Computers & Geosciences 158 (2022) 104962

Contents lists available at ScienceDirect

Computers and Geosciences

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

Updating geostatistically simulated models of mineral deposits in real-time with incoming new information using actor-critic reinforcement learning

Ashish Kumar *, Roussos Dimitrakopoulos

COSMO – Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering, McGill University, FDA Building, 3450 University Street, Montreal, Quebec, H3A 0E8, Canada

ARTICLE INFO

Keywords:
Data assimilation
High-order spatial statistics
Geostatistics
Reinforcement learning
Real-time
Production sensor data
Convolutional neural network

ABSTRACT

The existing technologies that update geostatistically simulated models of mineral deposits cannot self-learn from incoming new information generated in operating mines and do not account for high-order spatial statistics. This work proposes a novel self-learning artificial intelligence algorithm that learns from incoming new information and accounts for high-order spatial statistics, in order to update the geostatistically simulated models of mineral deposits in real-time. The proposed algorithm uses deep policy gradient reinforcement learning with an actor and a critic agent. The grid nodes of the geostatistically simulated model are visited sequentially in a random path, the environment generates the states for each grid node, and feeds the state to the actor and critic agents that respectively predict and evaluate the updated property of the grid node The data is stored in a replay memory, which is sampled at regular intervals to train the agents. The trained agents are then used for further rounds of self-learning. An application of the proposed algorithm at a copper mining operation with incoming drilling machine sensor data (collected spatially), and processing mill sensor data (collected over time), demonstrates its applied aspects in updating the geostatistically simulated models of copper grades of the mineral deposit in real-time, while also reproducing spatial patterns and high-order spatial statistics.

1. Introduction

New information is readily available with conventional and new digital technologies that are used during production activities in industrial environments. These technologies include advanced sensors and monitoring devices that are used during production activities in mines and oilfields, and in monitoring activities in the fields of hydrogeology, hydrology, meteorology, atmospheric sciences, geomorphology and oceanography. For example, in an industrial mining environment, global positioning systems can locate and monitor the status of the mining fleet in real-time (Chaowasakoo et al., 2014). Built-in control units can monitor the health and utilization of the mining fleet (Koellner et al., 2004). Radiofrequency identification tags can locate and track the flow of materials from mines to customer (Rosa et al., 2007), and infrared and X-ray sensors can measure the geological properties of the materials that are mined, hauled, conveyed, processed and sold (Dalm et al., 2018; De Jong, 2004; Goetz et al., 2009; Iyakwari et al., 2016). The incoming new information is typically used to update the relevant properties of geostatistically simulated models (hereafter, simulations). However, this new information, referred to as "soft data", is partial and noisy, and is therefore uncertain. The soft nature of the new information is attributed to the characteristics of the related sensors that generate indirect measurements compared to, for example, those derived from the analysis of drillhole samples in geochemical laboratories. Assimilating incoming new information in simulations is similar to history matching in petroleum reservoirs (Gilman and Ozgen, 2013; Oliver et al., 2008). History matching entails using production data such as oil production, flow rates and well pressure to update simulations of static reservoir properties, such as porosity and permeability, along with dynamic reservoir properties, such as pressure and fluid saturation, to better match the observed production data.

Ensemble Kalman filter (EnKF) is a thoroughly studied and applied method for history matching in petroleum and groundwater reservoirs (Aanonsen et al., 2009; Conjard and Grana, 2021; Oliver and Chen, 2011; Xu et al., 2013). Benndorf (2020, 2015) introduced the use of the EnKF for updating geostatistically estimated models of mineral deposits. Yüksel et al. (2017, 2016) used the EnKF method to update simulations of ash content with incoming new information at a coal deposit.

* Corresponding author. Vale, Digital Transformation Department, 2060 Flavelle Boulevard, Mississauga, Ontario, L5K 1Z9, Canada.
E-mail addresses: ashish.kumar@mail.mcgill.ca (A. Kumar), roussos.dimitrakopoulos@mcgill.ca (R. Dimitrakopoulos).
URL: https://github.com/ashishrokz1993/MineralDepositAICG (A. Kumar).

https://doi.org/10.1016/j.cageo.2021.104962
Received 14 June 2020; Received in revised form 16 September 2021; Accepted 6 October 2021

Wambeke and Benndorf (2017) proposed a combination of the EnKF method with a forward simulator, while incorporating a connected updating cycle and a local neighborhood technique, to update the simulations with the sensor data collected from the conveyor belt at a synthetic mining operation. Wambeke and Benndorf (2018) studied the effect of measurement volumes, blending ratios and sensor precision within the EnKF method. Wambeke et al. (2018) used a forward simulator and Bond's work theorem with the EnKF method to update the simulations of the Bond work index – a geometallurgical property – at the Tropicana gold mine with the processing mill sensor data about throughput, power draw, feed and product size. Other methods for updating simulations of pertinent properties in a mineral deposit include co-simulation with soft data (Journel and Alabert, 1990; Neves et al., 2018) and conditional simulation by successive residuals (Jewbali and Dimitrakopoulos, 2011; Vargas-Guzmán and Dimitrakopoulos, 2002). Methods such as gradual deformation (Hu, 2000), neighborhood algorithm (Sambridge, 1999), evolutionary algorithm (Schulze-Riegert and Ghedan, 2007), maximum a posteriori (Oliver, 1996), Markov chain Monte Carlo (Fu et al., 2017; Oliver et al., 1997), inverse sequential simulation (Xu and Gómez-Hernández, 2015), classification and regression tree algorithm (Gutiérrez-Esparza and Gómez-Hernández, 2017), randomized maximum likelihood (Chen and Oliver, 2012; Sarma et al., 2006; Vo and Durlofsky, 2014), Tau-model (Naraghi and Srinivasan, 2015), Markov mesh model (Panzeri et al., 2016), variants of the ensemble Kalman filter and co-simulation with soft data (Ángel et al., 2021; Journel and Alabert, 1990; Li et al., 2021; Mao and Journel, 1999a; Soares et al., 2017) have also been used for updating simulations of pertinent properties of petroleum and groundwater reservoirs. The above-mentioned methods update the relevant properties of simulations but do not learn from the incoming new information. Learning from new information refers to extracting complex patterns and relationships between new information and simulations, while maintaining this information for future use. Additionally, they do not account for nor respect high-order spatial statistics while updating the simulations.

Recent developments in history matching include artificial intelligence (AI) algorithms based on supervised machine learning, such as a convolutional neural network (CNN) with principal component analysis (PCA) (Liu et al., 2019) and stepwise CNN-PCA with recurrent neural network (RNN) (Tang et al., 2019). The CNN-PCA method trains a CNN to post-process a given PCA geostatistical model, which involves using a training dataset to learn to minimize the difference between the style and content of the generated post-processed PCA model, and the target style and content calculated from either a training image or an initial simulation. The CNN-PCA-RNN trains an RNN to generate flow simulation results for given simulations of porosity and permeability, and involves using a training dataset to learn to minimize the difference between the predictions of the RNN and the targets generated by a high-fidelity flow simulator. These methods aim to minimize the mismatch between the targets and generated outputs for a given training dataset and, therefore, cannot perform well if the inputs differ greatly from the training dataset.

The work presented herein, which is inspired by the continuous control algorithm (Lillicrap et al., 2015), proposes a novel self-learning AI algorithm that trains agents (typically function approximators like neural networks) to update simulations of pertinent properties of mineral deposits in real-time with new incoming information. The proposed algorithm uses deep deterministic policy gradient reinforcement learning with an actor and a critic agent (in this work, both are CNN) to learn about the relationships between incoming new information and simulations. These relationships are defined by high-order spatial statistics. High-order spatial statistics (Dimitrakopoulos et al., 2010; Minniakhmetov et al., 2018; Minniakhmetov and Dimitrakopoulos, 2021; Mustapha and Dimitrakopoulos, 2011; Yao et al., 2021a, 2021b) can capture complex spatial geological characteristics, curvilinear features, geometric relations and the connectivity of extreme values needed for updating spatially dependent geological phenomena. In the following sections, first the proposed self-learning AI algorithm is detailed. Next, an application at a synthetic copper mining operation is explored to illustrate the efficiency and applied aspects of the proposed algorithm. Conclusions and directions for future research follow.

2. Method

Section 2.1 provides the notations used throughout this section. Section 2.2 details how new information is collected during day-to-day production activities in a mining operation that transform raw materials to products. Section 2.2 provides the details of the proposed self-learning AI algorithm. Section 2.3 details the process of using the proposed algorithm in an operating mining environment.

2.1. Notations

Table 1 shows the sets, indices, and constants and Table 2 presents the variables used in the proposed algorithm.

2.2. Incoming new information in a mining operation and related notations

Let $Z(u)$ be a spatial random field with random variables $Z(x)$, representing a property of a mining block at location x, with $x = 1, \ldots, N$ being the index of the blocks. For example, without loss of generality and for simplicity let's assume that the grid shown in Fig. 1(a) represent $Z(u)$. Initial direct measurements, I, derived from the analysis of exploration drillhole samples in geochemical laboratories is denoted by D^{I} as shown in Fig. 1(a). A finite set of initial simulations, $\mathbb{S}^{\mathrm{I}}$, is generated using D^{I}, that represent realizations, $z^{s}(u)$ of $Z(u)$, and quantify the uncertainty about the spatial property of materials in the mine as shown in Fig. 1(b).

The sensors installed on the drilling machines, B, located spatially within the mine, measure the quality of materials drilled. The new information, $N\mathrm{I}$, generated spatially by the sensors on the drilling machine about the property of a block, $Z(x)$, within the mine is denoted by $\mathrm{NI}_B(x)$. The blasted materials are then loaded with shovels, S, into trucks, T. The sensors on the shovels measure the quantity, $q^i(x)$, and quality, $\mathrm{NI}_i(x)$, $\forall i \in S$, of a block, $Z(x)$, loaded. The sensors on the truck measure the quantity, $q^i(x)$, and quality, $\mathrm{NI}_i(x)$, $\forall i \in T$, of $Z(x)$ hauled. The incoming new information about the quality of blocks $Z(x)$ located spatially within the mine is herein referred to as "spatial sensor data" and denoted by $\mathscr{S} \in B \cup S \cup T$ as shown in Fig. 1(c). The trucks haul the materials to different destinations, D as shown in Fig. 1(d). Let f^d, $\forall d \in D$ represent a function that mimics the transformation of materials at destination d. The sensors at the destinations measure the quality $\mathrm{NI}_d(q^d)$, $\forall d \in D$, and quantity, q^d, of materials at destination d. For example, a destination in a mining operation can be crusher as shown in Fig. 1(e) which has sensors to monitor the quality and quantity of materials. The materials from the destinations are transported via conveyor belts, C, to processing streams, P. The conveyor belt analyzer monitors the rate, $q^{d,p,c}$, and quality, $\mathrm{NI}_c(q^{d,p,c})$, $\forall d \in D$, $p \in P, c \in C$, of material transported via conveyor c from destination d to processing stream p. For example, the materials from crusher is transported to processing mill via a conveyor belt and analyzed via a conveyor belt analyzer as shown in Fig. 1(f). The processing streams generate the products which are sold to customers as shown in Fig. 1(g). Let f^p, $\forall p \in P$ denote the function that mimics the transformation of materials at processing stream p. The sensors at the processing stream measure the quality, $\mathrm{NI}_p(q^p)$, and quantity, q^p, $\forall p \in P$, of products generated as shown in Fig. 1(g). Let $\mathscr{T}$ represent a set that consists of all the components in a mining operation that handle and process the materials and collect sensor data, i.e. $\mathscr{T} = \{D, C, P\}$. Let $\mathrm{NI}_i(q^i)$ and q^i represent the incoming new information collected over time with sensors at component $i \in \mathscr{T}$ about the quality and quantity of related materials, referred to herein as "temporal sensor data". For

Table 1
Sets, indices, and constants used in the proposed algorithm.

