Machine learning for groundwater pollution source identification and monitoring network optimization

Yiannis N. Kontos1,2 • Theodosios Kassandros2 • Konstantinos Perifanos3 • Marios Karampasis1 • Konstantinos L. Katsifarakis1 • Kostas Karatzas2

Received: 3 December 2021 / Accepted: 1 June 2022 / Published online: 26 June 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
The identification of the source in groundwater pollution is the only way to drastically deal with resulting environmental problems. This can only be achieved by an appropriate monitoring network, the optimization of which is prerequisite for the solution of the inverse modeling problem, i.e., identifying the source of the pollutant on the basis of measurements taken within the pollution field. For this reason, a theoretical confined aquifer with two pumping wells and six suspected sources is studied. Simulations of combinations of possible source locations, and hydraulic parameters, produce sets of measurement features for a $29 \times 29$ grid representing potential monitoring wells. Three sets of simulations are conducted to produce synthetic datasets, representing different groundwater pollution modeling methods. Features (input-X variables) coupled with respective sources (output-Y variables) are formulated in two different dataset formats (Types A, B) in order to train classification (random forests, multilayer perceptron) and computer vision (convolutional neural networks) algorithms, respectively, to solve the inverse modeling problem. In addition, appropriate feature selection and trial-and-error tests are employed for supporting the optimization of monitoring wells’ number, locations and sampling frequency. The methodology can successfully produce various sub-optimal monitoring strategies for various budgets.

Keywords Machine learning • Groundwater pollution • Source identification • Monitoring network • Convolutional neural networks • Modflow

1 Introduction

As the paper includes many technical terms and abbreviations, a relevant list is provided (Table 1). Moreover, Table 2 is an index of the files provided in supplementary material.

Groundwater pollution is a major problem worldwide. Water pollutants may originate from point or non-point sources. Point sources, like the ones considered in this paper, are characterized by the presence of identifiable, small-scale sources. Such sources can be leaking underground fuel tanks, landfills, septic systems, hazardous waste sites and leaks or spills of industrial chemicals at manufacturing facilities. Cost of groundwater pollution control and remediation methods, like pump-and-treat or hydraulic control, is high; its minimization is essential yet challenging [1, 2]. Moreover, when the cleanup process includes the simultaneous protection of operating pumping wells providing drinking or/and irrigation water, the complexity and level of required intervention increase. Timely pollution detection presupposes existence of a proper network of monitoring wells. A monitoring network is also necessary to identify the pollution source and apply “the polluter pays” principle [3], discouraging possible polluters.

Identifying source location and time history of a solute in groundwater is characterized as a problem of time inversion; the governing equations need to be solved...
backward in time. Modeling contaminant transport using reverse time is an ill-posed problem; the process is dispersive, hence irreversible, while solutions have discontinuous dependence on data and exhibit acute sensitivity to data errors [4]. For the groundwater contamination problem, the solution exists only when there are a perfect and consistent model and data that satisfy extremely restrictive conditions. Satisfaction of the stability criterion is a challenging task as numerical schemes are unstable for negative time steps. This renders the solution of contaminant transport problems backward in time impossible. Moreover, no method can bypass the intrinsic obstacle of the non-uniqueness of the solution. The stability and non-uniqueness criteria require assumptions about the nature of the unknown function, so that observation data suffice. Hence, stabilization or regularization methods are implemented for the conversion of the ill-posed to a properly posed problem. Specifically, the inverse problem of groundwater pollution identification requires additional information/assumptions, like a finite number of suspect sources or chemical fingerprints of the plume [5].

1.1 Groundwater inverse modeling

Groundwater inverse modeling is generally important and essential in field applications [6]. In the field, distributions of aquifer/flow/pollution properties are unknown; only sparse and noisy measurements of relevant parameters like hydraulic head, flow-rates, and concentration are available. Direct measurement is expensive and often not feasible; inverse modeling can give solutions with relatively few monitoring observations [7]. The unknown variables to be estimated are usually the hydraulic conductivity distribution, contaminant zone distribution, location of contaminant source(s) and time of its release, or a combination of those. The objective function to minimize is the error of field measured vs calculated values at various locations of the field. Constraints usually include upper and lower bounds of unknown parameters, while measured values usually include the time-dependent hydraulic head observations, concentrations at monitoring wells, and flow-rates.

Direct methods for groundwater inverse modeling exhibit reduced computational complexity but are problematic in groundwater transport due to measurement errors and nonlinearity in the transport reactions. Indirect, usually optimization-based, methods can be gradient-based and non-gradient-based. Gradient-based, specifically machine/deep learning (ML/DL) methods are discussed and implemented in this paper. Non-gradient approaches start with initial guesses for the unknown variables of the inverse problem, and all respective forward solutions are calculated and progressively improved based on rules. Non-gradient optimization methods can be global (e.g., genetic algorithms or GAs [8–11], simulated annealing, particle swarm optimization, ant colony optimization) or local (Nelder–Mead simplex method; Hooke–Jeeves method, Powell’s method). Hybrid methods have also been proposed [7]. These methods, like GAs, are really flexible in problem formulation, can handle discontinuities in the decision space, and can explore larger search spaces, resulting in a more robust search than that of gradient-based approaches.

Gradient-based methods (like conjugate gradients, steepest descent, Levenberg–Marquardt minimization, Gauss–Newton, sequential quadratic programming) use the
local gradient of the function under optimization to reach a locally better solution. They converge faster if: (a) the function is reasonably continuous and differentiable, (b) the search space is reasonably non-deceptive (not many local optima), and (c) quadratic assumption applies near the optimum. These criteria do not apply for many groundwater inverse problems, but gradient-based methods are popular [7], especially the ones falling under the category of ML. With the increase of the computational power over the years, ML’s arsenal was enriched with advanced techniques for hyperparameter optimization (Bayesian Optimization) in ensemble learning, like random forests (ensemble/bagging algorithms) and with a vast number of stochastic gradient descent-based optimization algorithms (e.g., Adam, AdaGrad, AdamW) for deep learning models, like multilayer perceptrons and convolutional neural networks (stochastic gradient descend-based optimization).

### 1.2 Source identification methods

#### 1.2.1 Early conventional methods

Various methods have been proposed and implemented for the source identification problem. The first group of methods is related to the heat transport inversion methods, as the physical/mathematical models of heat and mass transfer are similar. Early methods include optimization approaches using linear programming [12], or direct approaches, such as the quasi-reversibility method [13], regularization and stabilization [14], or space marching finite difference [15]. Other direct approaches include use of the Sobolev equations [16], the observer-based method [17], minimum relative entropy [18], Fourier series-based inverse technique [19], backward beam equation [20], and marching-jury backward beam equation [21]. Conventional optimization methods include the conjugate gradient

### Table 2 Index of files provided as Supplementary material

| Nr  | Filename                                    | Filetype  | Description                                                                 |
|-----|---------------------------------------------|-----------|-----------------------------------------------------------------------------|
| SM1 | SM1-Video1a_(Sim1-points).mp4               | video file| A video made by up to 2500 daily maps that presents particles’ trajectories, simulated by MovPo, that is actually one Type B dataset for the 9632nd out of 15,246 runs of Simulation 1 |
| SM2 | SM2-Video1b_(Sim1-plumes).mp4               | video file| A video made by up to 2500 daily maps that present plume’s movement, simulated by MovPo, that is actually one Type B dataset for the 9632nd out of 15,246 runs of Simulation 1 |
| SM3 | SM3-Video2_(Sim2).wmv                      | video file| A video made by up to 2500 daily concentration maps simulated by Flopy-Modflow that is actually one Type B dataset for the 9632nd out of 15,246 runs of Simulation 2 |
| SM4 | SM4-Video3_(Sim3).wmv                      | video file| A video made by up to 2500 daily concentration maps simulated by Flopy-Modflow that is actually one Type B dataset for the 875th out of 8100 runs of Simulations 3 |
| SM5 | SM5-Sim1-TypeA-results-sample.xlsx          | excel file| An excel file with four sheets presenting a sample of the Type A datasets produced in Sim1. Each sheet corresponds to one of the four scenarios. The union of the four datasets would constitute Scenario 5’s Type A dataset |
| SM6 | SM6-Sim2-TypeA-results-sample.xlsx          | excel file| An excel file with four sheets presenting a sample of the Type A datasets produced in Sim2. Each sheet corresponds to one of the four scenarios. The union of the four datasets would constitute Scenario 5’s Type A dataset |
| SM7 | SM7-Sim3-TypeA-results-sample.xlsx          | excel file| An excel file with four sheets presenting a sample of the Type A datasets produced in Sim3. Each sheet corresponds to one of the four scenarios. The union of the four datasets would constitute Scenario 5’s Type A dataset |
| SM8 | SM8-Sim1-TypeB-results-example-RUN_09632.zip| zip file  | Compressed folder containing the one Type B dataset of 2500 files (days/runs/frames) that corresponds to Sim1-Scenario 1-run 9632 (out of 15,246) |
| SM9 | SM9-Sim2-TypeB-results-example-RUN_09632.zip| zip file  | Compressed folder containing the one Type B dataset of 2500 files (days/runs/frames) that corresponds to Sim2-Scenario 1-run 9632 (out of 15,246) |
| SM10| SM10-All_Metrics.xlsx                       | excel file| An excel file with 30 sheets, presenting: a) all Classification methods-subsets (RF-FL, RF-BF, RF-GS, MLP-FL, MLP-BF, MLP-GS) metrics for all Simulations (Sim1-3) and Scenarios (1–5); b) all Computer Vision (CNN-FL, CNN-BF, CNN-GS) metrics for all simulations (Sim1, 2, 3), Scenarios (1–5), for the 80-d sampling interval. Metrics include: Accuracy, Precision, Recall, F1-score, AUC, confusion matrix |
| SM11| SM11-Min_Costs.xlsx                         | excel file| An excel file with 3 sheets, presenting best monitoring network strategies for all Simulations (Sim1-3) and Scenarios (1–5). Strategies are organized based on their monitoring network’s construction and operation depreciated annual costs for 2 price cases. The solutions presented are the ones with the larger sampling intervals that exhibit Min Recall > 0.9 |
method with minimization procedures [22], linear programming and multiple regression [23], nonlinear maximum likelihood estimation [24], and integer programming [25]. Researchers have also used probabilistic and geostatistical simulation: reverse time random walk particle method [26], stochastic differential equations backward in time [27], and Bayesian theory and geostatistical techniques [28]. Finally, analytical solutions [29] and regressions (e.g., nonlinear least square method [30]) were used.

Table 3 presents the conventional methods mentioned, categorizes them, and together with Table 4, delineates their main limitations, explaining whether or not these limitations apply to the proposed methodology.

### 1.2.2 Optimization with metaheuristics

Source detection is an optimization problem that in recent years has been solved with various modern metaheuristic methods like GAs. Chadalavada et al. [32] presented an overview and discussion of pollution source identification optimization approaches. Early research efforts include Aral and Guan [33] and their improved GAs, and Aral et al. [34] with their progressive GAs. Mahinthakumar and Sayeed [7] used a hybrid GA-local search algorithm to increase the computational efficiency avoiding local minima. Han et al. [35] addressed the identification of small-area site pollution sources, using a small amount of groundwater management monitoring data; they linked GAs and advection–dispersion equation (ADE) for mass transport. Li et al. [36] used a hybrid Homotopy-GA to avoid premature convergence and local optima. They implemented a 0–1 mixed integer nonlinear optimization model based on a kriging surrogate model to simultaneously identify hydraulic conductivity, location, and release history of pollution sources; they combined it with GAs and outperformed GAs alone in computational speed. There were also many efforts with use of particle swarm algorithms [37], tabu search-simulated annealing [38], evolutionary strategy [39] and other metaheuristics. Sun and You [40] combined particle swarm optimization and GAs to decrease computational time in accurate identification of one or two pollution sources.

