Computers and Chemical Engineering 203 (2025) 109320

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

A hybrid approach for steady-state production optimization of a real oil and gas platform: Integrating physics-based models, machine learning techniques, and field monitoring signals

Luca Trevisan [a], Ahmed Shokry [b,*], Marco Montini [c], Eric Moulines [b], Enrico Zio [a,d]

[a] Energy Department, Politecnico di Milano, Via Lambruschini 4, 20156 Milan, Italy
[b] Center for Applied Mathematics, Ecole Polytechnique, Route de Saclay, 91120 Palaiseau, France
[c] Eni SpA Upstream and Technical Services, Italy
[d] MINES-Paris, PSL University, Sophia Antipolis, France

ARTICLE INFO

Keywords:
Oil and gas
Subsea networks
Real Time production optimization
Surrogate based optimization
Hybrid models
Machine learning
Artificial neural network
Random forest regression

This work concerns the optimization of the daily production of a real offshore oil and gas network. The lack of accurate physics-based models for fundamental units in the system, particularly the multiphase pumps, makes it unfeasible to obtain an accurate physics-based model of the entire network to be used for optimization purposes. Therefore, this work innovatively addresses this issue by developing a hybrid model for the real oil and gas network under study. Extensive monitoring data obtained from the real multiphase pumps over an extended timeframe form the foundation for developing machine learning (ML) models, which are adept at capturing the complex relationships within the pump system. Subsequently, we integrate these ML-based pump models with the existing physics-based models of other network components, including wells, gathering networks, risers and separators. These physics-based models are crafted using state-of-the-art industrial software, ensuring robustness and accuracy in representing the components' actual behavior. The hybrid model's predictive capabilities are validated against real data from the offshore network, affirming its ability to accurately capture system behavior. Leveraging this validated hybrid model, optimization is performed using a differential evolution algorithm, for maximizing production efficiency while adhering to operational constraints. Our outcomes underscore several key findings: firstly, the ML-based pump models demonstrate remarkable accuracy in approximating the intricate relationships among pump variables, secondly, the hybrid model exhibits commendable predictive accuracy, effectively simulating the real behavior of the whole offshore production network; finally, the optimization yields tangible production enhancements, surpassing the network's actual performance under historical operating conditions.

1. Introduction

Offshore oil and gas production facilities are complex infrastructures that involve different heterogeneous systems with rather different dynamics, with each system composed of thousands of interconnected components. Such infrastructures include the reservoir, the wells, the gathering network (pipes, pumps, and manifolds), and the processing plant (separators, reactors, etc.). Given the complex and diverse nature of these systems and components, and the heterogeneous technologies utilized for their functionality, their optimal management and operation cannot be accomplished in a single-stage approach. On the contrary, they are managed and operated via hierarchical multi-layers structures, where each layer is concerned with decisions that take place at a different local and time scales (see Fig. 1). In this management hierarchy, each layer provides its decisions as setpoints or targets to the layer below, while the latter provides feedback on progress or any issues encountered in meeting the targets.

Within this hierarchical framework, the asset management layer focuses on strategic and long-term decisions, such as the investment strategy, the infrastructure configuration and the global operational models (Hülse, et al., 2020). Subsequently, the planning/scheduling layer(s) is tasked with setting the optimal production targets for mid-term horizons (weeks/days), ensuring alignment with external demand and basic production capacities, while also considering economic objectives. The steady-state Real Time Optimization (RTO) layer, which is the central focus of this work, handles short-term decisions (within

* Corresponding author.
E-mail address: ahmed.shokry@polytechnique.edu (A. Shokry).

https://doi.org/10.1016/j.compchemeng.2025.109320
Received 13 October 2024; Received in revised form 30 June 2025; Accepted 30 July 2025
Available online 5 August 2025

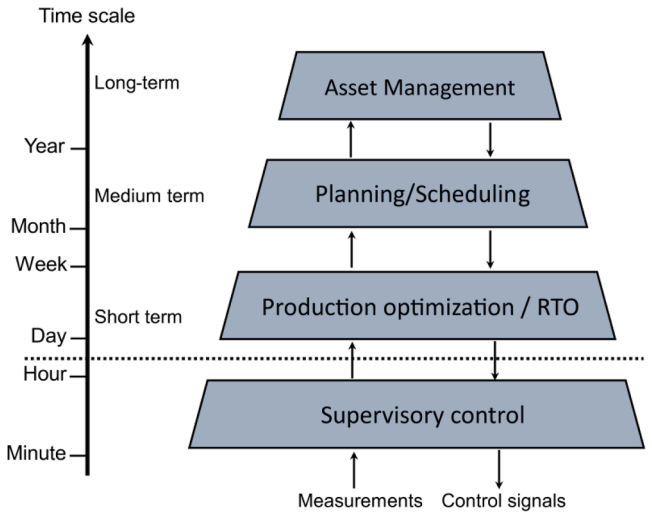


Fig. 1. Decision making hierarchy in the chemical and petrochemical industry (Hülse, et al., 2020).