Parameters	Definition
$Z(u)$	Spatial random field consisting of random variables $Z(x)$, $\forall x \in [1, N]$
$Z(x)$	Random variable representing a property of a mining block at location $x, \forall x \in [1, N]$ within the mine
D^{I}	Initial (I) drillhole (D) samples at the mine
$\mathbb{S}^{\mathrm{I}}$	Set of initial simulations generated usingD^{I} for all blocks in $Z(x)$
$Z^{\mathrm{s}}(x)$	Simulated property of a block located at x in $\mathrm{s} \in \mathbb{S}^{\mathrm{I}}$
$\mathbb{S}^{\mathrm{U}}$	Set of updated simulations for all blocks within the mine; $\mathrm{s}' \in \mathbb{S}^{\mathrm{U}}$
$Z^{\mathrm{s}'}(x)$	Simulated property of a block located at x in $\mathrm{s}' \in \mathbb{S}^{\mathrm{U}}$
B, T, S, D, C, P	Set of blasthole drilling machines (B), trucks (T), shovels (S), destinations (D), conveyor belts (C), and processing streams (P), respectively in a mining operation
$\mathrm{NI}_B(x)$	New information (NI) generated spatially by sensors located on B about the quality of material drilled
$q^i(x)$, $\mathrm{NI}_i(x)$	New information generated spatially about the quantity and quality of materials, respectively at component $i \in S \cup T$
$\mathscr{S}$	Set of sensors that generate spatial new information, i.e. $\mathscr{S} \in B \cup S \cup T$
q^d, $\mathrm{NI}_d(q^d)$	Temporal new information generated by the sensor at destination $d \in D$ about the quantity and quality of materials, respectively
f^d	A function that mimics the transformation of materials at destination $d \in D$
$q^{d,p,c}$, $\mathrm{NI}_c(q^{d,p,c})$	Temporal new information generated by the sensor at conveyor belt $c \in C$ about the quantity and quality of materials, respectively transported from destination $d \in D$ to processing stream $p \in P$
q^p, $\mathrm{NI}_p(q^p)$	Temporal new information generated by the sensor at processing stream $p \in P$ about the quantity and quality of materials, respectively
f^p	A function that mimics the transformation of materials at processing stream $p \in P$
$\mathscr{T}$	Set of sensors that generate temporal new information, i.e. $\mathscr{T} \in D \cup C \cup P$
Track	Sensors that tracks the flow of materials from mine to customer
$Neigh_x$	The neighborhood of a mining block located at x
$Z^{\mathrm{s}}(Neigh_x)$	Simulated property of all the block in s inside $Neigh_x$
$Z^{\mathrm{s}'}(Neigh_x)$	Simulated property of all the block in s' inside $Neigh_x$
$D_x^{\mathrm{s}'}$	Conditioning data event value for a block located at x in s'
N^{s}	Number of conditioning values in the data event in $\mathrm{s} \in \mathbb{S}^{\mathrm{I}}$
N_ζ^{s}	Number of replicates for data event $D_x^{\mathrm{s}'}$ found in s within $Z^{\mathrm{s}}(Neigh_x)$
$\zeta_{\mathrm{s},i,j}$	Value of node $j \in N^{\mathrm{s}'}$ in the replicate $i \in N_\zeta^{\mathrm{s}}$
$H_x^{\mathrm{s}'}$	The geometry of conditioning data event $D_x^{\mathrm{s}'}$ defined by a normalized three-dimensional distance vector found in $\mathrm{s} \in \mathbb{S}^{\mathrm{I}}$
$\mathrm{NI}_{\mathscr{S}}(Neigh_x)$	Spatial sensor data collected inside $Neigh_x$
$D_x^{\mathscr{S}}$	Conditioning data event value for block located at x in $\mathscr{S}$
$N^{\mathscr{S}}$	Number of conditioning values in the data event in $\mathscr{S}$
$N_\zeta^{\mathscr{S}}$	Number of replicates for data event $D_x^{\mathscr{S}}$ found in $\mathscr{S}$ within $\mathrm{NI}_{\mathscr{S}}(Neigh_x)$
$\zeta_{\mathscr{S},i,j}$	Value of node $j \in N^{\mathscr{S}}$ in the replicate $i \in N_\zeta^{\mathscr{S}}$
$H_x^{\mathscr{S}}$	The geometry of conditioning data event $D_x^{\mathscr{S}}$ defined by a normalized three-dimensional distance vector found in the $\mathscr{S}$
$V(z^{\mathrm{s}}(x))$	The conditional variance of a block located at x computed over $\mathbb{S}^{\mathrm{I}}$
E_x^{s}	Average conditional variance for a block located at x over $\mathrm{s} \in \mathbb{S}^{\mathrm{I}}$ inside $Neigh_x$
γ_{s}	Adjustment factor to adjust the magnitude of E_x^{s}
$E(\mathrm{NI}_{\mathscr{S}}(x))$	The error of spatial sensor data for a block located at x
$E_x^{\mathrm{NI}_{\mathscr{S}}}$	Average error with the spatial sensor data in the neighborhood $Neigh_x$ of a block located at x
E^{NI_i}	Error in the new information collected from the component $i \in D \cup C \cup P$ in a mining operation
W	Legendre series polynomial order for the approximation of conditional probability distribution function
τ	The soft target update parameter
	Batch size

Table 1 (*continued*)

Parameters	Definition
N_{BS}	
γ	Discount factor
N_R	Replay memory cache size
N_I	Training interval
γ_i^{MP}	The adjxustment factor for adjusting the magnitude of model-based prediction error
c	L2 regularization cost
N_U	Update training iterations
N_{TE}	Number of training episodes
N_T	Number of training iterations
N_{UE}	Number of update training episodes

Table 2
Variables used in the proposed algorithm.

Variable	Definition
f_{θ^μ}	Actor agent (a CNN) parameterized by θ^μ
f_{θ^Q}	Critic agent (a CNN) parametrized by θ^Q
s_t	State at time t
a_t	Action proposed by actor agent at time t
r_t	Reward computed by the environment at time t
$\mathscr{N}_t$	Random noise process added to the actions at time t, for exploration during training
f'_{θ^Q}	Target critic CNN agent parameterized by $\theta^{Q'}$
f'_{θ^μ}	Target actor CNN agent parameterized by $\theta^{\mu'}$

example, in Fig. 1 the temporal sensor data will be generated from a crusher, a conveyor belt and a processing mill. The tracking devices installed on component $i \in \mathscr{T} \cup \mathscr{S}$ of the mining operation, help to locate and track the flow of materials. Let *Track* represent an operator that can locate and track the flow of materials from the mine to the customers. For example, the materials flowing from mine to customer in Fig. 1(d–g) is tracked using the RFID tags in the blastholes and GPS installed on trucks. *Track* operator for this example will represent the data generated by RFID sensors and GPS about the location of materials while it flows from mine to customer.

2.3. A self-learning AI algorithm

The proposed self-learning AI algorithm for updating the simulations of pertinent properties of mineral deposits with incoming new information uses deep deterministic policy gradient (DDPG) reinforcement learning (Lillicrap et al., 2015) with an actor and a critic agent (in this work, both are CNN). The actor, μ, and critic, Q, agents (f_{θ^μ}, and f_{θ^Q} parametrized by θ^μ, and θ^Q, respectively) interact with an environment (see Sect. 2.3.1 for details of the environment) in discrete timesteps. Herein, a time step t denotes the point at which a block is visited along a random path visiting all blocks, similar to the sequential simulation approach (Deutsch and Journel, 1992; Gómez-Hernández and Srivastava, 2021; Journel, 1994). At each time step t the actor takes an action $a_t \in \mathbb{R}$ which is to predict the updated property $Z^{\mathrm{s}'}(x)$ of a block located at x based on a state, s_t – which is fully observable – for a given simulation $\mathrm{s} \in \mathbb{S}^{\mathrm{I}}$. The action is executed in an environment, meaning that the block property, $Z^{\mathrm{s}}(x)$, is updated with the taken action a_t to generate an updated simulation $\mathrm{s}' \in \mathbb{S}^{\mathrm{U}}$ and the agent receives a scalar reward r_t and a next state s_{t+1} (see Sect. 2.3.1 for calculation of r_t and s_{t+1}). The process continues until all the blocks are visited and updated. The state, action, reward, next state tuple (s_t a_t, r_t, and s_{t+1} respectively) from all the time steps is stored in a replay memory buffer, R, of finite-sized cache. The replay memory is sampled uniformly at regular intervals to train both the actor and critic agents (see Sect. 2.3.3 for the training of

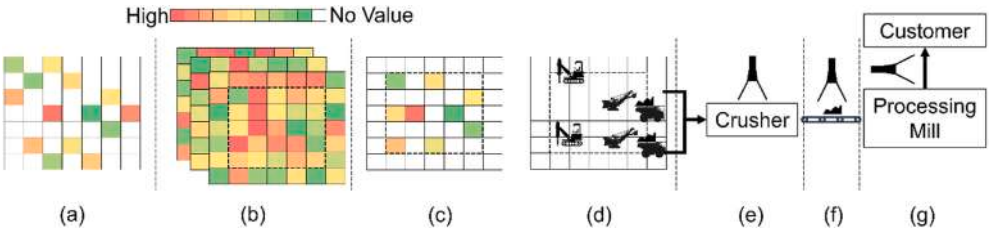

Fig. 1. (a) Exploration drillhole data; (b) set of initial simulations generated using the exploration drillhole data; (c) spatial sensor data collected via sensors installed on drilling machines, shovels and trucks; (d) GPS and RFID tags help to track which blocks are extracted from the mine and sent to crusher; (e) sensor at the crusher measure quality and quantity of crushed materials; (f) sensors on conveyor belt measure rate and quality of conveyed materials, and (g) sensor at the processing mill measure the quality and quantity of processed materials.

actor and critic agents). The algorithm terminates when N_T iterations are reached, and the trained actor agent f_{θ^μ} can be used to update the simulations $\mathbb{S}^{\mathrm{U}}$, of pertinent properties of mineral deposits with the new information collected during production activities in a mining operation as detailed in Sect. 2.4.

2.3.1. Environment

In the context of the present study, an environment is a model of the mining operation that encapsulates how materials are extracted from the mines are transformed from raw materials to products with production activities. The agents interact with the environment by visiting mining blocks in a random path. The environment provides a representation of the mining operation during its interaction with the agents. The representation is called a state, and includes information such as the property of blocks in initial simulations, conditioning data events and geometry, new information, conditional variance of the blocks in the initial simulations, error in the new information, and the model-based predictions. The environment is also responsible for evaluating and using the action a_t taken by the agent to generate a scalar value r_t, called a reward, and a new representation, called the next state s_{t+1}. The calculation of the state, reward, and next state in the environment are detailed next.

2.3.1.1. State. The state s_t generated by the environment is comprised of 10 components. The first component is the property of the blocks in the initial simulation $\mathbb{s} \in \mathbb{S}^{\mathrm{I}}$ including and surrounding the block located at x in consideration at time step t denoted by $Z^{\mathbb{s}}(Neigh_x)$. Here, $Neigh_x$ defines the neighborhood to consider around a block located at x, and is an input for the algorithm. The missing blocks in $Neigh_x$ are initialized to -1. For example, the neighborhood for the blocks close to the edges of the simulation grid will go outside the grid and, thus, these blocks are called missing blocks and are initialized to -1. The second component of the state is the conditioning data event $D_x^{\mathbb{s}'}$ which includes a property of $N^{\mathbb{s}}$ closest blocks for a block x in $\mathbb{s}' \in \mathbb{S}^{\mathrm{U}}$. $\mathbb{s}'$ is formed by first creating a copy of $\mathbb{s} \in \mathbb{S}^{\mathrm{I}}$ and then updating the simulated property of blocks that have been visited until time $t-1$ with the actions taken by the actor agent. This captures the history of actions taken until $t-1$. The quantity of conditioning data $N^{\mathbb{s}}$ is an input for the algorithm. Let's consider a simple example as shown in Fig. 2. Suppose $Z^{\mathbb{s}}(Neigh_x)$ and x are respectively represented by the dashed box and black circle in Fig. 2(a), then the ends of black arrows in Fig. 2(b) constitutes $D_x^{\mathbb{s}'}$.

