the modeling framework to incorporate heterogeneity in equipment capacities, stochastic failures, variable haul routes, and evolving environmental conditions. Further research could also explore multi-agent reinforcement learning to capture the decentralized nature of mine operations and investigate explainability techniques tailored to sequential decision-making. These extensions will contribute to the deployment of reinforcement learning as a practical decision-support tool in large-scale mining systems.

References

- [1] R. Bellman, «"A Markovian Decision Process",» *Indiana University Mathematics Journal*, vol. 6, nº 4, pp. 679-684, 1957.
- [2] C. J. C. H. Watkins y P. Dayan, «"Q-learning",» *Machine Learning*, vol. 8, nº 3, p. 279–292, 1992.
- [3] A. Zai y B. Brown, *Deep Reinforcement Learning in Action*, Manning Publications, 2020.
- [4] R. Noriega, Y. Pourrahimian y H. Askari-Nasab, «"Deep Reinforcement Learning based real-time open-pit mining truck dispatching system",» *Computers & Operations Research*, vol. 173, p. 106815, 2025.
- [5] D. Huo, Y. A. Sari y Q. Zhang, «"Smart dispatching for low-carbon mining fleet: A deep reinforcement learning approach",» *Journal of Cleaner Production*, vol. 435, p. 140459, 2024.
- [6] J. P. de Carvalho y R. Dimitrakopoulos, «"Integrating production planning with truck-dispatching decisions through reinforcement learning while managing uncertainty",» *Minerals*, vol. 11, nº 6, p. 587, 2021.
- [7] T. V. Chiarot Villegas, S. F. Segura Altamirano, D. M. Castro Cárdenas, A. M. Sifuentes Montes, L. I. Chaman Cabrera, A. S. Aliaga Zegarra, C. L. Oblitas Vera y J. C. Albán Palacios, «"Improving productivity in mining operations: a deep reinforcement learning model for effective material supply and equipment management",» *Neural Computing and Applications*, vol. 36, nº 9, p. 4523–4535, 2024.
- [8] V. Mukundan y A. Narayan, «"Deep Reinforcement Learning based Truck Dispatcher for Open-pit Mines",» de *6th International Workshop on Multi-Agent Path Finding (WoMAPF 2025)*, 2025.
- [9] B. Christian, *The Alignment Problem: Machine Learning and Human Values*, W. W. Norton & Company, 2020.
- [10] A. Bain, *La ciencia de la educación*, 1st ed., F. Beltrán, Ed., 1915.
- [11] D. Klein, «"Mighty mouse",» *MIT Technology Review*, 19 Diciembre 2018.
- [12] M. Minsky, «"Steps toward Artificial Intelligence",» *Proceedings of the IRE*, vol. 49, nº 1, pp. 8-30, 1961.
- [13] M. Á. Páliza Borja, *Atlas catastral, geológico, minero y metalúrgico*, 1st ed., Instituto Geológico, Minero y Metalúrgico (Ingemmet), 2022.
- [14] M. A. Minaya Cajjak, «"Implementación de la metodología Six Sigma para mejorar la vida útil de neumáticos de camiones mineros en el proyecto Mina Justa",» 2021.
- [15] R. S. Sutton y A. G. Barto, *Reinforcement Learning: An Introduction*, 2da ed., Cambridge, MA.: MIT Press, 2018.
- [16] N. Bäuerle y U. Rieder, *Markov Decision Processes with Applications to Finance*, Springer, 2011.

- [17] H. van Hasselt, A. Guez y D. Silver, «“Deep Reinforcement Learning with Double Q-Learning”»,» de *Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.
- [18] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, Leanpub, 2020.
- [19] A. M. Salih, Z. Raisi-Estabragh, I. B. Galazzo, P. Radeva, S. E. Petersen, K. Lekadir y G. Menegaz, «“A perspective on explainable artificial intelligence methods: SHAP and LIME”»,» *Advanced Intelligent Systems*, vol. 7, nº 1, p. 2400304, 2025.
- [20] A. Gramegna y P. Giudici, «“SHAP and LIME: An evaluation of discriminative power in credit risk”»,» *Frontiers in Artificial Intelligence*, vol. 4, p. 752558, 2021.