### 1.2.3 Machine and deep learning methods

Machine and deep learning methods have also been used; Shen [41], Rajabi et al. [42] and Sit et al. [43] provided reviews of ML/DL in water resources and hydrology. Singh and Datta [44] and Singh et al. [45] used a feed forward multilayer artificial neural network (ANN) to identify the unknown pollution sources and to simultaneously estimate aquifer parameters; they did so, even in
scenarios with part of concentration measurement data missing [46]. ANN was trained and tested to identify source characteristics based on simulated contaminant concentration measurements data at specified observation locations for various random pollution source fluxes. Perturbed measured concentration values were also used with satisfactory results with moderated level of uncertainty. As the number of potential sources increased, accuracy of identification decreased, especially with increased noise in measurements. Ayaz et al. [47] developed a linked ANN-optimization model for identification of source location and release period of an unknown groundwater pollution source. The objective function to be minimized required spatial and temporal data of observed and simulated concentrations, as well as the lag time, that was not known. An ANN model with a single hidden layer was trained using Levenberg–Marquardt algorithm to find the lag time required for the minimization of the objective function. Combinations of source locations and release periods were used as inputs and lag time was produced as the output. Evaluation was done for 2D and 3D cases with error-free and erroneous data. The model required only the upper half of the breakthrough curve to predict source parameters when the lag time was not known.

Mo et al. [31] developed a deep autoregressive neural network-based surrogate method for the forward model to solve efficiently a high-dimensional inverse problem: simultaneously identify a groundwater contaminant source and the permeability in a heterogeneous aquifer, training the surrogate with limited evaluations of the forward model. The problem was treated like an image-to-image regression problem, utilizing a dense convolutional encoder-decoder network architecture. An iterative local updating ensemble smoother algorithm was used as the

| Limitations | Do the limitations apply to the present methodology? |
|-------------|-----------------------------------------------------|
| L1 No advection | No; simulation 1 considers only advection; Simulations 2 and 3 consider advection and mechanical dispersion |
| L2 Poor performance more than 50% back in time | No; there are no limitations of this order found here |
| L3 Displayed spurious negative values | Not applicable (N/A) |
| L4 Accuracy decreases with increasing number of unknown parameters | N/A; this will be studied in future research |
| L5 Homogeneous parameters | Not tested here but relevant research has proven DL can deal with aquifer heterogeneity [31] |
| L6 Perturbation parameter is not known a priori | N/A |
| L7 Introduces phase-lag | N/A |
| L8 Sensitive to noise | Not tested. Inducing Gaussian noise a priori in the hydraulic simulations’ input data (ML) or a posteriori with data augmentation techniques in CNN is part of future research |
| L9 Results are highly sensitive to the accuracy of the inferred location | N/A; potential sources are known a-priori in this problem versions and the exact location of the source is investigated |
| L10 Excessive computational effort | Partially YES; this is quite relevant as the production, pre-processing and post-processing of data are the time-consuming processes, that only occur once; the ML/ DL algorithms can then be used directly in a real-time operational fashion |
| L11 Partial recovery | N/A; potential sources are known a-priori in this problem versions and the exact location of the source is investigated |
| L12 Stabilization parameter not easy to predict a priori | N/A |
| L13 Spill incidents are assumed to be instantaneous and occurring simultaneously | Leakage incidents are assumed to be instantaneous, meaning they last one timestep, in the first two simulations, but the same levels of accuracy can be produced for any known a priori duration. Only one pollution source is investigated here, more will be investigated in future research. Simulation 3 deals with unknown leak duration |
| L14 Dispersion part kept positive | No |
| L15 Potential source must be known a priori | Yes; the more the suspected polluters, the larger the datasets needed, disproportionately increasing the computational load |
| L16 Simple geometries and flow conditions | No; simple geometries and flow conditions are simulated in the current problem versions, but Modflow software can simulate any complex configuration, at the expense of computational effort |
inversion framework, and evaluation was done with synthetic datasets. Zhang et al. [48] went a step further, developing a surrogate-based Bayesian inversion framework to quantify and minimize the approximation error of the surrogate; most surrogate-based methods reduce the computational load, but neglecting the approximation error, leading to biased inversion results. Two strategies for calculating the surrogate’s uncertainty prediction were used: one Bayesian, and another, also surrogate, integrating polynomial chaos expansion and Gaussian process in three cases exhibiting high dimensionality, multimodality, and a real-world application. Both strategies yielded good results reducing bias by surrogate approximation error.

### 1.3 Monitoring network optimization

The inverse problem of pollution source identification is inextricably linked with the monitoring network in real-time applications, but also in optimal design. Evaluating remediation techniques and assessing environmental compliance requires time- and cost-wise expensive data collection efforts. Pollution source identification is an extremely difficult task without well-defined monitoring locations and sampling schedule capable to characterize contaminant spread within satisfactory accuracy levels. Moreover, installation and operation costs of the monitoring network may be quite high; thus, their minimization is essential.

Optimal design of a groundwater pollution monitoring network (GPMN) is necessary due to uncertainty in predicting the contaminant transport, and constrained budgets. Comprehensive reviews of monitoring network design are reported in literature [49–52]. GPMNs, dedicated to increasing the efficiency of the source identification models, were studied by many researchers; each one considered various objectives and implemented various optimization methods including: (a) maximization of detection possibility [53]; (b) minimization of number (Nr) of monitoring wells [54]; (c) minimization of undetected concentrations [55]; (d) minimization of contamination estimation variance [56]; (e) minimization of uncertainty (square root of estimation variance [57, 58]); (f) variance reduction with Kalman filter approach [59]; (g) minimization of monitoring cost [54, 57, 60, 61]; (h) minimization of squared deviation of estimated concentration from the actual one [57, 60]; (i) minimization of mass estimation [58, 61–63]; (j) minimization of error in locating plume centroid [58, 61, 63]. Some researchers combined many objectives using high-order Pareto optimization [58]. The optimization algorithms used included: integer programming [64]; mixed integer programming [55]; simulated Annealing [56, 65]; GAs [59–62, 65]; ant colony optimization [66]; Monte Carlo [67].

### 1.4 Problem outline and research goals

To the best of our knowledge, the GPMN optimization has never been approached with ML/DL methods. This paper aims at implementing ML/DL methods, already proven robust at environmental engineering problems [68], to

| Table 5 The different configurations of the three simulations implemented (Sim1, Sim2, and Sim3) |
|---|---|---|---|---|
| Sim | Scenarios | Input-X variables | Hydraulic simulations |
|---|---|---|---|---|
| 1 | 1, 2, 3, 4, 5 | $6 \times S + 21 \times J + 11 \times Q_1 + 11 \times Q_2 = 15246$ | Analytical Darcy equation | Lagrangian PTM | Instantaneous (LD = 1 $\Delta T$) | MovPo (Visual Basic) [1, 2] |
| 2 | 1, 2, 3, 4, 5 | $6 \times S + 21 \times J + 11 \times Q_1 + 11 \times Q_2 = 15246$ | FDM | FDM-ADE | Instantaneous (LD = 1 $\Delta T$) | MLdata-ModFloPyer (Flopy-Modflow in Python) [70] |
| 3 | 1, 2, 3, 4, 5 | $6 \times S + 3xJ + 50 \times LD + 3 \times Q_1 + 3 \times Q_2 = 8100$ | FDM | FDM-ADE | Varying duration (LD = 10–500; step 10) | MLdata-ModFloPyer (Flopy-Modflow in Python) [70] |

*aScenario 5 = Scenarios 1–4 combined

*S = sources; J = N–S hydraulic gradient; $Q_{1,2}$ = flow-rate of pumping well PW1,2; LD = pollution leak duration

‘FDM = finite differences’ method

‘PTM = particle tracking method; ADE = advection-dispersion equation

$\Delta T$ = timestep
achieve the following goals in a contaminant source detection problem: (a) identify the source location among given potential locations, (b) further investigate indirect optimization of the GPMN, implying (1) minimization of the number (Nr) of monitoring wells (MWs), (2) optimization of locations of MWs, (3) optimization of sampling schedule (min frequency) through feature selection methods and trial-and-error tests. The methods used are evaluated concerning their robustness, training speed, and accuracy toward optimization of the GPMN. In brief, short-term goal is: identify source; long-term is: optimize monitoring design, simultaneously solving the real-time/operational problem (which means, when proposing a monitoring scheme, also provide the respective/tailored to the scheme’s source prediction algorithm).

Three simulation series are implemented (Table 5): (a) Sim1 investigates source identification for a conservative pollutant, simulating advective (only) mass transport with “MovPo” (a tool of the “OptiManage” optimization software created by some of the authors [1, 2]); contaminant release of one of the potential sources is assumed instantaneous; (b) Sim2 investigates the same problem, but simulating advection–dispersion mass transport with Modflow software of the USGS suite [69]; (c) Sim3 resembles Sim2, but also considering unknown contaminant release duration.

Specifically, a known theoretical aquifer is studied (Fig. 1). Two irrigation/drinking water pumping wells (PWs) near the southern boundary, together with a north–south natural flow (N–S hydraulic gradient), define the flow. Six suspected possible sources (S1–S6), capable of instantaneous pollution leakage (contaminant release) in four different 6-source layouts (Scenarios 1–4) are considered. Each of the four scenarios holds six distinct sources. No scenario of 1–4 shares common sources with another one. A more complex scenario is also assumed, Scenario 5, which is nothing more than the combination/union of Scenarios 1–4 (6 × 4 = 24 suspect pollution sources). 29 × 29 = 841 inner field grid nodes (50 m cell size) serve as possible locations of sources, PWs and monitoring wells (MWs). The simplified problem version in this pilot approach is based on the following assumptions: (1) time \( t = 0 \) of initial pollution leakage is known, regardless of the source, (2) in Simulations (Sim) 1 and 2, leakage of the same amount of pollutant occurs instantly (during a timestep \( DT \)), while in Sim3, leakage occurs continuously for 10 to 500 10-day timesteps, (3) initially, all nodes bear MWs that record: a) yes/no pollutant detection (IsMWPol), (b) 1st day of pollution (DayPol), (c) pollution duration (PolDur), (d) hydraulic head drawdown (Drawdown; only for Sim1), e) pollutant concentrations at all MWs (only for Sim2 and 3). Features (a) relate to a single sampling at a predefined day and in situ/ex situ analyses. Features (b) relate to floating/fixed depth (low-cost smart) sensors sending a single signal upon pollution detection (e.g., measuring electric current variance) or consecutive manual samplings in each \( AT \), followed by in situ/ex situ analyses upon pollution detection. Features (c) imply continuation of the remote or manual measurements after first pollution detection. Features (d) entail a manual one-off drawdown measurement (steady-state flow) or, in case of an existing sensor, a fixed depth sensor-strip
solution (instead of floating sensor) that can also detect drawdown.

Practical research goals are: find pollution source for 6-source Scenarios 1–4, and the merged 24-source Scenario 5, using various ML/DL methods and evaluate them based on their accuracy metrics; decide on the most suitable prediction metric for the nature of the studied problem; conclude on the useful MWs/features to remove useless ones, a first indirect step toward the optimization of MW network; investigate each feature’s importance.

Fig. 2 Structure (flowchart) of the proposed methodology that includes 5 scenarios; each scenario is associated with three separate simulations (Sim 1, 2, 3); each simulation follows the exact same steps.
further decreasing their Nr (feature selection), with various techniques, while retaining the same accuracy levels. Given the specific spatial layout of the MW network, investigate further indirect MW network optimization (monitoring cost minimization), searching for the lowest temporal discretization (lowest sampling frequencies) that can provide unchanged source prediction accuracy metrics for all scenarios. This way, a novel unconventional methodology for GPMN optimization can be produced. Finally, evaluate each ML/DL method, concluding on (a) pros/cons (including training speed), (b) significance of features tested, (c) formulation of input–output (X–Y) datasets suitable for each ML/DL method for the ultimate next step/future goal: include best methods in a metaheuristic optimization algorithm that can automatically optimize the MW network (MW locations, sampling rates/time-instances, strategies).

2 Methodology

The proposed methodology and the Nr of simulations, scenarios and implementations are presented in the form of a flowchart in Fig. 2.

In summary, four different layouts of the six possible pollution source locations constitute four different scenarios (1–4), plus the merged Scenario 5. Each scenario is associated with three simulations (Table 5). Sim1 and 2 actually refer to a similar problem with different synthetic data production method. Sim3 is a more complex version of the problem with added variable contaminant release (pollution leakage) duration. Each simulation is associated with a separate synthetic data production process. The proposed GPMN optimization methodology requires the formulation of produced synthetic data in two different forms: Type A and B. Type A formulation facilitates the implementation of machine learning-classification algorithms, random forests and multilayer perceptron. It also facilitates the use of the respective correlation-based feature selection methods, best first and greedy stepwise, in order to select the smaller feature subsets that can guarantee accurate predictions. These methods are utilized to indirectly spatially optimize the monitoring scheme (Nr and locations of monitoring wells, minimizing construction costs). That way ML methods are used to predict the source, succeeding in the first research goal with ML, but also take the first step toward producing optimal monitoring strategies. Since this first indirect optimization step does not include the temporal dimension, the so far proposed monitoring strategies entail use of the initial full dataset (time-wise), namely daily sampling (1-d). On the other hand, Type B data formulation facilitates the implementation of deep learning–computer vision algorithms, here, convolutional neural networks. It also facilitates the use of trial-and-error tests of various sampling intervals, indirectly temporally optimizing the monitoring scheme, by minimizing the sampling frequency (and sampling costs). These tests, together with the use of the previously produced feature subsets, lead to an unconventional indirect optimization of the GMPN, minimizing both initial (construction and equipment) and operational (sampling) costs. All relevant details are provided in the following paragraphs.