The third component of the state is the incoming spatial new information $\mathscr{S}$ including and surrounding a block located at x, denoted by $\mathrm{NI}_{\mathscr{S}}(Neigh_x)$. The absent values in the spatial new data are initialized to -1. The fourth component of the state is the conditioning data event $D_x^{\mathscr{S}}$ found in the spatial sensor data which includes the $N^{\mathscr{S}}$ closest spatial sensors' data. Suppose $\mathrm{NI}_{\mathscr{S}}(Neigh_x)$ and x are respectively represented by the dashed box and the black circle in Fig. 2(c), then the ends of black arrows in Fig. 2(c) constitutes $D_x^{\mathscr{S}}$. The quantity of conditioning data $N^{\mathscr{S}}$ is computed based on the density of the spatial sensor data. The fifth component of the state is the geometry of the conditioning data events $D_x^{\mathbb{s}'}$ and $D_x^{\mathscr{S}}$,defined by normalized distance vectors $H_x^{\mathbb{s}'}$ and $H_x^{\mathscr{S}}$, respectively (the distance of the conditioning data point from a block located at x). The sixth component of the state is the average conditional variance $E_x^{\mathbb{s}}$ associated with a property of a block located at x in $\mathbb{s} \in \mathbb{S}^{\mathrm{I}}$ inside $Neigh_x$. $E_x^{\mathbb{s}}$ is calculated as:

$$E_x^{\mathbb{s}} = \frac{\gamma_{\mathbb{s}}}{|Neigh_x|} \sum_{i \in Neigh_x} V(Z^{\mathbb{s}}(x_i)),\ \forall \mathbb{s} \in \mathbb{S}^{\mathrm{I}} \tag{1}$$

where, $V(Z^{\mathbb{s}}(x_i))$ is the conditional variance of the initial simulated property of a block x_i and γ_s is an adjustment factor to adjust the magnitude of the conditional variance of the simulations. The seventh component of the state is the error in the spatial sensor data $E_x^{\mathrm{NI}_{\mathscr{S}}}$ collected in $Neigh_x$ of a block located at x. The error with each of the spatial sensor data is an input to the algorithm. $E_x^{\mathrm{NI}_{\mathscr{S}}}$ is computed by averaging the error in the available spatial sensor data in $Neigh_x$ as:

$$E_x^{\mathrm{NI}_{\mathscr{S}}} = \frac{1}{|Neigh_x|} \sum_{i \in Neigh_x} E(\mathrm{NI}_{\mathscr{S}}(x_i)) \tag{2}$$

The eighth component of the state is the incoming temporal sensor data, $q^i, \mathrm{NI}_i(q^i),\ \forall i \in \mathscr{T}$. The ninth component of the state is the errors in the temporal sensor data, $E^{\mathrm{NI}_i}, \forall i \in \mathscr{T}$, and is an input to the algorithm. Let's consider a simple example as shown in Fig. 3 where mining blocks 11 and 39 are entirely extracted on day 1 and blocks 20 and 34 are entirely extracted on day 2 The extracted materials are transported to

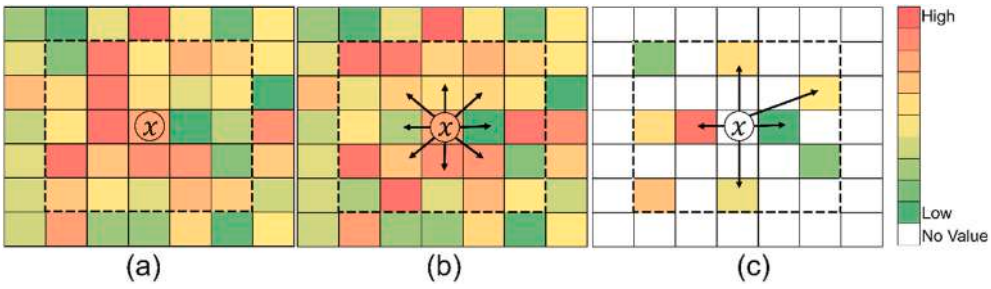

Fig. 2. State representation of (a) a property of the blocks in the neighborhood $Neigh_x$ of a block located at x in consideration at time step t in the initial simulation $\mathbb{s} \in \mathbb{S}^{\mathrm{I}}$; (b) conditioning data event found in the simulation $\mathbb{s}' \in \mathbb{S}^{\mathrm{U}}$ until $t-1$; and (c) spatial sensor data with its conditioning data event.

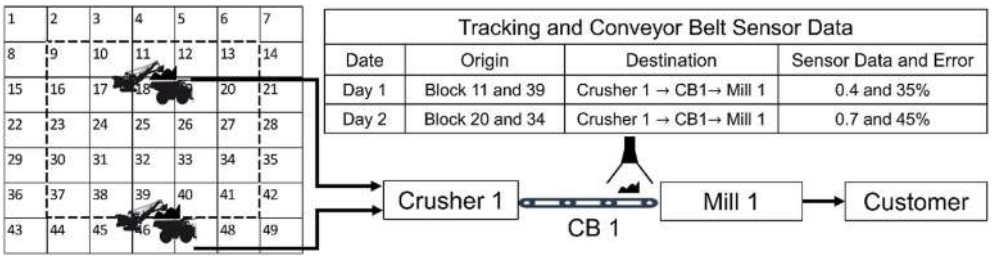

Fig. 3. State representation of incoming temporal sensor data and its associated errors in measurements.

crusher 1 and thereafter to the mill 1. The GPS on the truck and the RFID tags in the blastholes of those blocks will help to track the material by generating origin and destination data as shown in Fig. 3. To ensure that tracking of materials from mining blocks to destinations data is appropriate and useful, a mining operation needs to ensure that issues related to dilution and selective mining units are properly addressed as discussed in Parker (2012). The sensor installed on the conveyor belt CB1 measure the grade of materials passing on the conveyor belt along with an associated error for such measurements. The tenth component of the state is the model-based prediction for the updated simulation $\mathbb{s}' \in \mathbb{S}^{\mathrm{U}}$ at the location of the different temporal sensor data (see Sect. 2.3.1.2 shows details of how model-based prediction is generated). Calculating the model-based prediction for the updated simulation captures the history of the actions taken until time step $t-1$. If there are no new temporal sensor data, then such data along with their model-based predictions are initialized to -1.

2.3.1.2. Model-based predictions. Model-based predictions, MP, calculate the values that should have been observed based on simulations at the location where temporal sensor data was collected. The model-based prediction $\mathrm{MP}_d^{\mathbb{s}}$ at a destination d for a simulation $\mathbb{s}$ is calculated with Eq (3).

$$\mathrm{MP}_d^{\mathbb{s}} = f^d\left(\sum_{i\in T} Track(q_i(x); Z^{\mathbb{s}}(x))\right),\ \forall \mathbb{s}\in\mathbb{S}^{\mathrm{I}}, d\in D \tag{3}$$

Equation (3) uses the *Track* operator first to find the quantities of materials, $q_i(x), \forall i \in T$, hauled with trucks $i \in T$ to a destination d then utilizes the function f^d to calculate how the materials are transformed at the destination, and finally uses the simulated property $Z^{\mathbb{s}}(x)$ along with the tracking and transformation data to compute $\mathrm{MP}_d^{\mathbb{s}}$. The model-based prediction $\mathrm{MP}_c^{\mathbb{s}}$ for a simulation $\mathbb{s}$ at a conveyor belt c which transports materials from a destination d to a processing stream p is calculated as:

$$\mathrm{MP}_c^{\mathbb{s}} = Track\left(MP_d^{\mathbb{s}}\right)\mathbb{s}q^{d,p,c},\ \forall \mathbb{s}\in\mathbb{S}^{\mathrm{I}},\ d\in D,\ p\in P,\ c\in C \tag{4}$$

Equation (4) first uses the *Track* operator to find the quantity of material $q^{d,p,c}$ flowing from a destination d to a processing stream p with conveyor belt c and then utilizes the model-based prediction from Eq. (3) along with the tracking information to compute $\mathrm{MP}_c^{\mathbb{s}}$. For example, in Fig. 3, *Track* represent the origin and destination data, $q_i(x)$ for blocks 11 and 39 extracted on day 1 is 1 (since the block is extracted entirely), $Z^{\mathbb{s}}(x)$ is the value of blocks 11 and 39 in simulation $\mathbb{s}$, f^d is a summation function for simplicity, and $q^{d,p,c}$ is 2 blocks per day. The model-based prediction $\mathrm{MP}_p^{\mathbb{s}}$ at a processing stream p for a simulation $\mathbb{s}$ is calculated as follows:

$$\mathrm{MP}_p^{\mathbb{s}} = f^p\left(\sum_{c\in C} Track\left(\mathrm{MP}_c^{\mathbb{s}}\right)\right),\ \forall \mathbb{s}\in\mathbb{S}^{\mathrm{I}},\ p\in P \tag{5}$$

Equation (5) first uses the *Track* operator to find the quantity of materials fed to a processing stream with different conveyors, then uses the function f^p to find how materials are transformed into products, and finally uses the model-based prediction from Eq. (4) along with the tracking and transformation information to compute $\mathrm{MP}_p^{\mathbb{s}}$.

2.3.1.3. Action. The state s_t is fed to the actor agent, f_{θ^μ}, to generate an action $a_t \in \mathbb{R}^+$ as shown below:

$$a_t = f_{\theta^\mu}(s_t) + \mathcal{N}_t \tag{6}$$

where, $\mathcal{N}_t$ is the noise added to the action to ensure exploration during the training phase of the proposed AI algorithm. The action a_t is to predict the updated property, $Z^{\mathbb{s}'}(x)$, of a block located at x based on a state s_t in a simulation $\mathbb{s} \in \mathbb{S}^{\mathrm{I}}$.

2.3.1.4. Next state. The next state s_{t+1} is generated by first replacing $Z^{\mathbb{s}}(x)$ in $Z^{\mathbb{s}}(Neigh_x)$ with the action a_t taken by the actor agent to form $Z^{\mathbb{s}'}(Neigh_x)$, and then generating the model-based prediction with $Z^{\mathbb{s}'}(x)$. The rest of the information in the next state s_{t+1} remains the same as in the state s_t.

2.3.1.5. Reward. The action a_t taken by the agent in the state s_t is evaluated by the environment to generate a reward r_t. The reward r_t leverages high-order spatial statistics and consists of three parts, as shown below:

$$r_t = r_{\mathbb{s},t} + r_{\mathscr{J},t} + \sum_{i\in\mathscr{T}} r_{i,t} \tag{7}$$

The first part, $r_{\mathbb{s},t}$, evaluates the likelihood of a_t in the conditional probability distribution function (CPDF), which is generated by searching for replicates of a conditioning data event $D_x^{\mathbb{s}'}$ within a simulation. The second part, $r_{\mathscr{J},t}$, evaluates the likelihood of an action a_t in the CPDF, which is generated by searching for replicates of $D_x^{\mathscr{J}}$ within the spatial sensor data. The third part, $r_{i,t}, \forall i \in \mathscr{T}$, computes the error between a model-based prediction and temporal sensor data for a_t. The CPDF of a given conditioning data event is generated using high-order spatial Legendre moments (Dimitrakopoulos et al., 2010; Mustapha and Dimitrakopoulos, 2011; Yao et al., 2018). High-order spatial Legendre moments capture multi-point spatial statistics and approximates the CPDF of the center node $Z(x)$ for a data event $D_x^{\mathbb{s}'}$ using Legendre polynomials. The initial simulated property $Z^{\mathbb{s}}(Neigh_x)$ inside $Neigh_x$ is searched for all available replicates$N_\zeta^{\mathbb{s}}$, of $D_x^{\mathbb{s}'}$, defined by a distance vector, $H_x^{\mathbb{s}'}$. Let $\zeta_{\mathbb{s},i,j}$ denote the values of each node $j \in N^{\mathbb{s}}$ in the replicate $i \in N_\zeta^{\mathbb{s}}$. The replicates $\zeta_{\mathbb{s},i,j}$ are used to compute the CPDF of the center node $Z(x)$ with Legendre polynomials as:

$$f_{\mathbb{s}}\left(Z(x)\middle|D_x^{\mathbb{s}'}\right) \approx \widetilde{f_{\mathbb{s}}^{W}}\left(Z(x)\middle|D_x^{\mathbb{s}'}\right) = \frac{\sum_{i\in N_\zeta^{\mathbb{s}}} X_i(Z(x))\mathbb{s}\prod_{j\in N^{\mathbb{s}}} X_i\left(\zeta_{\mathbb{s},i,j}\right)}{\sum_{i\in N_\zeta^{\mathbb{s}}}\prod_{j\in N^{\mathbb{s}}} X_i\left(\zeta_{\mathbb{s},i,j}\right)},\ \forall \mathbb{s}\in\mathbb{S}^{\mathrm{I}} \tag{8}$$

where,

$$X_i(Z(j)) = \sum_{w=0}^{W}\left(w+\frac{1}{2}\right)P_w\left(\zeta_{\mathbb{s},i,j}\right)P_w(Z(j)) \tag{9}$$