hours/days) (Marchetti et al., 2014). These decisions involve determining optimal values for key variables, such as pressure, temperature, valve openings, and pump speeds, at which the plant and its units must operate to optimize specific performance indices like operating cost, profit and/or production yield, while adhering to safety, quality, environmental, and capacity constraints (Matias, et al., 2022). Achieving this goal entails solving a real-time optimization problem based on detailed, steady-state, physics-based model of the production facility. Depending on various factors such as model structure, transparency (e.g., white, grey, black-box), availability of derivative information, and formulation of objectives and constraints, different optimization algorithms can be employed. These include derivative-free algorithms (e.g., Genetic Algorithms), which utilize the explicit values of the objective functions to guide the optimization search, and derivative-based algorithms (e.g., interior point algorithms), which rely on derivatives of the objective functions with respect to decision variables (Ray and Sarker, 2007).

The steady-state RTO problem is typically solved at pre-scheduled time periods, often in the order of hours, allowing for adjustments to optimal decisions in response to various sources of variability that commonly affect operations in oil and gas networks (Fadda, 2017; Selvan, et al., 2022). These sources include events such as well start-ups, shutdown choke changes, unscheduled maintenance and equipment failures, among others. To minimize the mismatch between the plant and its model, reconciled estimates of measured steady-state data for plant variables are utilized to initialize the physics-based model and update its parameters (Biegler, 2010). Finally, the control layer is responsible for maintaining the plant at optimal configurations (set-points) determined by the RTO layer. This involves taking corrective action within minutes or seconds to address potential process disturbances and fluctuations.

A significant challenge for RTO in oil production facilities lies in developing an accurate and reliable physics-based model for such complex, large-scale, and heterogeneous infrastructures (Hülse, et al., 2020; Fetanat, 2024). The multitude of distinct processes and systems involved, along with their interrelationships, contributes to this difficulty (Wan, et al., 2023). For instance, the reservoir constitutes a geological system with relatively slow dynamics, involving geophysical and geomechanical phenomena. On the other hand, the gathering network represents a multiphase flow (MPF) system with very rapid dynamics. Lastly, the processing plant (comprising separators, exchangers, etc.) is a subsystem that involves chemical reaction kinetics and thermodynamic phenomena (Scaramellini, et al., 2015). Therefore, creating a single model for these distinct, large-scale and heterogeneous systems necessitates integrating various physical phenomena and dynamics, while balancing conflicts and interactions between variables, constraints and operational limitations (Giorgio, et al., 2014)

Numerous studies in the literature have delved into Real-Time Optimization (RTO) of oil and gas production facilities. In their work, Stanko and Golan (2015) focused on the steady-state RTO (SS-RTO) of three distinct onshore production networks. Their primary objective was to maximize the total oil production of the network by adjusting the frequency of Electrical Submersible Pumps (ESPs) installed on the wells. The work encompassed two synthetic case studies, each comprising two and eleven wells, respectively, alongside a real-world case involving 51 wells. Throughout these applications, the Pipesim software (SLB, 2024) was utilized for developing the network model, and various optimization algorithms, such as IPOPT and Mesh adaptive direct search nonlinear solvers, were compared. Giorgio et al. (2012) utilized commercial software to construct a black-box, high-fidelity, physics-based model of an actual onshore network comprising seven wells, a gathering network, a pressure-boosting pump and a crude treatment plant. They employed a customized evolutionary optimization algorithm named “e-Rabbit” (Risked Algorithms for Biogenetical Balance Integration Tool) to maximize production by adjusting the choke valve openings of the wells and the rotating speed of the pump. A similar workflow (involving the use of commercial software for network modeling and e-Rabbit for optimization) was employed by Brioschi et al. (2017) to maximize the production of a real oil and gas offshore platform comprising 11 wells distributed across two independent subnetworks. Additionally, the platform includes a Floating Production, Storage, and Offloading (FPSO) vessel equipped with a treatment plant consisting of a train of oil separators, alongside gas dehydration and compression units. Krishnamoorthy et al. (2018) developed a hybrid RTO (HRTO) method applied to a simple synthetic case study involving two wells, a riser and a separator. The HRTO uses transient data to update the parameters of a dynamic model of the network by means of an extended Kalman filter. Subsequently, optimization is carried out based on a static version of the model with the updated parameters. HRTO achieves a good tradeoff: on one side, it solves the limitation of SS-RTO represented by the steady-state waiting time or delay, while on the other side, it avoids the high computational cost of classical dynamic RTO (DRTO) where dynamic models are used for both parameter updating and online optimization. With a similar concept, Matias et al. (2022) developed Real-time Optimization with Persistent Adaptation (ROPA) for oil and gas networks, which was validated using a simple experimental test rig involving a water tank and piping system.