2.1 Flow field and mass transport simulation for synthetic data production

In order to balance the vast computational load needed for production of simulated synthetic datasets to feed data-driven methods, the simulation of a simplified surrogate 2D flow field is selected. The 1500 m × 1500 m theoretical flow field (Fig. 1) includes two pumping wells of known

### Table 6 Values, ranges and Nr of values used, concerning flow field variables/parameters that produce the combination of problem instances to be simulated (Scenarios 1–4)

| Variable*  | Sim1 + Sim2 | Sim3 |
|------------|-------------|------|
|            | Min | Max | Step | Nr values | Min | Max | Step | Nr values |
| Source     | 1   | 6   | 1    | 6         | 1   | 6   | 1    | 6         |
| J (‰)      | 0   | 2   | 0.1  | 21        | 0   | 2   | 1    | 3         |
| Q1 (L/s)   | 215  | 225 | 1    | 11        | 225 | 5   | 3    |           |
| Q2 (L/s)   | 225  | 235 | 1    | 11        | 225 | 5   | 3    |           |
| b (m)      | 50   | 50  | 0    | 1         | 50  | 0   | 1    |           |
| K (m/s)    | 10^{-5} | 10^{-4} | 0    | 1         | 10^{-4} | 0   | 1    |           |
| Por        | 0.2  | 0.2 | 0    | 1         | 0.2 | 0.2 | 0    | 1         |
| LD (d)     | 1    | 1   | 0    | 1         | 10  | 500 | 10   | 50        |
| Runs =     | 15,246 |     |     |           |     |     | 8100  |           |

*J, N–S hydraulic gradient; Q1,2: flow-rate of pumping well PW1,2; b: aquifer thickness; K: hydraulic conductivity; Por: porosity; LD: pollution leak duration
varying flow-rates during a year, with an inter-annually varying north–south natural flow also affecting the flow field. The confined aquifer is assumed homogeneous and isotropic with a plane, horizontal, single-phase steady flow. It is assumed infinite in Sim1, while in Sim2 and 3, boundary conditions apply. Combination of the constant/varying values of flow field parameters (Table 6) produces a 15,246 dataset package of cases per scenario for Sim1 and Sim2 and an 8100 dataset package of cases per scenario for Sim3. For Scenario 5 (merged Scenarios 1–4) datasets’ size is 15,246 × 4 = 60,984 for Sims1 and 2 and 8100 × 4 = 32,400, for Sim3.

2.1.1 Particle tracking method for mass transport with MovPo (for Sim1)

In Sim1, pollutant is assumed to spread dominantly via advection, with a Lagrangian particle tracking method.

Table 7 Data and results of Fig. 2’s right case as produced by “MovPo” software’s 9632nd out of 15,246 runs (Sim1—Scenario 1)

| RUN info (Sim1—Scenario 1) | Monitoring wells’ pollution results |
|---------------------------|-----------------------------------|
| RUN count                | 9632/15,246                       | Rank | *fast to slow pollution incident | IsMWPo | DayPo | DurPo |
| Source Nr                | 4                                 | 0    | 691 | 1 | 35 |
| Source node              | 691                               | 1    | 662 | 58 | 54 |
| PW1 node                 | 35                                | 2    | 603 | 242 | 44 |
| PW2 node                 | 111                               | 3    | 574 | 305 | 69 |
| N–S hydraulic grad. (%)  | 1.6                               | 4    | 545 | 381 | 61 |
| Q1 (L/s)                 | 221                               | 5    | 457 | 607 | 43 |
| Q2 (L/s)                 | 231                               | 6    | 428 | 654 | 668 |
| General results of simulation | 7                           | 399 | 712 | 65 |
| Nr of MWs polluted       | 10                                | 8    | 370 | 778 | 37 |
| DayPo PW1                | 9                                 | 9    | 225 | 995 | 60 |
| DayPo PW2                | 1073                              | 10   | 111 | 1073 | 1 |
(PTM) simulating mass transport [71]. “MovPo” software used (Visual Basic) is part of “OptiManage” optimization suite, created by some of the authors [1, 2]. Previous research offers guidelines to define best parameter values, e.g., Nr of particles for circular plumes’ simulation, temporal discretization of study period, suitable PWs’ approximate capture zone [71]. The pumping well pollution criterion is based on a circular approximation of time-of-travel (during a $\Delta T$) capture zones. A pollutant particle $P$ is assumed to pollute a well $W$ during a certain $\Delta T$, if and only if the line segment simulating the displacement of $P$ during $\Delta T$ intersects the approximate capture zone of $W$ [2]. The additional simulation burden here is selecting the mathematical criterion of a MW being polluted, as MWs do not exhibit a capture zone in the previous sense. Point-in-polygon method (“ray casting algorithm” or “even–odd rule”) [72] is implemented, so that in each $\Delta T$ (day), the algorithm checks if any MW center is inside the (moving) polygon defined by the coordinates of 16 moving particles simulating pollution plume/source (initially “circular” or actually a perfect hexadecagon).

In order to build the datasets for the training/validation/evaluation of the ML/DL algorithms, “MovPo” simulates all 15,246 different source-flow field cases (Tables 5 and 6), running for 46.5 h (Intel Core i7 7700 @3.60 GHz; 16 GB RAM @1197 MHz). The more complex Scenario 5 (merged Scenarios 1–4) and its 60,984 cases in Sim1 is also simulated. For economy of space, Fig. 3 graphically presents the merged results of such two random Scenario 1 layout cases: $N$–$S$ grad = 1.6%, $Q_1 = 221$L/s, $Q_2 = 231$ L/s, and source = $S1$ (left) and $S4$ (right). The right case of Fig. 3 is also provided as two videos in supplementary material, consisting of up to 2500 daily maps that show particles’ movement (SM1) and plume’s movement (SM2), simulated by MovPo.

Left case is a classic PTM graphical representation, where 16 separate particle trajectories are calculated as line segments for consecutive (1-day) $\Delta T$s, checked for polluting any PW. Right case represents the added simulation of moving pollution front, where a hexadecagon’s displacements are calculated for consecutive $\Delta T$s, checked for polluting any MW. The selected polygons drawn represent only: all days that plume is located upon the node of $S4$, and 1st day of pollution for any polluted MW up to PW2 pollution. Zoom 1 is a left case magnified region, presenting the 16 symmetrically placed points (+) on the initially circular plume ($S1$; starting points for tracked particles). The nodes’ enumeration is shown: node 785 is $S1$ of Scenario 1. Zoom 2 is a right case magnified region, showing $S4$ (node 691) polluted for 35 days (35 polygons over node). Day 58, 1st day node 662 is polluted, is also shown.

Table 7 presents the respective results of Fig. 3, as calculated for the 9632nd out of 15,246 runs of Scenario 1 of Sim1. The first two columns provide dataset/run info: source Nr; respective node enumeration; PW node Nrs; flow/aquifer parameters; Nr of MWs polluted; PW(s) polluted (here only PW2); Nr of days needed (1073). The ensuing three columns present raw non-zero results: (a) “IsMWPol” (short for “Is Monitoring Well Polluted”) includes Nrs of the 10 nodes (representing Source/MW/PW) polluted in this run/case; (b) “DayPol” (Day of Pollution) refers to the first day pollution is detected in the respective node; (c) “DurPol” (Duration of Pollution) refers to the duration (days) of each node being polluted. This is just a way of presenting an example of Sim1 results; it does not correspond to the Type A or Type B data formulation.

2.1.2 Advection–dispersion mass transport simulation with Modflow (for Sims 2 and 3)

In Sim2 and Sim3, the pollutant is assumed to spread via advection and mechanical dispersion (molecular diffusion not considered). This is assigned to the Modflow 6 software of the USGS suite [69], a finite difference method (FDM) tool, already validated and established for simulation of the flow-field and mass transport in aquifers, among other things. The vast Nr of the required simulations (same as Sim1, 15,246 for Scenarios 1–4 and 60,984 for Scenario 5) for building the datasets, dictates the automation of the procedure; “Flopy: Python Package for creating, running, and post-processing MODFLOW-Based Models” is utilized [73–77].

For the proper simulation of the $29 \times 29$ grid of 50 m $\times$ 50 m cells, two external rows and columns are added in the perimeter. The external ones are used for the activation of general head boundaries (GHB) [78], a special version of constant boundary conditions; the inner ones ensure reduced computational errors linked to the neighboring cells of constant head boundaries, an inherent finite differences’ established weakness. In an attempt to make Sim2 as similar to Sim1 as possible, as far as the initial and boundary conditions and aquifer characteristics are concerned, the best way to simulate an infinite confined aquifer in Modflow must be found. As something like that cannot be explicitly defined in an FDM, a simple but effective trick is to induce a sense of infinite aquifer through the field boundaries via the GHB boundaries; contrary to the fixed constant head (CHD) boundaries, without increasing the grid size, thus the computational load, GHB are added but in a declared theoretical distance of 5 km from the grid perimeter. Comparison tests produced small (expected) divergence between the two hydraulic approaches (Sim1 vs Sim2), which is out of context for this paper.
Apart from the constant head boundaries, two objects for the two pumping wells are added to the Modflow project, with their flow-rates varying exactly as in Sim1 (Table 6). Moreover, objects of initial pollution are added (respective sources S1-S6 for each scenario), with initial leakage of 1000 ppm for 1 timestep (1 day; Sim2) or variable durations (Sim3). For the N–S natural flow simulation, the Northern GHB boundary (upper row of cells) is attributed a progressively higher constant head value while the southern one (lower row of cells) stays constant so that the produced head difference for the given distance results in the desired hydraulic gradients (see values of “N–S hydr. Grad. (J)” in Table 6). For continuity reasons, the western (left column of cells) and eastern (right column) GHB boundaries are assigned values in a linearly increasing fashion from the South to the North GHB. The pumping well pollution criterion is different than Sim1: a PW is considered to be polluted when the calculated by Modflow contaminant concentration in the respective cell exceeds 0.1 ppm. This arbitrarily selected value could realistically be the lowest pollutant concentration a laboratory test method could detect, if ex situ analysis is implemented, or the lowest concentration a sensor could detect (in situ).

In order to build the datasets for the training/validation/evaluation of the ML/DL algorithms, “Flopy-Modflow” simulates all $4 \times 15,246$ different source-flow field cases (Tables 5 and 6), running for 99 h for Sim2, and all $4 \times 8100$ cases running for 53 h for Sim3 (Intel Core i7 7700 @3.60 GHz; 16 GB RAM @1197 MHz). The more complex Scenario 5 (merged Scenarios 1–4) and its 60,984 cases in Sim2, and 32,400 cases in Sim3, is also simulated in a proportionally larger time period. Figure 4 presents results (a concentration map) of one of the datasets of Scenario 1 (right case) presented in Fig. 3 (Sim2; Scenario 1; source = S4; N–S grad = 1.6%o; $Q_1 = 221$ L/s; $Q_2 = 231$ L/s). The dataset graphically presented in Fig. 4 is also provided as a video in supplementary material (SM3), consisting of up to 2500 daily concentration maps, simulated by Flopy-Modflow. Another dataset of Sim3 is provided as a video in supplementary material; SM4, made by up to 2500 daily concentration maps, simulated by Flopy-Modflow, is actually the 875th out of 8100 runs of Sim3 (Scenario 1; S1, N–S hydr. grad. = 1%o; $Q_1 = 225$ L/s; $Q_2 = 235$ L/s; leak duration = 250 d). The first day PW2 is polluted, that is the first day contaminant concentration exceeds 0.1 ppm in one of the PWs.