W is the degree of the Legendre polynomials. $P_w(Z(x))$ is the wth-degree Legendre polynomial for center node $Z(x)$ calculated as follows:

$$P_w(Z(x)) = \frac{1}{2^w w!}\frac{d^w}{dZ(x)^w}\left[\left(Z(x)^2-1\right)^w\right] \tag{10}$$

For more details on the computation of CPDF with high-order spatial Legendre moments, readers are referred to Mustapha and Dimitrakopoulos (2011) and Yao et al. (2018). The first part of the reward calculation is therefore calculated as:

$$r_{\mathbb{s},t} = \lambda_{\mathbb{s}} \cdot \left(\widetilde{f_{\mathbb{s}}^{W}}\left(Z(x)=Z^{\mathbb{s}}(x)\middle|D_x^{\mathbb{s}'}\right) - \widetilde{f_{\mathbb{s}}^{W}}\left(Z(x)=Z^{\mathbb{s}'}(x)\middle|D_x^{\mathbb{s}'}\right)\right), \forall \mathbb{s} \in \mathbb{S}^{\mathrm{I}} \tag{11}$$

where,

$$\lambda_{\mathbb{s}} = \frac{1-E_x^{\mathbb{s}}}{1-E_x^{\mathbb{s}}+1-E_x^{\mathrm{NI}_{\mathscr{J}}}+\sum_{i\in\mathscr{J}}1-E^{\mathrm{NI}_i}}, \forall \mathbb{s} \in \mathbb{S}^{\mathrm{I}} \tag{12}$$

$\lambda_{\mathbb{s}}$ is the weight associated with reproducing the spatial statistics of the initial simulation $\mathbb{s}$. The second part of the reward calculation, $r_{\mathscr{J},t}$, which defines the likelihood of action a_t compared to the initial simulated property $Z^{\mathbb{s}}(x)$ in the CPDF $f_{\mathscr{J}}(Z(x)|D_x^{\mathscr{J}})$ is calculated as:

$$r_{\mathscr{J},t} = \lambda_{\mathscr{J}} \cdot \left(\widetilde{f_{\mathscr{J}}^{W}}\left(Z(x)=Z^{\mathbb{s}}(x)\middle|D_x^{\mathscr{J}}\right) - \widetilde{f_{\mathscr{J}}^{W}}\left(Z(x)=Z^{\mathbb{s}'}(x)\middle|D_x^{\mathscr{J}}\right)\right), \forall\, \mathbb{s} \in \mathbb{S}^{\mathrm{I}} \tag{13}$$

where,

$$f_{\mathscr{J}}\left(Z(x)\middle|D_x^{\mathscr{J}}\right) \approx \widetilde{f_{\mathscr{J}}^{W}}\left(Z(x)|D_x^{\mathscr{J}}\right) = \frac{\sum_{i\in N_\zeta^{\mathscr{J}}} X_i(Z(x)) \cdot \prod_{j\in N^{\mathscr{J}}} X_i\left(\zeta_{\mathscr{J},i,j}\right)}{\sum_{i\in N_\zeta^{\mathscr{J}}} \prod_{j\in N^{\mathscr{J}}} X_i\left(\zeta_{\mathscr{J},i,j}\right)} \tag{14}$$

and

$$\lambda_{\mathscr{J}} = \frac{1-E_x^{\mathrm{NI}_{\mathscr{J}}}}{1-E_x^{\mathbb{s}}+1-E_x^{\mathscr{J}}+\sum_{i\in\mathscr{J}}1-E^{\mathrm{NI}_i}}, \forall\, \mathbb{s} \in \mathbb{S}^{\mathrm{I}} \tag{15}$$

$\lambda_{\mathscr{J}}$ is the weight associated with reproducing the statistics of the spatial sensor data. $\widetilde{f_{\mathbb{s}}^{W}}(Z(x)|D_x^{\mathscr{J}})$ is the CPDF and is computed using Eq. (14). For this, the spatial sensor data $\mathrm{NI}_{\mathscr{J}}(Neigh_x)$ is searched for all replicates, $N_\zeta^{\mathscr{J}}$ of $D_x^{\mathscr{J}}$, defined by a distance vector, $H_x^{\mathscr{J}}$. Let $\zeta_{\mathscr{J},ij}$, denote the values of each node $j \in N^{\mathscr{J}}$ in the replicate $i \in N_\zeta^{\mathscr{J}}$. The value of the replicates $\zeta_{\mathscr{J},i,j}$ are used in Eqs. (9) and (10) to compute the CPDF with Eq. (14). The third part of the reward calculation, $r_{i,t}, \forall i \in \mathscr{T}$, which defines the difference between the model-based prediction and the temporal sensor data for action a_t and the initial simulated property $Z^{\mathbb{s}}(x)$ is calculated as:

$$r_{i,t} = \lambda_i \cdot \left(\left|\mathrm{MP}_i^{\mathbb{s}} - \mathrm{NI}_i\right| - \left|\mathrm{MP}_i^{\mathbb{s}'} - \mathrm{NI}_i\right|\right) \cdot \gamma_i^{\mathrm{MP}},\ \forall i \in \mathscr{T}, \mathbb{s} \in \mathbb{S}^{\mathrm{I}}, \mathbb{s}' \in \mathbb{S}^{\mathrm{U}} \tag{16}$$

where,

$$\lambda_i = \frac{1-E^{\mathrm{NI}_i}}{1-E_x^{\mathbb{s}}+1-E_x^{\mathrm{NI}_{\mathscr{J}}}+\sum_{i\in\mathscr{J}}1-E^{\mathrm{NI}_i}},\ \forall i \in \mathscr{T}, \mathbb{s} \in \mathbb{S}^{\mathrm{I}} \tag{17}$$

$\mathrm{MP}_i^{\mathbb{s}}$ and $\mathrm{MP}_i^{\mathbb{s}'}$ are the model-based predictions at component $i \in \mathscr{T}$ (calculated using Eqs. (3)–(5)), for the initial simulated property $Z^{\mathbb{s}}(x)$ and the action a_t respectively. $\lambda_i, \forall i \in \mathscr{T}$ is the weight associated with minimizing the difference between the model-based prediction and the temporal sensor data at component i. $r_{i,t}$ is a subtraction of two differences, as seen in Eq. (16), as opposed to $r_{\mathbb{s},t}$ and $r_{\mathscr{J},t}$, which are subtractions of two probabilities. Therefore, an adjustment factor γ_i^{MP} is used in Eq. (16) to ensure that the third part of the reward calculation is of the same magnitude as the other parts.

2.3.2. Actor and critic architecture

The actor agent, f_{θ^μ}, is a CNN which takes as input the state s_t which includes the initial simulation, spatial sensor data, and some additional data. The additional data includes the temporal sensor data, average conditional variance associated with a property of a block calculated using Eq. (1), error in the sensor data, and model-based predictions calculated using Eqs. (3)–(5). The additional data are not added as inputs until the fully connected layer in the actor agent, as shown in Fig. 4 (a). The actor agent takes an action a_t based on the state s_t, which is to predict the updated simulated property $Z^{\mathbb{s}'}(x)$ of a block located at x. The critic agent, f_{θ^Q}, is also a CNN, which takes as input the state s_t and as an additional input the action taken by the actor agent. Similar to the actor agent, the additional data and the action are not added as inputs until the fully connected layer in the critic agent as shown in Fig. 4(b). The critic agent is an action-value function which evaluates the action a_t taken by the actor agent in the state s_t.

2.3.3. Actor-critic training

The actor and critic agents are trained using DDPG reinforcement learning, as shown in Fig. 5. The agents are initialized randomly at time, $t = 1$, with weights, θ^μ and θ^Q, respectively. In addition to the actor and critic agents, two target agents (target actor and target critic), denoted by f_{θ^μ}', and f_{θ^Q}', and parametrized by $\theta^{\mu'}$ and $\theta^{Q'}$, are created to avoid divergence issues. The parameters of the target network are initialized as $\theta^{\mu'} \leftarrow \theta^\mu$ and $\theta^{Q'} \leftarrow \theta^Q$. The replay memory buffer is initialized at time, $t = 1$.

A random path is defined to visit all the blocks in the mineral deposit, and the point at which a block is visited along this path is referred to as a time step t. At t an action is taken by the actor agent based on the state s_t generated from the environment (see Sect. 2.3.1.3) for the mining block in consideration. The action is evaluated in the environment to generate the reward r_t and the next state s_{t+1}. The state, action, reward, and next state tuple (s_t a_t, r_t, and s_{t+1} respectively) is stored in replay memory R. At every N_I iteration the memory is sampled to generate mini batches of transitions (s_t a_t, r_t, and s_{t+1}) of size N_{BS}. The sampled mini batches are used to train the actor and critic agents. More specifically, the parameters, θ^Q, of the critic agent are updated to minimize the temporal difference error loss L given by:

$$L = \frac{1}{N_{BS}}\sum_{i\in N_{BS}}\left(\left(r_i + \gamma f_{\theta^Q}'\left(s_{i+1}, f_{\theta^\mu}'(s_{i+1})\right)\right) - \left(f_{\theta^Q}(s_i,a_i)\right) + c\cdot\left\|\theta^Q\right\|^2\right) \tag{18}$$

$c\cdot\left\|\theta^Q\right\|^2$ is an L2 regularization added to the loss function with a penalty cost of c, to avoid overfitting. The actor agent is trained by the sampled policy gradient given as:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N_{BS}}\sum_{i\in N_{BS}}\left(\nabla_{f_{\theta^\mu}(s_i)} f_{\theta^Q}(s_i, f_{\theta^\mu}(s_i))\nabla_{f_{\theta^\mu}(s_i)} f_{\theta^\mu}(s_i)\right) \tag{19}$$

The sampled policy gradient first takes the gradient of the critic agent parameters, θ^Q, with respect to the action a_t taken by the actor, and then takes the gradient of the actor agent parameters, θ^μ, with respect to the action a_t. The parameters of the trained actor and critic agents are then used to perform soft updates to the target agents as follows:

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \tag{20}$$

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \tag{21}$$

τ defines the strategy to blend the target agent parameters with the trained agent parameters. The new parameters of the actor and critic agents are used for further learning. The replay memory is used to ensure the mini-batch samples used for training are independently and identically distributed. In addition, the proposed algorithm learns in mini batches, i.e., offline to make efficient use of hardware optimizations (Lillicrap et al., 2015; Sutton and Barto, 2017).

2.4. Responding to incoming new information

The proposed AI algorithm in Sects. 2.3.1–2.3.3 trains the CNN actor agent which can update the simulations of pertinent spatial properties of

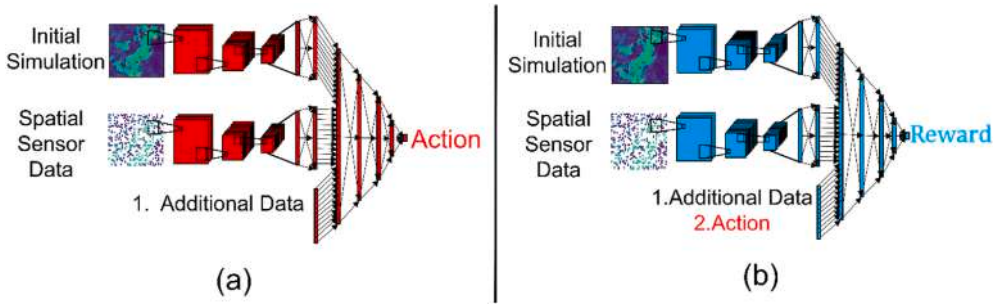

Fig. 4. (a) Actor and (b) Critic agent configuration in the proposed AI algorithm.