Fig. 4 Sim2 sample result: contaminant concentration map of day 286 (1st day that cell/node 111 = PW2 records over 0.1 ppm of pollution); that is run 9632/15,246 of Scenario 1 (S4, N–S grad = 1.6%o; $Q_1 = 221$ L/s; $Q_2 = 231$ L/s); red circle = source; green circles = PWs.
(here PW2 represented by cell 111), is day 286. The difference between DayPol(111) = 1073 for Sim1 and DayPol(111) = 286 for Sim2 originates from the difference in pollutants and mass transport simulation: Sim1 simulates only the advective pollutant front’s transport, while Sim2 additionally simulates mechanical dispersion that causes plume spread, and faster detection of the dispersive pollution front, so the well pollution criterion changes from “Sim1: what day is the polygon representing the pollution plume over cell 111 for the first time?” to “Sim2: what day is the calculated concentration in cell 111 over 0.1 ppm for the first time?”.

As the pollution spreads more in Sim2 due to dispersion, the results of Fig. 4 (Sim2; Scenario 1; S4; N–S grad = 1.6%; Q1 = 221 L/s; Q2 = 231 L/s) in Table 7 cannot be graphically presented like Sim1’s in Table 7. The two first columns would be the same, except for the Nr of MWs polluted, which is now 228.
instead of 10, and 1st day of pollution (DayPol PW2), which is now 286 instead of 1073. The results are presented in grid-like tables in Fig. 5: (a) IsMWPol; (b) DayPol; (c) DurPol in days, with an ascending values’ color scale from white to red. This is just a way of presenting an example of Sim2 results (Sim3 is similar) and does not correspond to the Type A or Type B data formulation.

### 2.2 Machine/deep learning implementation

Two basic ML/DL approaches are implemented: classification (CL) and computer vision (CV). In CL, random forests (RF [79]) and multilayer perceptron (MLP [80]) are tested. RF consists of a population of classification trees (forest), which are trained and then used as an ensemble, i.e., each tree, which is trained independently, using a random sample of the data contributes to the final result via weighted voting. MLP is a feedforward artificial neural network that maps sets of input data onto a set of appropriate outputs, consisting of three layers (input, output and one hidden layer). Many features introduce noise in data-driven models and can lead to reduced model performance; simplifying the modeling procedure is better in terms of interpreting the model results and reducing computational time. Additionally, as each feature represents a MW, reducing features leads indirectly to the optimization of the monitoring network. For that, correlation-based feature subset selection methods (CFS) [81] are used. They measure the correlation between nominal features and evaluate the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them; they then divide it with the correlation of this specific feature to all other features in the feature space. In CV, a convolutional neural network (CNN) is tested. CNNs [82] are very popular for image recognition and segmentation tasks. The U-net architecture, a modified fully convolutional network, [83] proposed for biomedical image segmentation, is utilized.

Each technique is implemented for the three simulation series: (a) Simulation 1—MovPo and instantaneous leak; (b) Simulation 2—Modflow and instantaneous leak; (c) Simulation 3—Modflow and varying duration leak (Table 5; Fig. 2). For evaluation purposes accuracy, precision, recall, F1-score, the area under the receiver operating characteristic curve (AUC) and confusion matrix are calculated for each prediction. The first four are given as:

\[
\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}
\]

\[
\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}
\]

\[
\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}
\]

\[
F1\text{-score} = \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where TP is the Nr of true positives; TN is the Nr of true negatives, FN is the Nr of false negatives; and FP is the Nr of false positives.

As far as the studied problem is concerned, the nature of the problem dictates the use of the strictest metric as a criterion of consistently accurate and safe predictions. The reason is that a failed prediction can endanger public health if water is pumped for drinking or even irrigation purposes. The selected metric to play the role of the strictest prediction criterion is the minimum value of Recall of all classes, further on referred to as “Min Recall.” The conversion of the problem into a classification one, the respective classes, and an example of why Min Recall is the strictest and safest metric/criterion, are presented later on in the text. Practically, Min Recall describes the worst performance of the model for each possible pollution source, and it is the safest measure for a modeling procedure in which public health is involved.

#### 2.2.1 Classification (machine learning)—random forests, multilayer perceptron

In CL approach, time dimension and spatial correlation of MWs are not considered, hence the simple formulation of the datasets (Type A).
Simulation 1—MovPo and instantaneous leak: Each simulated combination (see Table 6) produces a single dataset and is printed in a single row in the Scenario-specific results’ file (for file structure, see Table 8; for dataset sample, see SM5). The dataset presented in Table 7 is actually the 9632nd dataset of a table like Table 8 (SM5, Sheet “Sim1-Scenario1,” row 15), for Sim1-Scenario 1. The target variable (output-Y), as far as the prediction is concerned, is a discrete class, with nominal values from 1 to 6 (source Nr). The four types of input-X variables are: (a) whether node i is polluted at all (IsMWPol = 1 or 0), (b) day Nr that node i is polluted since the leakage start (DayPol, = 0–2500; integer), (c) duration of node i pollution (DurPol, = 0–2500; integer), and (d) hydraulic head drawdown at node i (Drawdown, ≥ 0; real). The initial Type A data package is a (15,246 cases) rows × (4 features × (29 × 29) grid + 1 y) columns = 15,246 × 3365 matrix per scenario for Scenarios 1–4. The initial Type A data package for Scenario 5 (merged Scenarios 1–4) is a (15,246 cases × 4 scenarios) rows × (4 features × (29 × 29) grid + 1 y) columns = 60,984 × 3365 matrix.

Simulation 2—Modflow and instantaneous leak: Just like Sim1, each simulated combination (see Table 6) produces a single dataset and is printed in a single row in the Scenario-specific results’ file (for dataset sample, see SM6). The dataset presented in Fig. 5 is actually the 9,632nd dataset of a table like Table 8 (SM6, Sheet “Sim2-Scenario1,” row 15), for Sim1-Scenario 1. The configuration is similar to Sim1 (classes, input-X and output-Y variables etc.,) except for the Drawdown values that are not used (as explained in the respective results’ section). Input-X and output-Y variables are the same and Type A data formulation is used. The initial Type A data package is a (15,246 cases) rows × (3 features × (29 × 29) grid + 1 y) columns = 15,246 × 2526 matrix per scenario for Scenarios 1–4. The initial Type A data package for Scenario 5 (merged Scenarios 1–4) is a (15,246 cases × 4 scenarios) rows × (3 features × (29 × 29) grid + 1 y) columns = 60,984 × 2526 matrix.

Simulation 3—Modflow and varying duration leak: Just like Sim1 and Sim2, each simulated variable combination (see Table 6) produces a single dataset and is printed in a single row in the Scenario-specific results’ file (for dataset sample, see SM7). The configuration is similar to Sim2 (classes, input-X and output-Y variables, etc.,) with some differences due to the different values N–S hydraulic gradient, Q1 and Q2 can receive, and due to the previously fixed, now varying value of leak duration (see Tables 5 and 6). Input-X and output-Y variables are the same, and Type A data formulation is used. The initial Type A data package is a (8100 cases) rows × (3 features × (29 × 29)
**Fig. 6** a Sim1 (see SM8) and b Sim2 (see SM9) results of Fig. 3 right case (Scenario 1, $J = 1.6\%$; $Q_1 = 221$ L/s; $Q_2 = 231$ L/s; run 9632/15,246; see Table 7, Fig. 5); Type B data formulation for CV approach; batch of 2500 files of 29 x 29 frames; Sim3 resembles Sim2

**Fig. 7** Sim1 graphical representation of output- Y (target) input- X variables: a Y for S3, b X for S3, c Y for all 6 sources, d X for all 6 sources (Scenario 1, step = 1 day, t52–t309)
Fig. 8 Sim2 graphical representation of output-$Y$ (target) input-$X$ variables:
- $a$ $Y$ for $S3$
- $b$ $X$ for $S3$
- $c$ $Y$ for all 6 sources
- $d$ $X$ for all 6 sources (Scenario 1, step = 1 day, $t_{52}$–$t_{309}$)

Fig. 9 Sim3 graphical representation of output-$Y$ (target) input-$X$ variables:
- $a$ $Y$ for $S3$
- $b$ $X$ for $S3$
- $c$ $Y$ for all 24 sources
- $d$ $X$ for all 24 sources (Scenario 5, step = 1 day, $t_{52}$–$t_{309}$)
Simulation 1—MovPo and instantaneous leak: Each file $i$ out of the $N$ files produced in each dataset (1 ≤ $N$ ≤ 2500 days/runs) is a 29 × 29-sized matrix/frame containing “1” or “0” elements, depending on yes/no pollution of the specific node that specific day $i$. Day 1 frame contains only one “1” element in the source-node (all others being “0”), constituting the target variable (output-Y). All ensuing frames can be used as training data for the DL algorithm (input-X variables). Figure 6a graphically presents the single dataset/run presented in Table 7, but in Type B form (the real dataset of 2500 files/days/runs/frames is provided as SM8). The full Type B dataset package is a batch of 15,246 folders (cases) of 2500 files (days/29 × 29 frames with “0” or “1” elements) = 38,115,000 files/frames for each one of the five scenarios.

Simulation 2—Modflow and instantaneous leak: Each file $i$ out of the $N$ files produced in each dataset (1 ≤ $N$ ≤ 2500 days/runs) is a 29 × 29-sized matrix/frame containing pollutant concentrations in ppm (real positive numbers instead of binary like Sim1). Day 1 frame contains only one “1000” (ppm) element in the source-node (all others being “0”), constituting the target variable (output-Y). All ensuing frames can be used as training data for the DL algorithm (input-X variables). Figure 6b graphically presents the single dataset/run presented in Fig. 5, but in Type B form (real dataset of 2500 files/days/runs/frames is provided as SM9). The full Type B dataset package is a batch of 15,246 folders (cases) × 2500 files (days/29 × 29 frames with “0” or “1” elements) = 38,115,000 files/frames for each one of the 5 scenarios.

Simulation 3—Modflow and varying duration leak: The configuration is similar to Sim2, but there are differences due to the different values $N$–$S$ hydr. grad., $Q1$ and $Q2$ can receive, and due to the previously fixed, now varying value of leak duration (see Tables 5 and 6). The full Type B dataset package is now a batch of 8100 folders (cases) × 2500 files (days/29 × 29 frames with “0” or “1” elements) = 20,250,000 files for each one of the 5 scenarios.

Algorithm 2, the common process for source prediction with Computer Vision for each Scenario and Simulation (using Google Colab [86]), includes the following steps:

1. **Datasets’ temporal masking:** Not all produced 2500 frames per simulation are fed into the CNN. The leading frames, of days when source-nodes are polluted, must be concealed ($t$1 to $t$51, $t$49, $t$48, $t$47 for Scenarios 1, 2, 3, 4, respectively), as well as the latter frames, when PWs are polluted ($t$469, $t$782, $t$327, $t$310 to $t$2500 for Scenarios 1,2,3,4, respectively). Finally, the strictest common time window $t$52–$t$309 (258 days/frames) is selected for all simulations (including Scenario 5), for realism/uniformity/comparability reasons. As a result, input-X variables actually consist of the sum of all available frames; all available matrices are added together creating a super-frame/image (see Fig. 7 for Sim1, Fig. 8 for Sim2, and Fig. 9 for Sim3). Practically, Fig. 7, presents a graphical representation of output-Y (target) and input-X variables for Sim1: (a) $Y$ for $S_3$, (b) $X$ for $S_3$, (c) $Y$ for all 6 sources, (d) $X$ for all 6 sources (Scenario 1, step = 1 day, $t$52–$t$309). Figure 8 presents the respective graphical representation for Sim2, and Fig. 9 for Sim3. 29 × 29 frames are cropped into 28 × 28 ones, for proper CNN operation (29th column and 29th series deleted). Finally, the values are normalized to [0,1] interval, with a min–max scaler, resulting to the filtered dataset CNN-FL.

2. **Time-variant datasets:** Type B data formulation accommodates data manipulation in the temporal dimension: different versions of datasets are created by summing the frames every 1, 10, 20, ..., 80 days (nine time-variant data packages/scenario). Each image version corresponds to a different sampling (manual or sensors-telemetry based) frequency, leading to an indirect optimization of the time-dimension of the monitoring network.

3. **Import CFS subsets from CL:** Feature subsets (MWs to be constructed) selected by CFS methods in CL, are used to indirectly introduce spatial optimization of monitoring network in the temporal optimization of CV (Step 2). The use of subsets entails the respective masking of the finalized super-images (unselected elements in the matrices are given zero values). Finally, 3 subsets (filtered subset CNN-
FL + subsets CNN-BF and CNN-GS) combined with 9 temporal variations produce 27 datasets/scenario.

**Step 4 - Implementation:** For each one of the 27 datasets a CNN is implemented, using the U-Net segmentation architecture (Adam optimizer; learning rate = 0.0001; binary cross-entropy loss function; 500 epochs; batch size = 712) [83]. The CNN was built using the Tensorflow 2.0 library [87]; a set of test runs was initiated to find the simplest architecture and most suitable hyperparameters with the highest accuracy. The final architecture (Fig. 10) consists of two types of layers (Convolutions and Maxpooling) succeeding each other in the contracting path, followed by the expansive path which applies only Convolutions and copies of Convolutions from the contracting path, until the same dimensions of the initial image are achieved. The models are evaluated with the train-validation-test split method (60% train—20% validation—20% test). In order to (1) tackle the unbalanced binary dataset in

| Scenario | Subset | Nr MWs | Nr Feat | IsMWPol | DayPol | DurPol | Drawdown |
|----------|--------|--------|---------|---------|--------|--------|----------|
| 1        | FL     | 657    | 1086    | 143     | 143    | 143    | 657      |
|          | BF     | 6      | 4       | 2       | 0      | 0      |          |
|          | GS     | 23     | 55      | 17      | 18     | 20     | 0        |
| 2        | FL     | 658    | 958     | 100     | 100    | 100    | 658      |
|          | BF     | 6      | 6       | 6       | 0      | 0      |          |
|          | GS     | 21     | 51      | 21      | 20     | 10     | 0        |
| 3        | FL     | 657    | 1137    | 160     | 160    | 160    | 657      |
|          | BF     | 6      | 6       | 6       | 0      | 0      |          |
|          | GS     | 23     | 53      | 23      | 18     | 12     | 0        |
| 4        | FL     | 657    | 1122    | 155     | 155    | 155    | 657      |
|          | BF     | 6      | 6       | 5       | 0      | 1      | 0        |
|          | GS     | 21     | 43      | 21      | 10     | 12     | 0        |
| 5        | FL     | 507    | 2028    | 507     | 507    | 507    | 507      |
|          | BF     | 23     | 26      | 10      | 10     | 6      | 0        |
|          | GS     | 45     | 71      | 30      | 20     | 21     | 0        |

![Fig. 11](image-url) Graphical representation of Sim1 feature selection results (BF and GS subsets; see Table 9); these two layouts of the two respective MWs’ networks are adopted by the authors as two indirect proposals of spatially (sub)optimal monitoring network strategies.
the evaluation step, since the non-polluted cases are massively more than the polluted ones, (2) make the results comparable with the previous approach (Type A), and more importantly (3) validate the models in the most robust and meaningful way, the binary predictions are transformed to one of the initial six classes. The map of ‘0’ and ‘1’ (Sim1) or ‘0’ and ‘1000’ (Sim2 and 3) corresponds to the initial class (pollution Source), based on where the ‘1’ or ‘1000’ is located, and the Min Recall of all the classes is used as the evaluation metric.

3 Results and discussion

All calculated metrics for all scenarios (1–5) and simulations (1–3) are available in supplementary material (SM10).

3.1 Classification results

3.1.1 Classification Sim1 results

In Sim1, Algorithm 1’s Step 3 “Feature Selection” produces different feature subsets. Table 9 presents all relevant information for Sim1: Nr of MWs, total Nr of features, and the respective breakdown in exact feature types (IsMWPol; DayPol; DurPol; Drawdown). The specific layouts of the two respective MWs’ networks are adopted by the authors as two indirect proposals of spatially (sub)optimal monitoring network strategies (Fig. 11). The BF subsets propose the least Nr of MWs and features, two to four times less than GS. However, GS subsets offer a big decrease of MWs compared to the “full dataset,” represented by the FL subsets. Drawdown measurements are unsurprisingly never selected as important features; the simplified steady flow field does not vary depending on the source location. This is why drawdown results are not included in the datasets of Sim2 and Sim3.

A close inspection of the plethora of metrics produced for all scenarios and simulations (see SM10) leads the authors to the conclusion that Min Recall is the strictest and safest choice for the criterion/metric to evaluate predictions. A failed prediction can lead to direct consumption of polluted water by humans and animals, or indirect consumption of pollutants by humans or animals through plants irrigated by polluted groundwater. Only Min recall can guarantee the highest safety in predicting the pollution source each consecutive time. The case presented in Table 10 can be used as an example; it relates to the

| Table 10 | Example of why Min Recall is the strictest and safest metric to use for evaluation of a prediction; metrics and confusion matrix of classification case (Type A datasets): 1st Scenario, MLP-BF method-subset (see SM10, Sheets 1, 2) |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Metrics   | Classification—Simulation 1—Scenario 4 - MLP-BF                                                                                                                                         |
| Accuracy (%) | 93.329                                                                                                                                         |
| Class     | Precision  | Recall  | F1-score  | AUC       |
| 1         | 1.000       | 1.000   | 1.000     | 1.000     |
| 2         | 1.000       | 1.000   | 1.000     | 1.000     |
| 3         | 0.900       | 0.900   | 0.900     | 0.992     |
| 4         | 0.909       | 1.000   | 0.952     | 0.998     |
| 5         | 0.909       | 1.000   | 0.952     | 0.986     |
| 6         | 0.875       | 0.700   | 0.778     | 0.928     |
| w.avg     | 0.932       | 0.933   | 0.930     | 0.984     |
| min       | 0.875       | 0.700   | 0.778     | 0.928     |

| Table 11 Sim1 source prediction Min Recall of ML Classification methods (RF and MLP) for all subsets; values > 0.90 are highlighted as bold |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scenario  | Sim1 – ML - Min recall                                                                                                                                         |
|           | RF-FL | RF-BF | RF-GS | MLP-FL | MLP-BF | MLP-GS |
| 1         | 1.000 | 0.99   | 1.000 | 0.50   | 0.99   | 0.50   |
| 2         | 1.000 | 1.000  | 1.000 | 0.50   | 0.50   | 0.60   |
| 3         | 1.000 | 1.000  | 1.000 | 0.60   | 0.60   | 0.79   |
| 4         | 1.000 | 1.000  | 1.000 | 0.60   | 0.70   | 0.80   |
| 5         | 0.99  | 0.98   | 0.99  | 0.00   | 0.00   | 0.99   |
| Mean 1–4  | 1.00  | 1.00   | 1.000 | 0.55   | 0.70   | 0.67   |
| StDev 1–4 | 0.00  | 0.00   | 0.00  | 0.06   | 0.21   | 0.15   |
Classification case of Sim1 (type A datasets), Scenario 4, MLP-BF method-subset combination (see SM10).

The accuracy metric (here 93.329%) is obviously not indicative of the situation, as it is actually an average value summing up all true positives from all classes. Accuracy actually smooths and flattens variations in prediction performance, concealing the extend of failure in the worst performances. In a problem like the studied one, where minimum failure in prediction is acceptable, the worst performance is the one defining and characterizing the prediction performance of the process as a whole. So instead of weighted averages, min values of the other metrics of all classes should be investigated. In this example, as in most metrics’ results provided in supplementary material (SM10), the superiority of Min Recall metric is obvious. It is the most conservative metric, as it actually considers the class with the most actual failures; this corresponds to the source out of six (Scenarios 1–4) or

| Scenario | Subset | Nr MWs | Nr Features | IsMWPol | DayPol | DurPol |
|----------|--------|--------|-------------|---------|--------|--------|
| 1        | FL     | 585    | 1754        | 585     | 585    | 584    |
|          | BF     | 16     | 16          | 0       | 14     | 2      |
|          | GS     | 63     | 74          | 14      | 42     | 18     |
| 2        | FL     | 572    | 1713        | 572     | 572    | 569    |
|          | BF     | 4      | 4           | 0       | 4      | 0      |
|          | GS     | 67     | 77          | 26      | 22     | 29     |
| 3        | FL     | 620    | 1853        | 620     | 620    | 613    |
|          | BF     | 8      | 8           | 0       | 8      | 0      |
|          | GS     | 49     | 57          | 8       | 31     | 18     |
| 4        | FL     | 410    | 1223        | 403     | 410    | 410    |
|          | BF     | 19     | 20          | 2       | 17     | 1      |
|          | GS     | 61     | 65          | 11      | 44     | 10     |
| 5        | FL     | 494    | 1482        | 494     | 494    | 494    |
|          | BF     | 61     | 70          | 4       | 31     | 35     |
|          | GS     | 61     | 105         | 6       | 45     | 54     |
out of 24 (Scenario 5) that is worse predicted. In the presented example (Table 10), the worse class, as far as source prediction is concerned, is class 6 (cases where pollution source is Source 6; see Fig. 1). AUC value (0.928) is way too high and misleading, just like Accuracy metric was (93.329%). In class 6, Precision is rather more conservative, while $F_1$-score is even more. Min Recall, though, exhibits the lowest value as a prediction metric. Moreover, one should take into account that the proposed methodology/tool is meant for water resources' management scientists to use, and ultimately, to the policy/decision-makers and respective stakeholders to study its results. The Min Recall metric has the advantage to relate to a more realistic, instinctive, and easily-comprehended concept for all them. This is why, this paper evaluates the various methods and their performance with the Min Recall (of all classes) metric and proposes the future use of it by the scientific community in relevant problems.

Table 11 presents Min Recall for all ML methods/scenarios for Sim1. In Scenario 5, dataset package is differently sized and has six times more classes (24 instead of 6) and that is why its metrics do not participate in statistical calculations (mean, standard deviation for all classes). RF performance is excellent in all subsets, also exhibiting consistency. On the other hand, MLP performance is poor, with a few isolated exceptions. At this point, the first goal of solving the real-time/operational problem for a given monitoring network is achieved, using machine learning (RF). The methodology up to this point can accurately predict a pollution source (for example if the given network is the BF or GS or any other subset). Moreover, the second explicitly declared goal is partially fulfilled. Indirect optimization of the monitoring network occurs, though in the spatial dimension only. CFS methods BF an GS can minimize the Nr of MWs, but there is no control of the temporal dimension, which entails sampling in every timestep (daily in our case). The significance of the indirect spatial optimization alone is not to be underestimated; it practically resembles a sensor-based groundwater pollution monitoring network, since new sensors can provide real-time information continuously, in a frequency in the order of seconds even.

Table 13 Sim2 source prediction Min Recall of ML Classification methods (RF and MLP) for all subsets; values > 90 are highlighted as bold

| Scenario | RF-FL | RF-BF | RF-GS | MLP-FL | MLP-BF | MLP-GS |
|----------|-------|-------|-------|--------|--------|--------|
| 1        | 1.00  | 1.00  | 1.00  | 0.30   | 0.60   | 0.60   |
| 2        | 1.00  | 1.00  | 1.00  | 0.50   | 0.80   | 0.70   |
| 3        | 1.00  | 1.00  | 1.00  | 0.30   | 0.90   | 0.60   |
| 4        | 1.00  | 1.00  | 1.00  | 0.50   | 0.70   | 0.49   |
| 5        | 1.00  | 1.00  | 1.00  | 0.00   | 0.30   | 0.20   |
| Mean 1–4 | 1.00  | 1.00  | 1.00  | 0.40   | 0.75   | 0.60   |
| StDev 1–4| 0.00  | 0.00  | 0.00  | 0.11   | 0.13   | 0.08   |

Table 14 Sim3 features’ selected subsets (FL, BF, GS) per scenario: Nr of MWs selected; Nr of features (Nr of Type A datasets’ columns; see Table 8 or SM3); feature types (IsMWPoL, DayPoL, DurPoL)

| Scenario | Subset | Nr MWs | Nr Feat | IsMWPoL | DayPoL | DurPoL |
|----------|--------|--------|---------|---------|--------|--------|
| 1        | FL     | 585    | 1565    | 639     | 420    | 506    |
|          | BF     | 12     | 12      | 2       | 10     | 0      |
|          | GS     | 76     | 91      | 19      | 59     | 13     |
| 2        | FL     | 625    | 1875    | 625     | 625    | 625    |
|          | BF     | 59     | 68      | 19      | 38     | 11     |
|          | GS     | 57     | 66      | 14      | 41     | 11     |
| 3        | FL     | 639    | 1917    | 639     | 639    | 639    |
|          | BF     | 70     | 78      | 31      | 21     | 26     |
|          | GS     | 50     | 53      | 21      | 11     | 21     |
| 4        | FL     | 420    | 1252    | 412     | 420    | 420    |
|          | BF     | 6      | 6       | 0       | 6      | 0      |
|          | GS     | 59     | 63      | 14      | 37     | 12     |
| 5        | FL     | 506    | 1518    | 506     | 506    | 506    |
|          | BF     | 42     | 42      | 1       | 39     | 2      |
|          | GS     | 72     | 74      | 2       | 66     | 6      |
3.1.2 Classification Sim2 results

In Sim1, Algorithm 2’s Step 3 “Feature Selection” produces different feature subsets. Table 12 presents all relevant information for Sim2: Nr of MWs, total Nr of features, and the respective breakdown in exact feature types (IsMWPol; DayPol; DurPol). The specific layouts of the two respective MWs’ networks are adopted by the authors as two indirect proposals of spatially (sub)optimal monitoring network strategies (Fig. 12). The BF subsets again propose the least Nr of MWs and features, up to 17 times less than GS, except for Scenario, where BF and GS coincide. However, GS subsets offer a big decrease of MWs compared to the “full dataset,” represented by the FL subsets.