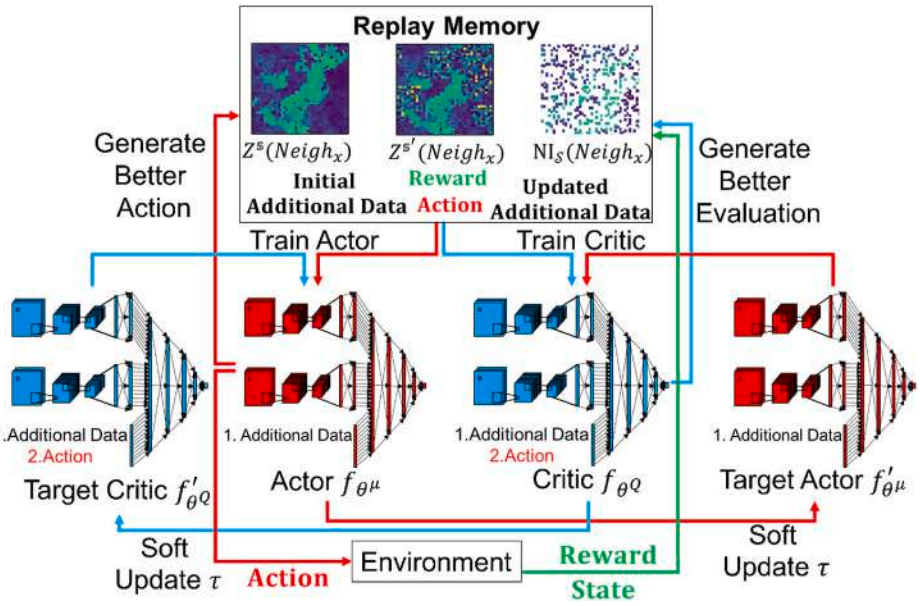

Fig. 5. Actor and critic agents training in the proposed AI algorithm.

mineral deposits with incoming new information in an operating mining environment. The spatial sensor data, $\mathrm{NI}_i(\boldsymbol{x}),\ i \in \mathscr{S}$, and temporal sensor data, $q^i,\ \mathrm{NI}_i(q^i),\ \forall i \in \mathscr{T}$, along with the tracking sensor data, *Track*, and initial simulations, $\mathbb{s} \in \mathbb{S}^{\mathrm{I}}$, are fed to the AI algorithm as shown in Fig. 6. The AI algorithm then initializes the environment presented in Sect. 2.3.1 with all the information. A random path is then defined by the environment to visit all the blocks within the mineral deposit. At each block, a state generated by the environment is fed to the trained actor agent that predict the updated simulated property of the block. The environment uses the action to generate the next state, and the process continues until all the blocks are visited. The updated simulated property of all the blocks forms the set of updated simulations $\mathbb{S}^{\mathrm{U}}$. The updated simulations are then used to generate the updated model-based predictions. In parallel, the agent's parameters are updated by training over the updated simulations and newly collected information.

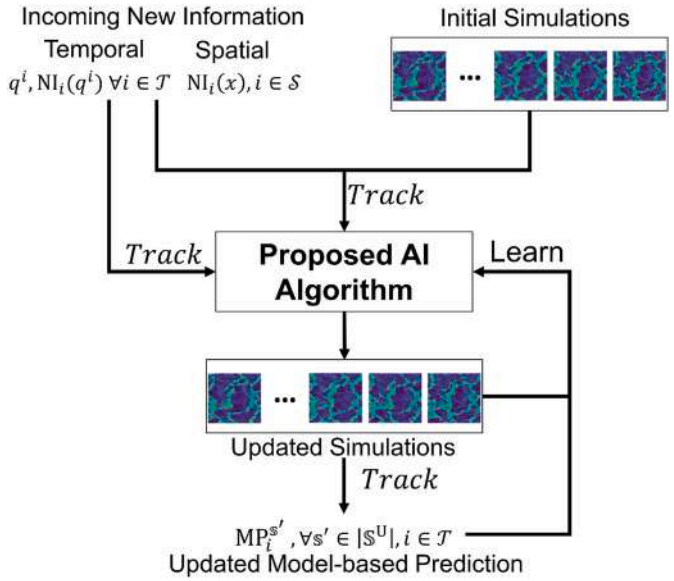

Fig. 6. Real-time learning and updating with incoming new spatial and temporal information.

3. Application at a synthetic copper mining operation

The proposed AI algorithm is programmed using Python and Tensorflow. It is applied in this section to a fully known public dataset (Mao and Journel, 1999b), which is adopted to represent a copper deposit in the present case study. Twenty initial simulations of copper grades are generated for two different areas, Area 1 and Area 2, of the copper deposit using high-order simulation (Minniakhmetov et al., 2018; Yao et al., 2018). The two areas contain 416 drillhole data points each, sampled from the corresponding areas of the fully known dataset, with an average spacing of 5m using random stratified sampling. Each area of the deposit consists of 13,000 blocks of size 1 x 1 x 1 m^3. The incoming

spatial sensor data of copper grades is generated by randomly sampling 2,600 data points with an average spacing of 1.3 m from the same sections of the fully known dataset. The error in each spatial sensor data point is sampled from a normal distribution with a mean of 0 and a standard deviation of 0.45. The temporal sensor data of copper grades from the processing mill is generated by randomly sampling the same section of the fully known dataset in such a way that it imitates the process of collection of such data in a mining operation. The generated temporal sensor data contains 2,600 data points. The error in each temporal sensor data point is sampled from a normal distribution with a mean of 0 and a standard deviation of 0.6. The standard deviation of the normal distribution for error in the temporal sensor data is lower than in the spatial sensor data to reflect the quality of the respective sensors. In actual mining environments these sensors will show proportional effect in their related measurement errors, however, for simplicity in demonstrating the applied aspects of the proposed method proportional effect is not considered in this synthetic case study.

The copper mining operation considered is shown in Fig. 7, and consists of a mine, a waste dump, a processing mill and a customer. Multiple drilling machines located at the mine perform the drilling operations and capture the spatial new information about the grade of the drilled mining blocks. The materials from the mine are extracted by two shovels and are loaded into trucks that haul the materials to either a processing mill or a waste dump. The processing mill blends and processes the received materials to generate copper products, which are transported to the customers. The sensors at the processing mill capture the temporal new information by measuring the grade of the generated copper products.

Two different datasets – training and testing – are generated using the process outlined above to represent the operations and the collection of new information in two areas, namely Area 1 and Area 2, of the deposit. The data from Area 1 of the deposit is used to train the proposed AI algorithm. In an operating mine, this dataset would be the historical data. The training results shows the learning capabilities of the proposed algorithm. The data from Area 2 of the deposit is used for testing the proposed algorithm. This dataset was never used for training and represents how the proposed algorithm would be used in an operating mine to update the simulations of pertinent properties of the deposit with incoming new information. Section 3.2 discusses the training and testing results of the proposed algorithm and shows that the algorithm can be generalized and has practical applications in an operating mine. Throughout the presentation and discussion of the results, the fully known data and its histogram, variogram and spatial cumulant (Dimitrakopoulos et al., 2010; Mustapha and Dimitrakopoulos, 2011) maps are also shown for reference. The fully known data is referred to hereafter as the ground truth model. The algorithm takes less than 30 s to update the simulations of copper grade of the different areas of the deposit.

3.1. Parameters

The proposed AI algorithm is trained on an Intel® i7-8700 machine with an 8-core processor and an NVIDIA GeForce GTX 1050 GPU for approximately 2 days. The parameters used for the case study at the copper mining operation is detailed in Table 3.

The actor and critic agent architecture are shown in Fig. 8. The total number of parameters for the actor agent is ≈ 497000, and the critic agent is ≈ 498000.

3.2. Results

The results described in this section for Area 1 of the deposit are related to the training of the agents in the proposed algorithm and shows its learning capabilities. The results for Area 2 of the deposit are related to testing of the agent and shows the generalization and applicability of the proposed algorithm.

Fig. 9 and Fig. 10 show the drillhole data and two of the initial simulations of copper grade for Areas 1 and 2 of the deposit, respectively. The initial simulations for Area 2 in Fig. 10 show a presence of very different geological patterns compared to Area 1: the curvilinear structures are horizontal instead of vertical in Fig. 9. Fig. 11(a–c) and Fig. 12(a–c) show the spatial sensor data, the error in the spatial sensor data and the processing mill sensor data, respectively, that are collected during operations in Areas 1 and 2 of the deposit, respectively.

Fig. 13(a–b) and Fig. 14(a–b) show one of the initial simulations and its corresponding updated simulation of copper grades, respectively, for

Table 3
Parameters used for the case study at the copper mining operation.

Parameters	Value
$Neigh_x$	21 x 21
$\mathbb{S}^I$	20
N^s	8, $\forall s \in \mathbb{S}^I$
γ_s	10, $\forall s \in \mathbb{S}^I$
$\mathcal{N}_t$	Ornstein-Uhlenbeck method, with a standard deviation of 1e-1 (Uhlenbeck and Ornstein, 1930).
W	10
τ	1e-3
N_{BS}	500
γ	0.99
N_R	1e6
N_I	65000
γ_i^{MP}	10, $\forall i \in \mathcal{I}$
c	1e-3
N_U	0, No training was performed over testing data
N_{TE}	100
N_T	200
N_{UE}	0, No training was performed over testing data

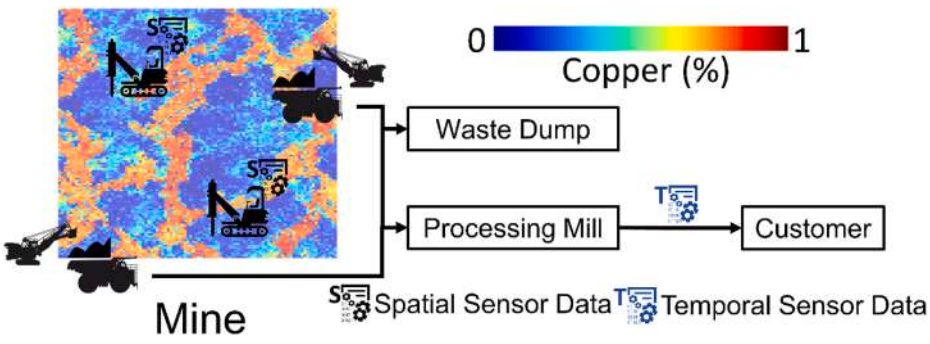

Fig. 7. The copper mining operation.

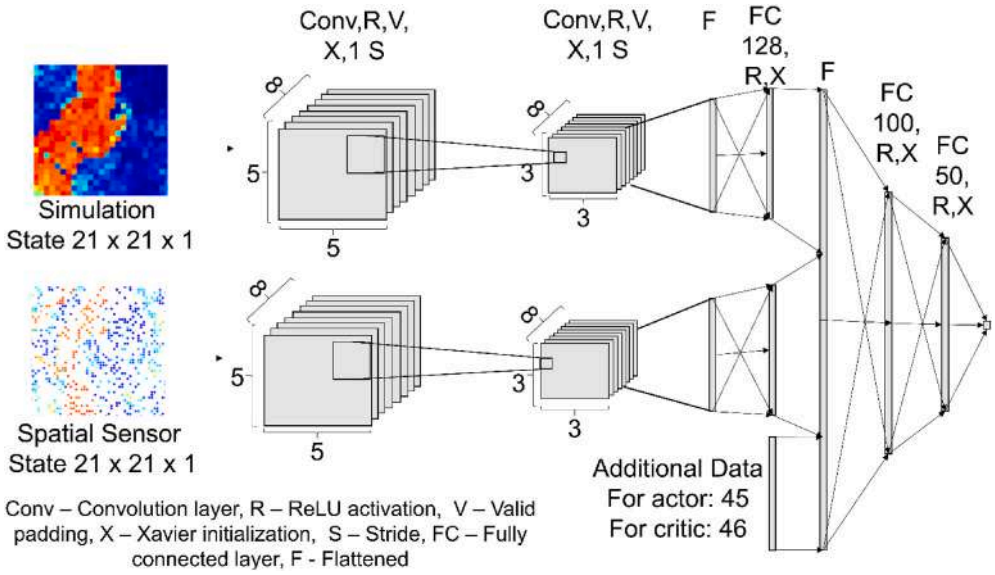

Fig. 8. Actor and critic agent architecture used for the case study at the copper mining operation.

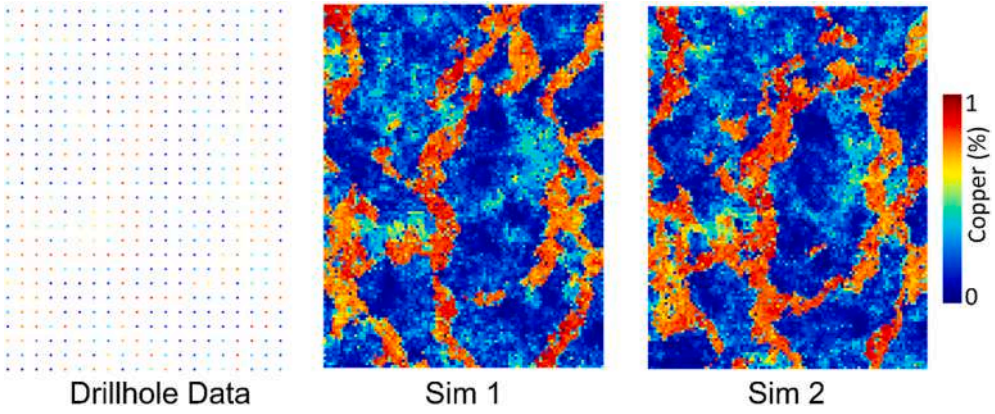

Fig. 9. Drillhole data and the 2 of the initial copper grade simulations for Area 1 of the deposit.

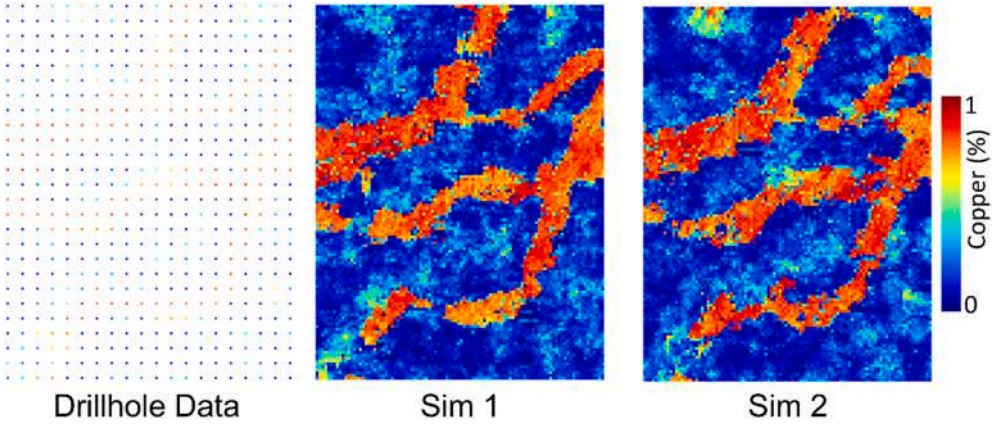

Fig. 10. Drillhole data and two of the initial simulations of copper grade for Area 2 of the deposit.

Areas 1 and 2 of the deposit. The updated simulation as shown in Fig. 13 (b) during training of the agents reproduces the curvilinear vertical structures, as inferred from initial simulations and spatial sensor data for this area. The proposed AI algorithm can also reproduce the horizontal curvilinear structures in the updated simulation, as inferred from the initial simulation, and the spatial sensor data for this area, even though the training data set had vertical curvilinear structures. In addition, the updated simulation closely resembles the ground truth model shown in Figs. 13(c) and Figure 14(c) for Area 1 and 2, respectively.

The initial and updated simulations are validated through the

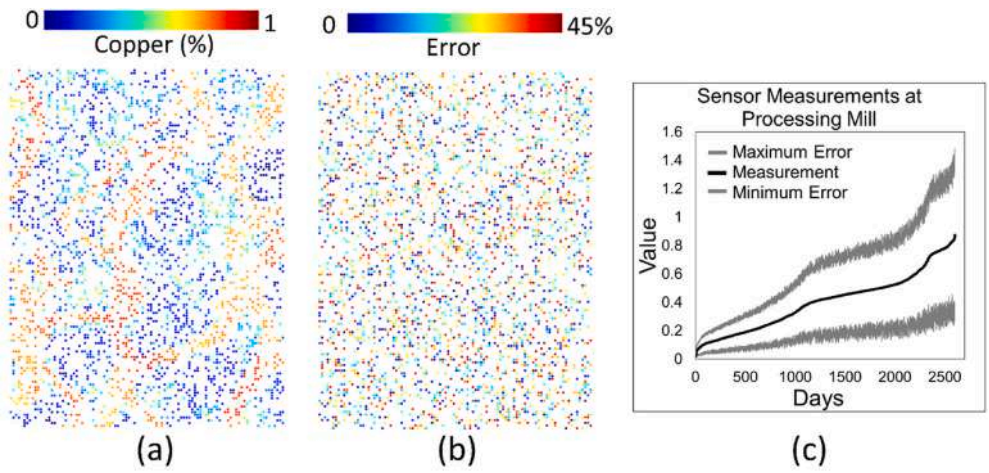

Fig. 11. (a) Spatial sensor data; (b) error in the spatial sensor data; and (c) processing mill sensor data of copper grades from Area 1 of the deposit.

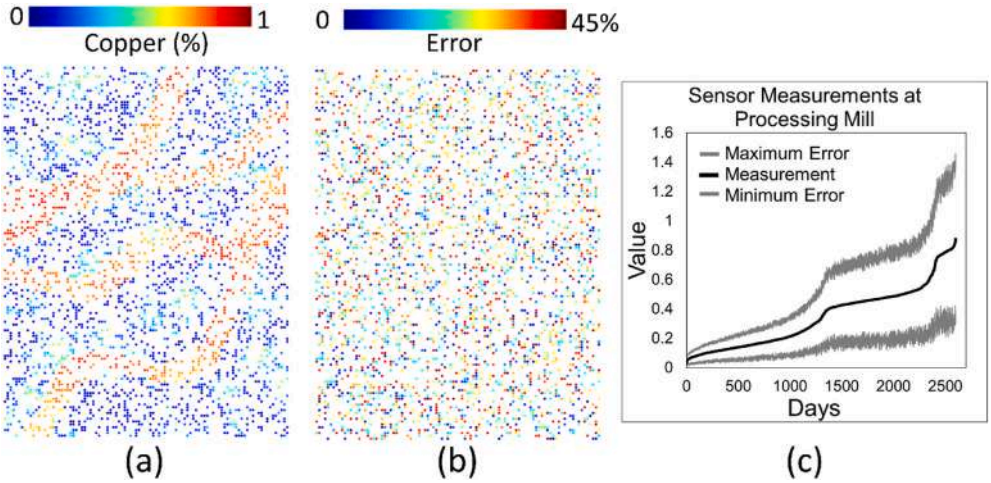

Fig. 12. (a) Spatial sensor data; (b) error in the spatial sensor data; and (c) processing mill sensor data of copper grades from Area 2 of the deposit.

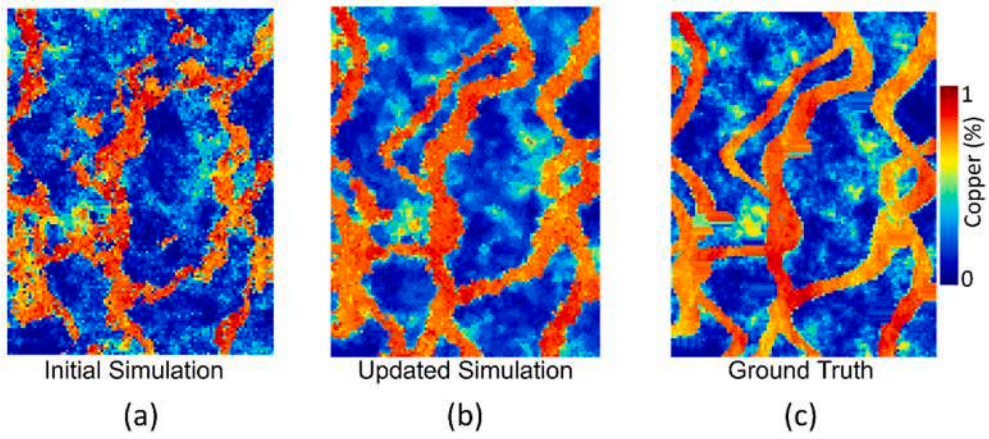

Fig. 13. (a) One of the initial simulations compared to (b) its corresponding updated simulation, and (c) the ground truth model of copper grades for Area 1 of the deposit.

analysis of the histogram and variogram reproduction in Fig. 15. The initial and updated simulations both respect the histogram and variogram of the drillhole data. The updated simulations present a closer resemblance to the histogram and variogram of the spatial sensor data, as shown in Fig. 15(c) and (d), respectively. Fig. 16(a–b) show the 3rd and 4th order spatial cumulant maps (Dimitrakopoulos et al., 2010; Mustapha and Dimitrakopoulos et al., 2011), respectively, for the initial copper grade simulation for Area 2 of the deposit. The spatial cumulant maps of the updated simulation (Fig. 16(c–d)) show more connected structures, as seen in the spatial cumulant maps of the ground truth

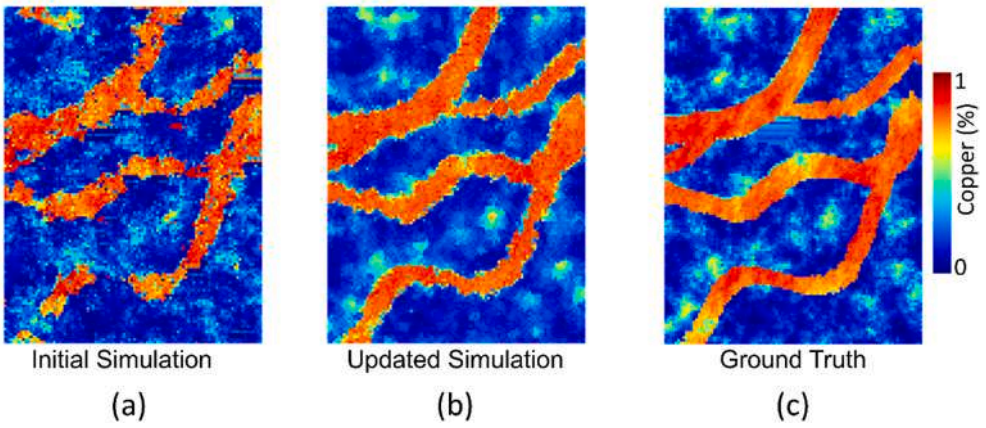

Fig. 14. (a) One of the initial simulations compared to (b) its corresponding updated simulation, and (c) the ground truth model of copper grades for Area 2 of the deposit.

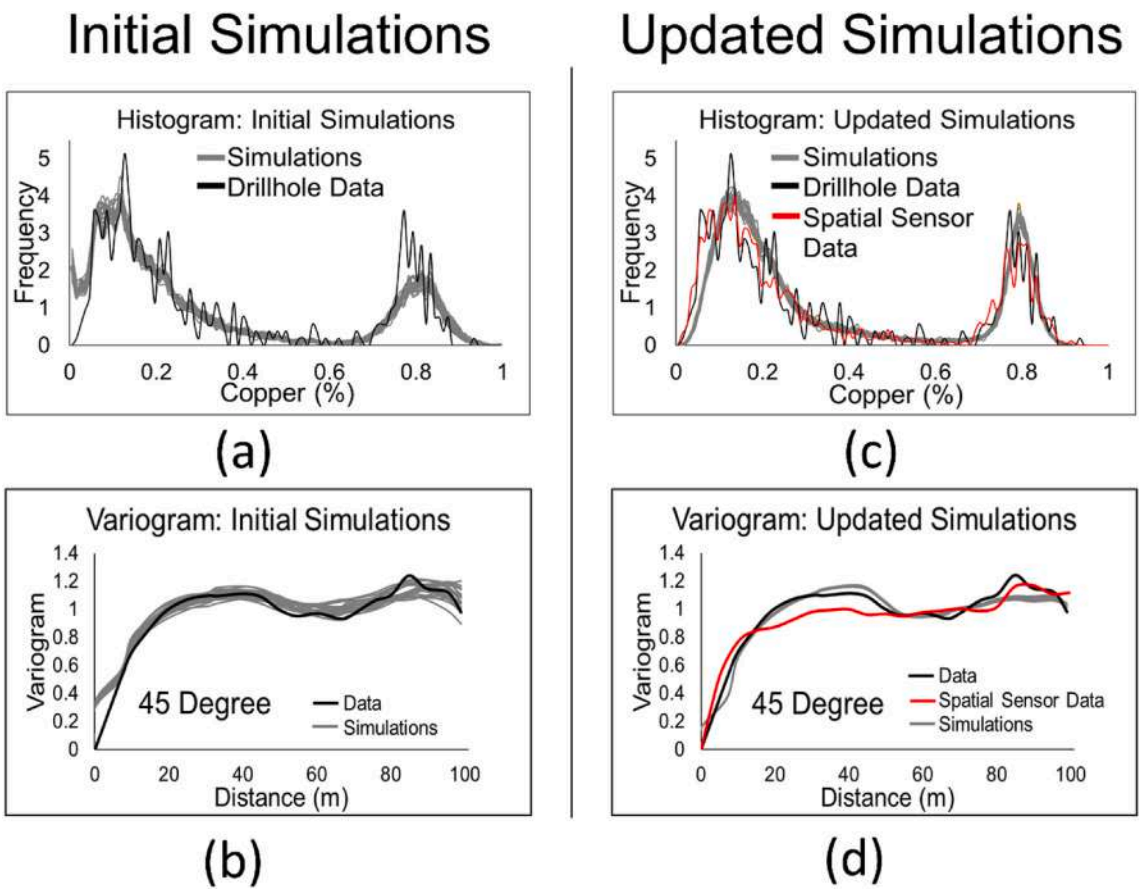

Fig. 15. Histogram and variogram of (a–b) the initial simulations compared to (c–d) the updated simulations of copper grades for Area 2 of the deposit.

model (Fig. 16(e–f)). The proposed AI algorithm was not trained with data from this area, yet it still reproduces the histogram, variogram and spatial cumulant maps, while also updating the initial simulations of copper grades for Area 2 of the deposit. The histogram, variogram, and cumulant maps validation results for Area 1 is provided in the supplementary materials.