Table 13 presents Min Recall for all ML methods/scenarios for Sim2. RF performance is consistently flawless in all subsets. MLP performance is consistently poor to mediocre; Min Recall = 0.70 in some cases is interesting but not acceptable in this kind of problems. At this point, the first goal and partially the second goal of the paper are solved, using Machine Learning (RF), through a different hydraulic simulation tool than Sim1. Results are even better, although the limits of improvement were marginal, anyway. This was expected as the synthetic data produced by Sim2 were richer and more realistic.

3.1.3 Classification Sim3 results

In Sim3, Algorithm 2’s Step 3 “Feature Selection” produces different feature subsets, just like in Sim2. Table 14 presents Sim3 feature selection results (BF and GS subsets; see Table 14); these two layouts of the two respective MWs’ networks are adopted by the authors as two indirect proposals of spatially (sub)optimal monitoring network strategies.
presents all relevant information for Sim3: Nr of MWs, total Nr of features, and the respective breakdown in exact feature types (IsMWPol; DayPol; DurPol). The specific layouts of the two respective MWs’ networks are again adopted by the authors as two indirect proposals of spatially (sub)optimal monitoring network strategies (Fig. 13). The BF subsets do not always propose the least Nr of MWs and features, like Sim1 and 2. The ratio of GS/BF Nr of MWs’ values varies here from 10 to 0.7. Both methods, though, offer a big decrease of MWs compared to the “full dataset,” represented by the FL subsets.

Table 15 presents Min Recall for all ML methods/scenarios for Sim3. RF performance is again consistently flawless in all subsets, just like in Sim2 (the synthetic datasets’ production method is the same). MLP performance ranges from extremely poor to good; Min Recall = 0.80 in some cases is promising but not acceptable in this kind of problems. At this point, the first goal and partially the second goal of the paper are solved, using Machine Learning (RF), with the same hydraulic simulation tool for synthetic data production as Sim2, but for more complex and realistic problem version: variable contaminant release duration. The results are again flawless, despite the added complexity.

Table 16 Sim1 CNN min recall for 9 temporal discretizations and 3 spatial MW distributions (filtered monitoring network: CNN-FL, BF/GS-optimized network: CNN-BF/CNN-GS); values > 90 are highlighted as bold

| Scenario | Sampling frequency = 1 d | Subset | 10 d | 20 d | 30 d | 40 d | 50 d | 60 d | 70 d | 80 d |
|----------|-------------------------|--------|------|------|------|------|------|------|------|------|
| 1        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 0.68 | 0.20 | 0.47 |
|          | CNN-BF                  | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 0.68 | 0.20 | 0.50 |
| 2        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.82 | 0.98 |
|          | CNN-BF                  | 0.97   | 0.91 | 0.70 | 0.63 | 0.70 | 0.91 | 0.58 | 0.03 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.82 |
| 3        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.84 | 0.78 |
|          | CNN-BF                  | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 1.00 | 0.84 |
| 4        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.84 | 0.78 |
|          | CNN-BF                  | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 0.82 | 0.50 | 0.03 |
| 5        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.58 | 0.03 |
|          | CNN-BF                  | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 0.82 | 0.50 | 0.03 |

Table 17 Sim2 CNN min recall for 9 temporal discretizations and 3 spatial MW distributions (filtered monitoring network: CNN-FL, BF/GS-optimized network: CNN-BF/CNN-GS); values > 90 are highlighted as bold

| Scenario | Sampling frequency = 1 d | Subset | 10 d | 20 d | 30 d | 40 d | 50 d | 60 d | 70 d | 80 d |
|----------|-------------------------|--------|------|------|------|------|------|------|------|------|
| 1        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|          | CNN-BF                  | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 2        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|          | CNN-BF                  | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 3        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|          | CNN-BF                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 4        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|          | CNN-BF                  | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 5        | CNN-FL                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|          | CNN-BF                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
|          | CNN-GS                  | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
3.2 Computer vision results

3.2.1 Computer vision Sim1 results

In computer vision, there is no new feature selection involved. The classification Sim1 feature subsets/network layouts apply (Table 9; Fig. 11). Table 16 presents Min Recall for all CNN subsets-sampling frequencies combinations, for Sim1. CNN performs consistently excellent to flawlessly in the unrealistic “full dataset” FL subsets in Sim1, with Min Recall > 0.90, up to totally acceptable sampling frequencies that vary from 50 to 80 d, depending on the scenario. The FL subset of course, is unrealistic as it corresponds to the construction of (and sampling from) several hundreds of monitoring wells, practically in all not masked grid nodes of each scenario. CNN’s performance is disappointing for the BF subset in all scenarios, with trivial random, not generalizable exceptions (Scenario 2, 1d and 10d sampling intervals). On the other hand, in Sim1, CNN combined with the GS subset is absolutely successful in consistently predicting the source, for sampling intervals from 50 to 80 d for Scenarios 1–4, and for the more complex Scenario 5. At this point, the first goal of solving the real-time/operational problem for a given monitoring network is achieved, using deep learning (CNN). The proposed methodology can accurately predict a pollution source (for example if the given network is the BF or GS or any other subset) with CNN. Moreover, the second explicitly declared goal is totally fulfilled. Indirect optimization of the monitoring network occurs, in both the spatial dimension and temporal dimension. CFS methods minimize the Nr of MWs, while the trial-and-error tests of various sampling intervals lead to (sub)optimal monitoring strategy proposals.

3.2.2 Computer vision Sim2 results

The classification-produced Sim2 feature subsets also apply here (Table 11; Fig. 12). Table 17 presents Min Recall for all CNN subsets-sampling frequencies combinations, for Sim2. As expected, due to the more realistic synthetic data production method, and the grid-like form of the concentration results of Sim2, CNN’s performance is even better. Actually it is absolutely flawless in the FL and GS subsets, even for the max tested 80 d sampling interval. The CNN-BF combination is a roller-coaster; performance in Scenarios 1, 2, 4 is a complete failure, while in Scenarios 2 and 5, flawless! The goals of research are again succeeded in Sim2, in an emphatic way, using DL.

3.2.3 Computer vision Sim3 results

The classification-produced Sim3 feature subsets also apply here (Table 14; Fig. 13). Table 18 presents Min Recall for all CNN subsets-sampling frequencies combinations, for Sim3. CNN’s performance is excellent in the FL and GS subsets, even for 80 d. The same inconsistent behavior is observed in BF subset, on the other hand. The goals of research are also achieved in the complex Sim3, using DL.

3.3 Monitoring network cost minimization

For a better understanding of the practical use of the proposed methodology, Fig. 14 is created. It is actually a
colored table presenting an effort to identify and enumerate best GPMN strategies, produced by the proposed methodology for Scenario 1 for all simulations. Moreover, the strategies are organized based on their, simplistically yet realistically calculated, GPMN construction and operation depreciated annual costs. The solutions presented are the ones with the larger sampling intervals that exhibit Min Recall > 0.9. In supplementary material, SM11 provides the respective results for all scenarios. Figure 14 includes the ML/DL-CFS combination used, the Nr of MWs needed, the produced prediction performance (Min Recall), the respective sampling frequency in days, the overall annual GPMN cost for (a) a manual sampling solution, and (b) for a sensor-based GPMN solution for two different cost cases.

The two cases are used to prove the sensitivity of GPMN cost due to external non-scientific/hydraulic/computing factors. These can be political/financial circumstances, like the chip shortage and technological equipment’s (including sensors) market price increase, due to the ongoing COVID-19 pandemic [88, 89], or unrest/conflicts [90, 91]). The optimization’s sensitivity toward these external factors dictate even more the production and identification of many sub-optimal solutions that may constitute alternative strategies, perhaps relocated in the hierarchized list of optimal dedicated GPMN strategies, due to labor/equipment price variations. Case 1 assumes a sampling cost of SC = 10€/sample/yr and CC = 2000€ initial construction cost for a manual sampling strategy, and SC = 2€/sample/

| Sim Nr | Method - Subset | Nr of MWs | Min Recall | Sampl freq (d) | Manual sampling | Sensor-based sampling | Manual sampling | Sensor-based sampling |
|--------|----------------|-----------|------------|---------------|-----------------|----------------------|-----------------|----------------------|
| 1      | RF-FL          | 657       | 1.00       | 1             | 2,568,213       | 1,075,181            | 5,059,853       | 554,607              |
| 2      | RF-BF          | 6         | 0.99       | 1             | 23,454          | 9,819                | 46,209          | 5,065                |
| 3      | RF-GS          | 23        | 1.00       | 1             | 89,907          | 37,640               | 177,133         | 19,415               |
| 4      | MLP-BF         | 6         | 0.99       | 1             | 23,454          | 9,819                | 46,209          | 5,065                |
| 5      | CNN-FL         | 657       | 1.00       | 50            | 218,124         | 605,163              | 359,675         | 319,598              |
| 6      | CNN-GS         | 23        | 1.00       | 50            | 7,636           | 21,185               | 12,591          | 11,188               |
| 1      | RF-FL          | 585       | 1.00       | 1             | 2,286,765       | 957,353              | 4,505,348       | 493,828              |
| 2      | RF-BF          | 16        | 1.00       | 1             | 62,544          | 26,184               | 123,223         | 13,506               |
| 3      | RF-GS          | 63        | 1.00       | 1             | 246,267         | 103,100              | 485,191         | 53,181               |
| 4      | CNN-FL         | 585       | 1.00       | 80            | 178,206         | 535,641              | 288,230         | 282,972              |
| 5      | CNN-GS         | 63        | 1.00       | 80            | 19,191          | 57,684               | 31,040          | 30,474               |
| 1      | RF-FL          | 585       | 1.00       | 1             | 2,286,765       | 957,353              | 4,505,348       | 493,828              |
| 2      | RF-BF          | 12        | 1.00       | 1             | 46,908          | 19,638               | 92,417          | 10,130               |
| 3      | RF-GS          | 76        | 1.00       | 1             | 297,084         | 124,374              | 585,310         | 64,155               |
| 4      | CNN-FL         | 839       | 1.00       | 80            | 255,580         | 768,209              | 413,375         | 405,835              |
| 5      | CNN-BF         | 12        | 0.95       | 80            | 3,656           | 10,988               | 5,912           | 5,805                |
| 6      | CNN-GS         | 76        | 1.00       | 80            | 23,152          | 69,588               | 37,445          | 36,762               |

Fig. 14 Best GPMN solutions (Min Recall > 0.90) produced by the proposed methodology for Scenario 1 of Sim1-3; construction and operation annual costs for manual/sensor-based sampling, for 2 cases of cost parameters (see SM11 for all scenarios’ best strategies)
yr and CC = 7000€ for a sensor-based monitoring strategy. Case 2 assumes SC = 20€/sample/yr and CC = 3000€ for manual sampling, and SC = 1€/sample/yr and CC = 4000€ for sensor-monitoring. Initial construction costs are converted to the respective depreciated annual costs for a period of 10 years and an annual percentage flat rate of 5%. The equation providing estimates of the GPMN’s cost, practically assigned the role of an objective function to be minimized, is:

$$\text{GPMN\_Cost} = \text{SC} \cdot \text{NrMWs} \cdot \left( \frac{365}{\text{SF}} \right) + \text{DCC} \cdot \text{NrMWs}$$

where GPMN_cost is the total annual cost of groundwater pollution monitoring network’s depreciated construction and operation; SC is the sampling cost (per sample); NrMWs is the Nr of MWs; SF is the sampling frequency; DCC is the annual depreciated construction cost of each MW for an interest of 5% for 10 years.

In Case 1, the best solution of Sim1 providing Min Recall > 0.99 is Nr 6 (CNN-GS) for 7636€/yr including manual sampling in the 23 MWs of Sim1-GS subset (Fig. 11; red), every 50 days. This solution in a sensor-based approach, would cost 21,185€/yr (+ 13,549€/yr or + 177%) including daily automatic sampling in the same MWs. The respective Case 1 sensor-based best solutions (RF-BF, MLP-BF) cost 9819€/yr (+ 2183€/yr or + 29% compared to the best manual) including daily automatic sampling in the 6 MWs of Sim1-BF subset (Fig. 11; green). These solutions in manual sampling approach, would cost 23,454€/yr (+ 13,635€/yr or + 139% compared to their sample-based version) involving daily manual sampling in the same MWs. The external factors’ influence and the change of hierarchy concerning the less costly strategy can be seen in solution Sim1-Nr 6: while in the first case of construction/sampling costs, a manual sampling strategy is promoted, a certain change in their “current” Case1 values (Case 2) may invert things and behold: the sensor-based strategy is more cost-effective, given the objective function (total GPMN cost) assigned. Many such examples can be found in the other simulations and proposed solutions/strategies provided in Fig. 14 (and SM11).

4 Conclusions

Current research managed to implement machine/deep learning methods in order to solve the pollution source complex inverse groundwater pollution source identification problem, achieving the goals: (a) identify the source location among given potential locations with known contaminant release history, (b) achieve indirect optimization of the groundwater pollution monitoring network, with (1) min number and, (2) optimal locations of monitoring wells, (3) optimized sampling schedule (min frequency) through feature selection and trial-and-error tests. The methodology proposed, was analytically presented, step-by-step, including the data formulations needed for the various implementations (Machine Learning with Type A, and deep learning with Type B datasets). The proposed process included the Min Recall of all classes as the safest and most suitable metric for problems involving public health. The methodology was evaluated and proved robust, with an acceptable training speed, and excellent prediction accuracy for groundwater pollution monitoring network optimization.

As far as the classification-machine learning methods were concerned, multilayer perceptron proved to be not suitable for this kind of problem. Random forest, on the other hand, performed excellently in the (quite unrealistic due to the large costly monitoring network involved) full-data mode (FL subsets) in all scenarios, even Scenario 5 with 24 potential sources, providing consistently perfectly accurate predictions. With the proposed use of the correlation-based feature selection feature selection methods, best friend and greedy stepwise, combined with the proposed data formulation (Type A), indirect spatial optimization of the monitoring network was achieved. Multilayer perceptron while performing poorly in FL subsets, exhibited a better performance when implemented in the Best Friend and Greedy Stepwise subsets, but is not considered suitable for safe predictions and should be further dismissed. Random Forest, on the other hand, performed excellently in a consistent fashion, even on the demanding and complex Scenario 5. The first goal was reached: it was proven that in a given aquifer, for a given monitoring network, namely known monitoring wells’ locations (manual or sensor-based sampling every time-step; here daily), machine learning method random forest can robustly and instantly identify a pollution source from a number of potential ones when the instantaneous (duration = 1 timestep; here 1 day) leak start time is known (Sim1 and 2), and when start time is known but leak duration information is absent (Sim3). Moreover, the use of correlation-based feature selection methods, as pseudo-optimization tools can indirectly optimize the spatial dimension of the monitoring network. This is not to be considered as a half-measure: spatial optimization alone practically resembles a sensor-based groundwater pollution monitoring network; the implied daily sampling frequency is not prohibitive, as perhaps instinctively one would initially think.

A computer vision-deep learning method used, convolutional neural networks, also performed excellently in the full data mode in all scenarios and simulations. While it
exhibited inconsistent behavior when combined with best friend subsets, it performed flawlessly with the greedy stepwise subsets in every test, even for large sampling intervals. Thus, convolutional neural networks can be added to the arsenal of machine/deep learning methods that can solve the groundwater pollution source identification problem for a given aquifer monitoring network. Moreover, targeted trial-and-error tests of various sampling frequencies prove to be a robust method to further optimize the monitoring network in the temporal dimension. Convolutional neural networks performed flawlessly with the greedy stepwise subsets in all scenarios and simulations, even for high sampling frequencies, ranging from 40 to 80 d. Hence, the second goal was also achieved: exploiting the classification feature selection method in order to indirectly optimize the spatial dimension of a dedicated (to a specific aquifer) groundwater pollution monitoring network, and targeted trial-and-error sampling frequency tests we managed to further indirectly optimize the temporal dimension of the monitoring network. Finally, we provided a novel unconventional way of a dedicated to an aquifer monitoring network optimization when (a) contaminant release history is known in an advection-dominated confined aquifer-pollutant combination (Sim1); (b) contaminant release history is known in an aquifer-pollutant combination where advection and dispersion govern mass transport (Sim2); (c) contaminant release history is only partly known (given time of initiation of release but not duration) in an aquifer-pollutant combination where advection and dispersion govern mass transport (Sim3). The currently proposed unconventional optimization methodology simultaneously provides optimized ground-water pollution monitoring network strategies together with the trained and validated identification algorithm dedicated and tailored for the specific aquifer/network.

Practically, the fact that convolutional neural networks’ structure facilitates search of time-variant feature subsets, conducted here by trial-and-error tests, provides managing authorities with alternative monitoring strategies. For example, in Scenario 1-Case1-Sim1, the authorities are provided with at least two monitoring strategies: (a) one with the six monitoring wells of Fig. 11 and daily automatic sensor-based sampling (RF-BF), or (b) with the 23 monitoring wells of Fig. 11 and 50 d sampling (CNN-GS). Both strategies exhibit similar low budgets, and feature guaranteed prediction success and a dedicated/tailored-to-the-GPMN source prediction algorithm. Strategy “a” (sensors-based sampling and in situ analysis) can predict the source within seconds, 309 days after the leak incident. Strategy “b” (manual sampling and ex situ analysis) can predict the source within hours (depending on the pollutant type), 309 days following the leak.

The progress of this research with the proposed methodology paves the way for the next step in future research: assignment of a non-gradient metaheuristic optimization algorithm, like a genetic algorithm, to repetitively train/validate CNN controlling the masking of the spatial (testing random various subsets) and temporal dimensions (summing random various frames) to directly optimize the monitoring network/schedule, minimizing the respective cost function.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-022-07507-8.

Acknowledgements This research is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme “Human Resources Development, Education and Lifelong Learning 2014-20” in the context of the project “Evolution of Computational Intelligence in Environmental Engineering-Generalization, Improvement, Optimal Combination of Methodologies in Air Quality & Water Resources Problems” (MIS 5052163).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Kontos Y, Katsifarakis K (2012) Optimization of management of polluted fractured aquifers using genetic algorithms. Eur Water 40:31–42
2. Kontos Y, Katsifarakis K (2017) Optimal management of a theoretical coastal aquifer with combined pollution and salinization problems, using genetic algorithms. Energy 136:32–44. https://doi.org/10.1016/j.energy.2016.10.035
3. European Parliament and Council: Directive 2004/35/CE of the E.P. and E.C. of 21-4-2004 on environmental liability with regard to the prevention and remediaying of environmental damage, OJ L 143, 30.4.2004, pp 56–75 (current version 26/06/2019) (2004)
4. Tikhonov AN, Arsenin VY (1977) Solutions of Ill-posed problems. Wiley, New York
5. Atmadja J, Bagtzoglou AC (2001) State of the art report on mathematical methods for groundwater pollution source identification. Environ Forensics 2(3):205–214. https://doi.org/10.1006/enfo.2001.0055
6. Poeter EP, Hill MC (1997) Inverse models: a necessary next step in ground-water modeling. Groundwater 35(2):250–260. https://doi.org/10.1111/j.1745-6584.1997.tb00082.x
7. Mahinthakumar G, Sayeed M (2005) Hybrid genetic algorithm—local search methods for solving groundwater source identification inverse problems. J Water Resour Plan Manag 131(1):45–57. https://doi.org/10.1061/(ASCE)0733-9496(2005)131:1(45)
8. Holland JH (1975) Adaptation in natural and artificial systems. Univ of Mich Press, Ann Arbor
9. Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, p 412
10. Karpozous DK, Delay F, Katsifarakis KL, De Marsily G (2001) A multipopulation genetic algorithm to solve the inverse problem in hydrogeology. Water Resour Res 37(9):2291–2302. https://doi.org/10.1029/2000WR900411
11. Giacobbo F, Marseguerra M, Zio E (2002) Solving the inverse problem of parameter estimation by genetic algorithms: the case of a groundwater contaminant transport model. Ann Nucl Energy 29(8):967–981. https://doi.org/10.1016/S0306-4549(01)00084-6

12. Cannon JR (1966) Some numerical results for the solution of the heat equation backwards in time. In: Proceedings of advanced symposium numerical solution of nonlinear differential equation. Wiley, New York, pp 21–54

13. Lattes R, Lions JL (1969) The method of quasi-reversibility, applications to partial differential equations. Elsevier, New York. https://doi.org/10.1006/jmaa.1999.6462

14. Neupauer RM, Borchers B, Wilson JL (2000) Comparison of inverse methods for reconstructing the release history of a groundwater contamination source. Water Resour Res 36(9):2469–2475. https://doi.org/10.1029/2000WR900176

15. Carasso AS (1992) Space marching difference schemes in the inverse problem of the heat equation backwards in time. In: Proceedings of advanced symposium numerical solution of nonlinear differential equation. Wiley, New York, pp 21–54

16. Ewing RE (1975) The approximation of certain parabolic equations backward in time by Sobolev equations. SIAM J Math Anal 6(2):283–294. https://doi.org/10.1137/0506029

17. Marquardt W, Auraner H (1990) An observer-based solution of inverse heat conduction problems. Int J Heat Mass Transf 33(7):1545–1562. https://doi.org/10.1016/0017-9310(90)90050-5

18. Woodbury A, Sudicky E, Ulrych TJ, Ludwig R (1998) Three-dimensional plume source reconstruction using minimum relative entropy inversion. J Contam Hydro 32:131–158. https://doi.org/10.1016/S0269-7722(97)00088-0

19. Birchwood RA (1999) Identifying the location and release characteristics of a groundwater pollution source using spectral analysis. In: Proceedings of 19th annual American geophysical union hydrology days conference, Fort Collins, Colorado, Colorado State University, pp 37–50

20. Buzbee BL, Carasso A (1973) On the numerical computation of parabolic problems for preceding times. Math Comput 27(122):237–266. https://doi.org/10.1090/S0025-5718-1973-0368448-3

21. Atmadja J, Bagtzoglou AC (2001) Pollution source identification in heterogeneous porous media. Water Resour Res 37:2113–2125. https://doi.org/10.1029/2001WR000223

22. Silva Neto AJ, Ozisik MN (1993) Inverse problem of simultaneity for pollution source location and travel time probabilities for a multi-dimensional groundwater system. Water Resour Res 37(6):1657–1668. https://doi.org/10.1029/2000WR900388

23. Snodgrass MF, Kitaniidis PK (1997) A geostatistical approach to contaminant source identification. Water Resour Res 33(4):537–546. https://doi.org/10.1029/96WR03753

24. Sidaourk P, Cheng A, Ouazar D (1998) Ground water contaminant source and transport parameter identification by correlation coefficient optimization. Ground Water 36(2):208–214. https://doi.org/10.1111/j.1745-6584.1998.tb01085.x

25. Alapati S, Kabala ZJ (2000) Recovering the release history of a groundwater contaminant via the non-linear least-squares estimation. Hydrology Process 14(6):1003–1016. https://doi.org/10.1002/(SICI)1099-1085(20000430)14:6<1003::AID-HYP871>3.0.CO;2-W

26. Mo S, Zabaras N, Shi X, Wu J (2019) Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification. Water Resour Res 55(5):3856–3861. https://doi.org/10.1029/2018WR024638

27. Chadalavada S, Datta B, Naidu R (2011) Optimisation approach for pollution source identification in groundwater: an overview. Int J Environ Waste Manag 8(1–2):40–61. https://doi.org/10.1004/JEWM.2011.040964

28. Aral MM, Guan J (1996) Genetic algorithms in search of groundwater pollution sources. In: Aral MM (ed) Advances in groundwater pollution control and remediation. Springer, Dordrecht, pp 347–369. https://doi.org/10.1007/978-94-009-0205-3_17

29. Aral MM, Guan J, Maslia ML (2001) Identification of contaminant source location and release history in aquifers. J Hydrol 225:225–234. https://doi.org/10.1016/S0168-1694(01)00225