Fig. 17(a–b) and Fig. 18(a–b) show the model-based prediction for the initial and its corresponding updated simulation of copper grades, respectively, for Areas 1 and 2 of the deposit. The model-based prediction for the updated simulation shows fewer deviations from the processing mill sensor measurements, when compared to the initial simulation both for Area 1 and Area 2. A spread reduction (*SR*) criterion defined by Eq. 22

$$SR = \frac{1}{|\mathbb{S}^{\mathrm{I}}|}\sum_{s\in\mathbb{S}^{\mathrm{I}}}\frac{1}{|\mathscr{T}|}\sum_{i\in\mathscr{T}}\left|\mathrm{MP}_i^{s}-\mathrm{NI}_i\right|-\left|\mathrm{MP}_i^{s'}-\mathrm{NI}_i\right| \quad ,\ \forall\, s'\in\mathbb{S}^{\mathrm{U}} \tag{22}$$

is used to quantify the reduction in measurement vs predicted values error, for updated and initial simulations. The spread reduction value for Areas 1 and 2 of the deposit are 0.07 and 0.05, respectively, given the method proposed in this work.

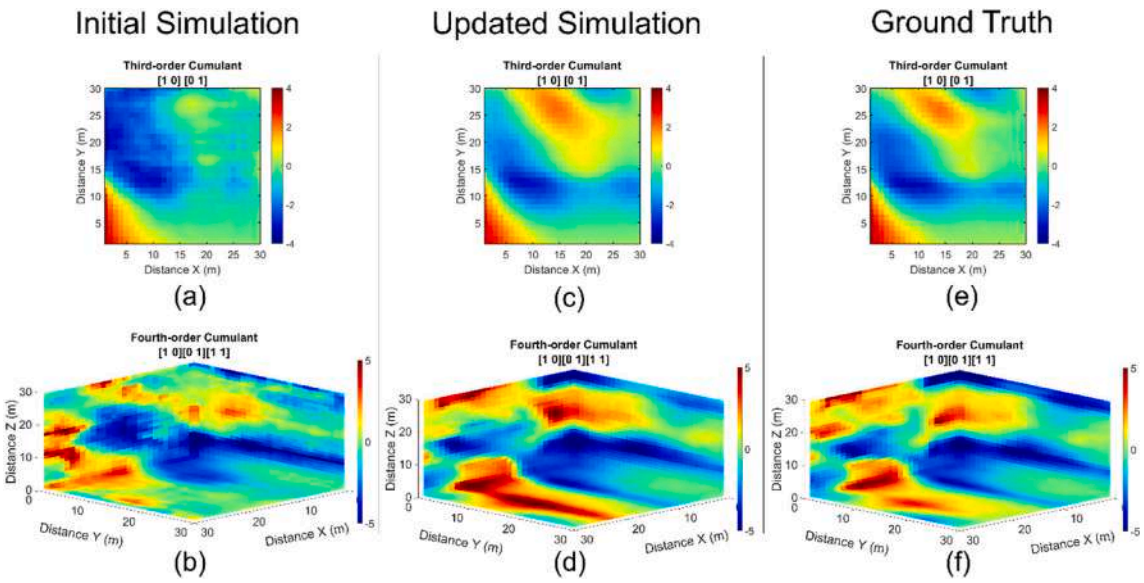

Fig. 16. Third- and fourth-order spatial cumulant maps for (a–b) the initial simulations compared to (c–d) its corresponding updated simulations and (e–f) the ground truth model of copper grades for Area 2 of the deposit.

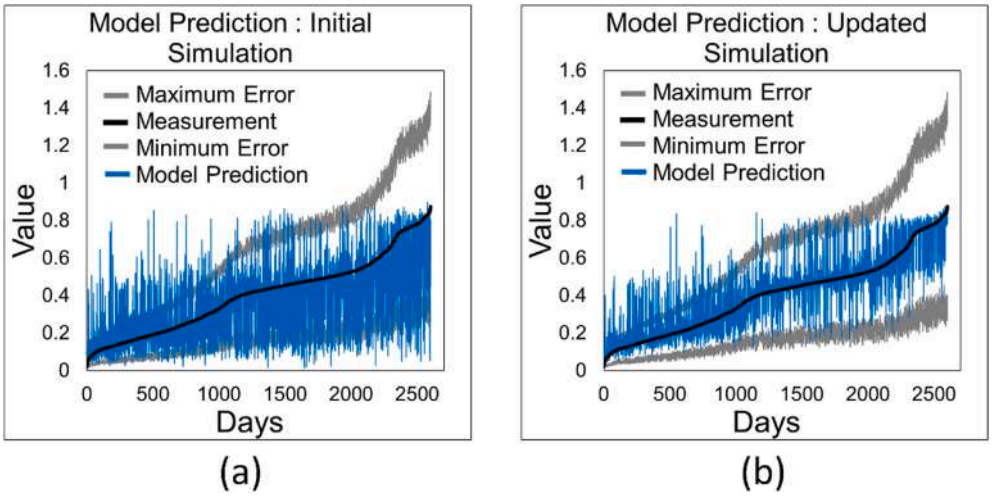

Fig. 17. Model-based prediction generated with (a) an initial simulation compared to (b) its corresponding updated simulation of copper grades for Area 1 of the deposit.

4. Conclusions

This paper proposes a new self-learning artificial intelligence algorithm that uses deep policy gradient reinforcement learning and leverages high-order spatial statistics to train actor and critic agents to update the simulations of pertinent spatial properties of a mineral deposit with incoming new information. The algorithm is general and can be applied to any mining operation with multiple sources of incoming new information. The algorithm visits the mining blocks within the deposit following a random path. For each block, a state representation is generated and fed to the actor and critic agent, in this case a convolutional neural network. The actor agent takes an action, which is to predict the updated spatial properties of the blocks based on the state representation. The action is evaluated by the critic agent. In parallel, the environment also evaluates the action using high-order spatial statistics and generates both a reward and the next state representation. The state, action, reward and next state data is stored in a replay memory, which is sampled at regular intervals to train the agents. The improved agents are then used for further training. An application of the proposed algorithm at a synthetic copper mining operation demonstrates its efficiency and applied aspects. The case study shows that the algorithm can account for softness in the incoming new information (both spatial and temporal) to update the copper grade simulations of the deposit while reproducing geological patterns and high-order spatial statistics. The case study also highlights the learning and generalization capabilities of the algorithm through its application in different parts of the deposit, which have different geological patterns and curvilinear structures. The algorithm proposed in this work is not limited to the number of mining blocks used during training, however, the updating time during testing will increase as the number of blocks that need to be updated increases. The proposed algorithm assumes that the training and testing areas present similar geological characteristics. The extension of the proposed algorithm to 3-dimensional deposits is straightforward by using convolution 3D layers in the agent's architecture. Future research will focus on expanding and applying the algorithm for multiple elements and using preferentially sampled new incoming information from an operating mine.

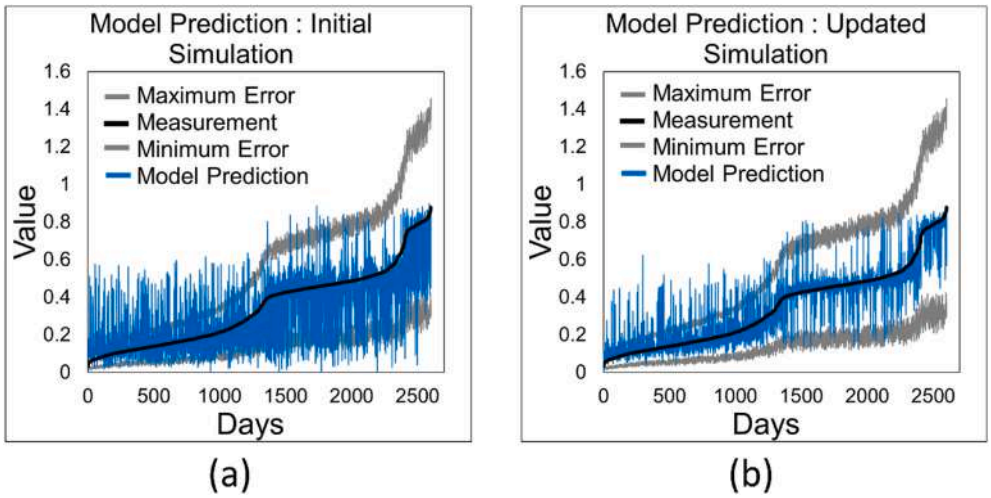

Fig. 18. Model-based prediction generated with (a) an initial simulation compared to (b) its corresponding updated simulation of copper grades for Area 2 of the deposit.

CRediT authorship contribution statement

Ashish Kumar: Conceptualization, Methodology, Software, Data Preperation, Visualization, Validation, Writing – original draft, Writing – review & editing. **Roussos Dimitrakopoulos:** Supervision, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The work in this paper was funded by the National Sciences and Engineering Research Council (NSERC) of Canada CRD Grant 500414-16 and NSERC Discovery Grant 239019, the industry consortium members of McGill University's COSMO Stochastic Mine Planning Laboratory (AngloGold Ashanti, Barrick Gold, BHP, De Beers, IAMGOLD, Kinross Gold, Newmont Corporation, and Vale); and the Canada Research Chairs Program.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cageo.2021.104962.

Computer code availability

- Name of code: MineralDepositAI
- Developer: Ashish Kumar (McGill University, FDA Building, Quebec, Canada; ashish.kumar@mail.mcgill.ca)
- Year first available: 2020
- Hardware required: Windows with NVIDIA GPU of at least 4 GB memory and compute capability of 6 and higher
- Software required: Python Integrate Development Environment (Visual Studio, 2015 preferred)
- Program language: Python 3.6.1
- Program size: 85.8 MB (including source code, documentation, results, and data)
- Access to code: download from https://github.com/ashishrokz1993/MineralDepositAICG

References

Aanonsen, S., Nævdal, G., Oliver, D., Reynolds, A., Vallès, B., 2009. The ensemble Kalman Filter in reservoir engineering: a Review. SPE J. 14, 393–412.
Ángel, P., Benndorf, J., Mueller, U., 2021. Resource and grade control model updating for underground mining production settings. Math. Geosci. 53, 757–779. https://doi.org/10.1007/s11004-020-09881-2.
Benndorf, J., 2020. Data assimilation for resource model updating. In: SpringerBriefs in Applied Sciences and Technology. Springer, pp. 19–60. https://doi.org/10.1007/978-3-030-40900-5_3.
Benndorf, J., 2015. Making use of online production data: sequential updating of mineral resource models. Math. Geosci. 47, 547–563. https://doi.org/10.1007/s11004-014-9561-y.
Chaowasakoo, P., Leelasukseree, C., Wongsurawat, W., 2014. Introducing GPS in fleet management of a mine: impact on hauling cycle time and hauling capacity. Int. J. Technol. Intell. Plann. 10, 49–66. https://doi.org/10.1504/IJTIP.2014.066711.
Chen, Y., Oliver, D.S., 2012. Ensemble randomized maximum likelihood method as an iterative ensemble smoother. Math. Geosci. 44, 1–26. https://doi.org/10.1007/s11004-011-9376-z.
Conjard, M., Grana, D., 2021. Ensemble-based seismic and production data assimilation using selection kalman model. Math. Geosci. 1–24 https://doi.org/10.1007/s11004-021-09940-2.
Dalm, M., Buxton, M.W.N., van Ruitenbeek, F.J.A., 2018. Ore–waste discrimination in epithermal deposits using near-infrared to short-wavelength infrared (NIR-SWIR) hyperspectral imagery. Math. Geosci. 51, 1–27. https://doi.org/10.1007/s11004-018-9758-6.
De Jong, T.P.R., 2004. Automatic sorting of minerals. In: IFAC Proceedings, pp. 441–446. https://doi.org/10.1016/s1474-6670(17)31064-9.
Deutsch, C., Journel, A.G., 1992. In: GSLIB: Geostatistical Software Library and User's Guide, second ed. Oxford University Press, New York.
Dimitrakopoulos, R., Mustapha, H., Gloaguen, E., 2010. High-order statistics of spatial random fields: exploring spatial cumulants for modeling complex non-Gaussian and non-linear phenomena. Math. Geosci. 42, 65–99. https://doi.org/10.1007/s11004-009-9258-9.
Fu, J., Gómez-Hernández, J.J., Du, S., 2017. A gradient-based blocking Markov chain Monte Carlo method for stochastic inverse modeling. In: Quantitative Geology and Geostatistics. Springer, Cham, pp. 777–788. https://doi.org/10.1007/978-3-319-46819-8_53.
Gilman, J., Ozgen, C., 2013. Reservoir Simulation: History Matching and Forecasting. Society of Petroleum Engineers, Texas.
Goetz, A.F.H., Curtiss, B., Shiley, D.A., 2009. Rapid gangue mineral concentration measurement over conveyors by NIR reflectance spectroscopy. Miner. Eng. 22, 490–499. https://doi.org/10.1016/j.mineng.2008.12.013.
Gómez-Hernández, J.J., Srivastava, R.M., 2021. One step at a time: the origins of sequential simulation and beyond. Math. Geosci. 53, 193–209. https://doi.org/10.1007/s11004-021-09926-0.
Gutiérrez-Esparza, J.C., Gómez-Hernández, J.J., 2017. Inverse modeling aided by the classification and regression tree (CART) algorithm. In: Quantitative Geology and Geostatistics. Springer, Cham, pp. 805–819. https://doi.org/10.1007/978-3-319-46819-8_55.
Hu, L.Y., 2000. Gradual deformation and iterative calibration of Gaussian-related stochastic models. Math. Geol. 32, 87–108. https://doi.org/10.1023/A:1007506918588.
Iyakwari, S., Glass, H.J., Rollinson, G.K., Kowalczuk, P.B., 2016. Application of near infrared sensors to preconcentration of hydrothermally-formed copper ore. Miner. Eng. 85, 148–167. https://doi.org/10.1016/j.mineng.2015.10.020.

Jewbali, A., Dimitrakopoulos, R., 2011. Implementation of conditional simulation by successive residuals. Comput. Geosci. 37, 129–142. https://doi.org/10.1016/j.cageo.2010.04.008.
Journel, A.G., 1994. Modeling uncertainty: some conceptual thoughts. In: Geostatistics for the Next Century. Quantitative Geology and Geostatistics. Springer, pp. 30–43. https://doi.org/10.1007/978-94-011-0824-9_5.
Journel, A.G., Alabert, F.G., 1990. New method for reservoir mapping. J. Petrol. Technol. 42, 212–218. https://doi.org/10.2118/18324-pa.
Koellner, W.G., Brown, G.M., Rodríguez, J., Pontt, J., Cortés, P., Miranda, H., 2004. Recent advances in mining haul trucks. IEEE Trans. Ind. Electron. 51, 321–329. https://doi.org/10.1109/TIE.2004.825263.
Li, Y., Sepúlveda, E., Xu, C., Dowd, P., 2021. A rapid updating method to predict grade heterogeneity at smaller scales. Math. Geosci. 1–24 https://doi.org/10.1007/s11004-020-09901-1.
Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. arXiv Prepr. 1–14. http://arxiv.org/abs/1509.02971.
Liu, Y., Sun, W., Durlofsky, L.J., 2019. A deep-learning-based geological parameterization for history matching complex models. Math. Geosci. 51, 725–766. https://doi.org/10.1007/s11004-019-09794-9.
Mao, S., Journel, A.G., 1999a. Conditional 3D simulation of lithofacies with 2D seismic data. Comput. Geosci. 25, 845–862. https://doi.org/10.1016/S0098-3004(99)00006-0.
Mao, S., Journel, A.G., 1999b. Generation of a reference petrophysical/seismic data set: the Stanford V reservoir. Report. In: Stanford Cent. Reserv. Forecast (Stanford, CA).
Minniakhmetov, I., Dimitrakopoulos, R., 2021. High-order data-driven spatial simulation of categorical variables. Math. Geosci. 1–23 https://doi.org/10.1007/s11004-021-09943-z.
Minniakhmetov, I., Dimitrakopoulos, R., Godoy, M., 2018. High-order spatial simulation using Legendre-like orthogonal splines. Math. Geosci. 50, 753–780. https://doi.org/10.1007/s11004-018-9741-2.
Mustapha, H., Dimitrakopoulos, R., 2011. HOSIM: a high-order stochastic simulation algorithm for generating three-dimensional complex geological patterns. Comput. Geosci. 37, 1242–1253. https://doi.org/10.1016/j.cageo.2010.09.007.
Naraghi, M.E., Srinivasan, S., 2015. Integration of seismic and well data to characterize facies variation in a carbonate reservoir-the tau model revisited. In: Petroleum Geostatistics. European Association of Geoscientists and Engineers, EAGE, pp. 243–247. https://doi.org/10.3997/2214-4609.201413633.
Neves, J., Pereira, M.J., Pacheco, N., Soares, A., 2018. Updating mining resources with uncertain data. Math. Geosci. 51, 905–924. https://doi.org/10.1007/s11004-018-9759-5.
Oliver, D., Reynolds, A., Liu, N., 2008. Inverse Theory for Petroleum Reservoir Characterization and History Matching. Cambridge University Press, Cambridge.
Oliver, D.S., 1996. Multiple realizations of the permeability field from well test data. SPE J. 1, 145–154.
Oliver, D.S., Chen, Y., 2011. Recent progress on reservoir history matching: a review. Comput. Geosci. 15, 185–221. https://doi.org/10.1007/s10596-010-9194-2.
Oliver, D.S., Cunha, L.B., Reynolds, A.C., 1997. Markov chain Monte Carlo methods for conditioning a permeability field to pressure data. Math. Geol. 29, 61–91. https://doi.org/10.1007/BF02769620.
Panzeri, M., Della Rossa, E.L., Dovera, L., Riva, M., Guadagnini, A., 2016. Integration of Markov mesh models and data assimilation techniques in complex reservoirs. Comput. Geosci. 20, 637–653. https://doi.org/10.1007/s10596-015-9540-5.
Parker, H.M., 2012. Reconciliation principles for the mining industry. Trans. Institutions Min. Metall. Sect. A Min. Technol. 121, 160–176. https://doi.org/10.1179/1743286312Y.0000000007.
Rosa, L., David, Valery, W., Wortley, M., Ozkocak, T., Pike, M., 2007. The use of radio frequency ID tags to track ore in mining operations. In: Proceedings of the 33rd Application of Computers and Operations Research in the Mineral Industries, pp. 601–606.
Sambridge, M., 1999. Geophysical inversion with a neighbourhood algorithm–II. Appraising the ensemble. Geophys. J. Int. 138, 727–746. https://doi.org/10.1046/j.1365-246X.1999.00900.x.
Sarma, P., Durlofsky, L.J., Aziz, K., Chen, W.H., 2006. Efficient real-time reservoir management using adjoint-based optimal control and model updating. Comput. Geosci. 10, 3–36. https://doi.org/10.1007/s10596-005-9009-z.
Schulze-Riegert, R., Ghedan, S., 2007. Modern techniques for history matching. In: International Forum on Reservoir Simulation, pp. 9–13.
Soares, A., Nunes, R., Azevedo, L., 2017. Integration of uncertain data in geostatistical modelling. Math. Geosci. 49, 253–273. https://doi.org/10.1007/s11004-016-9667-5.
Sutton, R., Barto, A., 2017. In: Reinforcement Learning: an Introduction, Second. MIT press.
Tang, M., Liu, Y., Durlofsky, L.J., 2019. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. arXiv Prepr. 1–47. http://arxiv.org/abs/1908.05823.
Uhlenbeck, G.E., Ornstein, L.S., 1930. On the theory of the brownian motion. Phys. Rev. 36, 823. https://doi.org/10.1143/JPSJ.13.1266.
Vargas-Guzmán, J.A., Dimitrakopoulos, R., 2002. Conditional simulation of random fields by successive residuals. Math. Geol. 34, 597–611. https://doi.org/10.1023/A:1016099029432.
Vo, H.X., Durlofsky, L.J., 2014. A new differentiable parameterization based on principal component analysis for the low-dimensional representation of complex geological models. Math. Geosci. 46, 775–813. https://doi.org/10.1007/s11004-014-9541-2.
Wambeke, T., Benndorf, J., 2018. A study of the influence of measurement volume , blending ratios and sensor precision on real-time reconciliation of grade control models. Math. Geosci. 50, 801–826. https://doi.org/10.1007/s11004-018-9740-3.
Wambeke, T., Benndorf, J., 2017. A Simulation-based geostatistical approach to real-time reconciliation of the grade control model. Math. Geosci. 49, 1–37. https://doi.org/10.1007/s11004-016-9658-6.
Wambeke, T., Elder, D., Miller, A., Benndorf, J., Peattie, R., 2018. Real-time reconciliation of a geometallurgical model based on ball mill performance measurements – a pilot study at the Tropicana gold mine. Min. Technol. 127, 115–130. https://doi.org/10.1080/25726668.2018.1436957.
Xu, T., Gómez-Hernández, J.J., 2015. Inverse sequential simulation: performance and implementation details. Adv. Water Resour. 86, 311–326. https://doi.org/10.1016/j.advwatres.2015.04.015.
Xu, T., Jaime Gómez-Hernández, J., Li, L., Zhou, H., 2013. Parallelized ensemble Kalman filter for hydraulic conductivity characterization. Comput. Geosci. 52, 42–49. https://doi.org/10.1016/j.cageo.2012.10.007.
Yao, L., Dimitrakopoulos, R., Gamache, M., 2021a. Training image free high-order stochastic simulation based on aggregated kernel statistics. Math. Geosci. 1–21. https://doi.org/10.1007/s11004-021-09923-3.
Yao, L., Dimitrakopoulos, R., Gamache, M., 2021b. Learning high-order spatial statistics at multiple scales: a kernel-based stochastic simulation algorithm and its implementation. Comput. Geosci. 149, 1–11. https://doi.org/10.1016/j.cageo.2021.104702.
Yao, L., Dimitrakopoulos, R., Gamache, M., 2018. A new computational model of high-order stochastic simulation based on spatial Legendre moments. Math. Geosci. 50, 929–960. https://doi.org/10.1007/s11004-018-9744-z.
Yüksel, C., Benndorf, J., Lindig, M., Lohsträter, O., 2017. Updating the coal quality parameters in multiple production benches based on combined material measurement: a full case study. Int. J. Coal Sci. Technol. 4, 159–171. https://doi.org/10.1007/s40789-017-0156-3.
Yüksel, C., Thielemann, T., Wambeke, T., Benndorf, J., 2016. Real-time resource model updating for improved coal quality control using online data. Int. J. Coal Geol. 162, 61–73. https://doi.org/10.1016/j.coal.2016.05.014.