30. Han K, Rui Z, Pengcheng N et al (2020) Application of a genetic algorithm to groundwater pollution source identification. J Hydrol 589:125343. https://doi.org/10.1016/j.jhydrol.2020.125343

31. Li J, Lu W, Fan Y (2021) Groundwater pollution sources identification based on hybrid homotopy-genetic algorithm and simulation optimization. Environ Eng Sci. https://doi.org/10.1089/ees.2020.0117

32. Bharat TV, Sivapullaiah PV, Allam MM (2009) Swarm intelligence based inverse model for characterization of groundwater contaminant source. Electron J Geotech Eng 14(B):1–4

33. Yeh HD, Chang TH, Lin YC (2007) Groundwater contaminant source identification by a hybrid heuristic approach. Water Resour Res. https://doi.org/10.1029/2005WR004731

34. Mirghani BY, Mahinthakumar KG, Tryby ME, Ranjithan RS, Zechman EM (2009) A parallel evolutionary strategy based simulation–optimization approach for solving groundwater source identification problems. Adv Water Resour 32(9):1373–1385. https://doi.org/10.1016/j.advwatres.2009.06.001

35. Sun M, You XY (2019) Method of traceability of pollution source by machine learning. In: InIOP conference series: earth and environmental science. IOP Publishing. https://doi.org/10.1088/1755-1315/310/5/052082

36. Shen C (2018) A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resour Res 54(11):8558–8593. https://doi.org/10.1029/2018WR022643

37. Rajabi MM, Ataie-Ashtiani B, Simmons CT (2018) Model-data fusion techniques for deep learning and environmental science. IOP Publishing. https://doi.org/10.1088/1755-1315/310/5/052082

38. Sit M, Demiray BZ, Xiang Z, Ewing GJ, Sermet Y, Demir I (2019) Application of particle methods to reliable identification of groundwater sources. In: Aral MM (ed) Advances in groundwater pollution control and remediation. Springer, Dordrecht, pp 347–369. https://doi.org/10.1007/978-94-009-0205-3_17

39. Han K, Rui Z, Pengcheng N et al (2020) Application of a genetic algorithm to groundwater pollution source identification. J Hydrol 589:125343. https://doi.org/10.1016/j.jhydrol.2020.125343

40. Li J, Lu W, Fan Y (2021) Groundwater pollution sources identification based on hybrid homotopy-genetic algorithm and simulation optimization. Environ Eng Sci. https://doi.org/10.1089/ees.2020.0117
45. Singh RM, Datta B, Jain A (2004) Identification of unknown groundwater pollution sources using artificial neural networks. J Water Resour Plan Manag 130(6):506–514. https://doi.org/10.1061/(ASCE)0733-9496(2004)130:6(506)

46. Singh RM, Datta B (2007) Artificial neural network modeling for identification of unknown pollution sources in groundwater with partially missing concentration observation data. Water Resour Manag 21(3):557–572. https://doi.org/10.1007/s11269-006-9029-z

47. Ayaz M, Srivastava R, Jain A (2014) Groundwater pollution source identification using linked ANN-optimization model. In: InEGU General Assem Abstr

48. Zhang J, Zheng Q, Chen D, Wu L, Zeng L (2020) Surrogate-based Bayesian inverse modeling of the hydrological system: an adaptive approach considering surrogate approximation error. Water Resour Res 56:1. https://doi.org/10.1029/2019WR025721

49. Loaiciga HA, Charbeneau RJ, Everett LG, Fogg GE, Hobbs BF, Rouhani S (1992) Review of ground-water quality monitoring network design. J Hydraulic Eng 118(1):11–37. https://doi.org/10.1061/(ASCE)0733-9429(1992)118:1(1)

50. Minsker B (2003) Long-term groundwater monitoring—the state of the art. Am Soc Civil Eng

51. US EPA (2005) Roadmap to long-term monitoring optimization

52. Kollat JB, Reed PM, Maxwell RM (2011) Many-objective groundwater monitoring network design using bias-aware ensemble Kalman filtering, evolutionary optimization, and visual analytics. Water Resour Res 47(2):1–18. https://doi.org/10.1029/2010WR009194

53. Yesilpinar NB, Elfeky AM, Gehrels JC, van den Akker C, Hensbergen AT, Dekking FM (2005) Reliability assessment of groundwater monitoring networks at landfill sites. J Hydrol 308(1):1–17. https://doi.org/10.1016/j.jhydrol.2004.10.017

54. Meyer PD, Valocchi AJ, Eheart JW (1994) Monitoring network design to provide initial detection of groundwater contamination. Water Resour Res 30(9):2647–2659. https://doi.org/10.1029/94WR00872

55. Dhar A, Datta B (2007) Multiobjective design of dynamic monitoring networks for detection of groundwater pollution. Water Resour Plan Manag 133(4):329–338. https://doi.org/10.1061/(ASCE)0733-9496(2007)133:4(329)

56. Nunes LM, Paralta E, Cunha MC, Ribeiro L (2004) Groundwater nitrate monitoring network optimization with missing data. Water Resour Res. https://doi.org/10.1029/2003WR002469

57. Kollat JB, Reed P (2007) A framework for visually interactive decision-making and design using evolutionary multi-objective optimization (VIDEO). Environ Model Softw 22(12):1691–1704. https://doi.org/10.1016/j.envsoft.2007.02.001

58. Reed P, Minsker BS (2004) Striking the balance: long-term groundwater monitoring design for conflicting objectives. Water Resour Plan Manag 130(2):140–149. https://doi.org/10.1061/(ASCE)0733-9496(2004)130:2(140)

59. Zhang Y, Pinder GF, Herrera GS (2005) Least cost design of groundwater quality monitoring networks. Water Resour Res. https://doi.org/10.1029/2005WR009396

60. Reed P, Minsker B, Valocchi AJ (2000) Cost-effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation. Water Resour Res 36(12):3731–3741. https://doi.org/10.1029/2000WR900232

61. Wu J, Zheng C, Chien CC, Zheng L (2006) A comparative study of Monte Carlo simple genetic algorithm and noisy genetic algorithm for cost-effective sampling network design under uncertainty. Adv Water Res 29(6):899–911. https://doi.org/10.1016/j.adwres.2005.08.005

62. Wu J, Zheng C, Chien CC (2005) Cost-effective sampling network design for contaminant plume monitoring under general hydrogeological conditions. J Contam Hydrol 77(1–2):41–65. https://doi.org/10.1016/j.jconhyd.2004.11.006

63. Montas HJ, Mohtar RH, Hassan AE, Al Khal FA (2000) Heuristic space-time design of monitoring wells for contaminant plume characterization in stochastic flow fields. J Contam Hydrol 43(3–4):271–301. https://doi.org/10.1016/S0169-7722(99)00108-4

64. Hudak PF, Loaiciga HA, Marino MA (1995) Regional-scale ground water quality monitoring via integer programming. J Hydrol 164(1–4):153–170. https://doi.org/10.1016/0022-1694(94)02559-T

65. Mugunthan P, Shoemaker CA (2004) Time varying optimization for monitoring multiple contaminants under uncertain hydrogeology. Bioremediation J 8(3–4):129–146. https://doi.org/10.1080/1061/(ASCE)0733-9429(1992)118:1(11)

66. Li Y, Hilton ABC (2007) Optimal groundwater monitoring design using an ant colony optimization paradigm. Environ Model Softw 22(1):110–116. https://doi.org/10.1016/j.envsoft.2006.05.023

67. Bagkis E, Kassandros T, Karteris M, Karteris A, Karatzas K (2021) Analyzing and improving the performance of a particulate matter low cost air quality monitoring device. Atmosphere (2021) Analyzing and improving the performance of a particulate matter low cost air quality monitoring device. Atmosphere 12(2):251. https://doi.org/10.3390/atmos12020251

68. Langevin CD, Hughes JD, Banta ER, Provost AM, Niswonger RG, Panday S (2021) MODFLOW 6 modular hydrological model version 6.2.2. U.S. Geological Survey Software Release. https://doi.org/10.5066/F76Q1VQV

69. Karampasis M (2021) Pollution source identification and monitoring network optimization in aquifers with machine learning and Modflow (in Greek). Diploma Thesis. Aristotle University of Thessaloniki, Greece. https://doi.org/10.13140/RG.2.2.19392.87043

70. Anderson M, Woessner W, Hunt R (2015) Chapter 8-particle tracking. In: Anderson M, Woessner W, Hunt R (eds) Applied groundwater modeling, 2nd edn. Academic Press, San Diego, pp 331–373

71. Shimrat M (1962) Algorithm 112: position of point relative to polygon. Commun ACM 5(8):434. https://doi.org/10.1145/368637.368653

72. Python Package for Creating, Running, and Post-Processing MODFLOW-Based Models. USGS Flopy. https://www.usgs.gov/software/flopy-python-package-creating-running-and-post-processing-moflow-based-models Accessed 10 Nov 2021

73. Langevin CD, Hughes JD, Banta ER, Provost AM, Niswonger RG, Zuber S (2021) MODFLOW 6 modular hydrological model version 6.2.2. U.S. Geol Surv https://doi.org/10.5066/F76Q1VQV

74. Bakker M, Post V, Langevin CD, Hughes JD, White JT, Fienen MN (2016) FloPy: Python package for creating, running, and post-processing MODFLOW-based models. U.S. Geol Surv

75. Bakker M, Post V, Langevin CD, Hughes JD, White JT, Fienen MN (2016) Scripting MODFLOW model development and post-processing MODFLOW-based models. U.S. Geol Surv

76. Bakker M, Post V, Langevin CD, Hughes JD, White JT, Starn JJ, Fienen MN (2016) Flopy. Python package for creating, running, and post-processing MODFLOW-based models. U.S. Geol Surv

77. USGS (2021) MODFLOW 6 Program documentation. USGS. http://www.usgs.gov/software/flopy-python-package-creating-running-and-post-processing-moflow-based-models Accessed 10 Nov 2021

78. USGS (2021) Flopy Documentation 3.3.4. https://doi.org/10.5066/F76Q1VQV

79. USGS (2021) General Head Boundaries—Modflow 6 Program documentation https://modflow6.readthedocs.io/en/latest/_mf6io/gwf-ghb.html. Accessed 10 Nov 2021

80. Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324
80. McCulloch W, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133. https://doi.org/10.1007/BF02478259
81. Hall M (2000) Correlation-based feature selection for machine learning. PhD Thesis, Univ. of Waikato Hamilton
82. LeCun Y et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551. https://doi.org/10.1162/neco.1989.1.4.541
83. Ronneberger O, Fischer P, Brox T (2015). U-Net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A (eds) Medical image computing and computer-assisted intervention—MICCAI 2015. MICCAI 2015. Lecture notes in computer science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28
84. Frank E, Hall M, Witten I (2016) The WEKA workbench. Morgan Kaufmann, 4th edn. Springer, Boston. https://doi.org/10.1007/978-0-387-09823-4_66
85. Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J Roy Stat Soc Ser B (Methodological). 36(2):111
86. Bisong E (2019) Building machine and deep learning models on google cloud platform: a comprehensive guide for beginners. Apress, Berkeley, pp 59–64. https://doi.org/10.1007/978-1-4842-4470-8
87. Abadi M et al (2015) TensorFlow: large-scale machine learning on heterogeneous systems software. tensorflw.org. https://doi.org/10.5281/zenodo.4758419
88. Casper H, Rexford A, Riegel D, Robinson A, Martin E, Awwad M (2021). The impact of the computer chip supply shortage. In: Proceedings of the international conference on industrial engineering and operations management, Bangalore, India, August 16–18, 2021. IOEM Society International, pp 236–245
89. Tian X, Yang L, Huang Y, Fang Y (2022) Research on supply chain shortage of electronic technology companies under the pandemic situation: a case study of Apple Inc. and Xiaomi Inc. In: Proceedings of IC4E 2022: 2022 13th international conference on E-Education, E-Business, E-management, and E-Learning (IC4E), January, 14–17, 2022, Tokyo, Japan. Association for Computing Machinery, New York, NY, pp 605–610 https://doi.org/10.1145/3514262.3514330
90. Ruiz Estrada MA (2022) Who can damage severely to the world economy? COVID-19 vs. Russian-Ukrainian War. https://doi.org/10.2139/ssrn.4063845
91. Mbah RE, Wasum DF (2022) Russian-Ukraine War: a review of the economic impact of Russian-Ukraine crisis on the USA, UK, Canada, and Europe. Adv Soc Sci Res J 9(3):144–153. https://doi.org/10.14738/assrj.93.12005

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.