Since a physics-based model of an oil and gas network can often be overly complex (e.g., highly nonlinear), making it impractical for use in SS-RTO, several studies have attempted to simplify it to obtain a reduced-order mathematical model with a manageable computational burden (Hoffmann and Stanko, 2017). In this regard, Codas and Camponogara (2012), Codas, et al. (2012) and, Silva and Camponogara (2014) proposed the approximation of complex and nonlinear oil network models, such as well deliverability models, vertical lift performance relationships, and the flow pressure behavior of gathering and surface systems, using piecewise linear functions. This piecewise linearization approach enables the transformation of the RTO problem from a complex nonlinear mixed-integer programming (MINLP) formulation to a simpler Mixed-Integer Linear Programming (MILP) problem that can be solved within a reasonable computational time frame. Their approach has been applied to real-world cases, such as the Urucu field, which involves 28 wells and 9 separators, where decision variables included the bottom hole pressure and routing of the wells. The aforementioned approach was further extended by Hülse and Camponogara (2017) to incorporate a robust formulation, aiming to address uncertainties impacting the operation of the network. Epelle and Gerogiorgis (2019) introduced a MINLP formulation, where the complex model of the network is replaced by simple polynomial relations. Their

approach was applied to a synthetic case consisting of six wells, two manifolds, two pipelines and two separators, with the objective being the Net Present Value and the decision variables including optimal well controls, lift gas allocation and routing strategy; the Open-source Nonlinear Mixed Integer Programming (BONMIN) solver was utilized. Also, Carpio et al. (2021) compared the performances of MILP and MINLP formulations for production optimization, with a focus on a real case study. Camponogara, et al. (2018) and Luguesi et al., 2023 developed a method for optimizing the operation of a subsea gas network that collects production from various offshore platforms and delivers it to an onshore terminal. The high-fidelity simulator of the network is replaced by a set of polynomial regression models identified using simulation data. Subsequently, based on the simplified model, optimization is conducted using the orthogonal Mesh Adaptive Direct Search optimization algorithm.

The accuracy of the aforementioned mathematical simplification methods, based on piecewise linear or polynomial functions, may be compromised as the nonlinearity or complexity of the high-fidelity model increases. Additionally, a limitation frequently encountered in works employing this approach is that the prediction accuracy of the linearized or simplified mathematical model is not demonstrated with respect to the original high-fidelity model. Moreover, the effectiveness of the original model itself in accurately reproducing the behavior of the real network is often not demonstrated.

Another approach involves the use of machine learning models as accurate and computationally efficient surrogates to replace, wholly or partially, high-fidelity physics-based models of the network in optimization problems (Koroteev and Tekic, 2021; Bishnu, et al., 2023). These machine learning-based models are constructed using data generated by simulating the physics-based model itself. In this vein, Gongbo et al. (2023) developed a hybrid model for a simple synthetic production network, combining ANNs as a pressure drop model with analytical models for Inflow Performance Relationships (IPR) and separators. An optimization algorithm based on flower pollination and sea horse was utilized to determine optimal decisions, leveraging this hybrid network model. Mohammadzaheri et al. (2016) developed a method for optimizing the operation of electrical submersible pumps (ESPs) installed on subsea wells, which was applied to a simple synthetic case. They employed a pair of nested ANNs combined with affinity laws to predict the pressure head and power of each ESP as functions of the pump's rotational speed, inlet pressure, and flow rate. An evolutionary algorithm was used to seek the optimal solution by manipulating the flow rate and rotational speed of each ESP. Cadei et al. (2020) developed a method for optimizing a real production network, which includes 27 producing wells, a complex gathering network and a substantial processing plant consisting of several treatment trains. Two ANN models were employed to approximate the behavior of high-fidelity simulators: one ANN to emulate the fluid dynamics behavior in wells and gathering network, and another ANN to mimic the chemical stoichiometry and thermodynamics for the treatment plant. The aforementioned "e-Rabbit" optimization tool was utilized for optimizing production based on the ANN models, although no details have been disclosed regarding this phase. Andreasen (2020) employed Gaussian process (GP) models to approximate a complex simulation model, based on Aspen HYSYS (Aspentech, 2024), of an oil and gas separation plant. Subsequently, an evolutionary algorithm was utilized to optimize the operating profit based on the GP model. In a similar vein, Mendoza et al. (2021) utilized an ANN model to mimic the behavior of an oil and gas separation model based on ASPEN HYSYS, which was then used as a surrogate in a multi-objective optimization formulation employing a non-dominating sorting genetic algorithm. Grimstad, et al. (2016) Used multivariate splines to approximate very complex oil and gas network models developed by GAP software from Petroleum Experts. Then the production optimization problem is solved based on the splines through MINLP formulation using Convex ENvelopes for Spline algorithm based on branch and bound techniques. They applied their method to three different benchmark oil and gas production platforms. Yin et al. (2022) addressed the control of a natural gas network, utilizing ANN models to simplify a complex network model based on partial differential equations. These ANN models map the relationship among the network's flow rates, pressure, and control valve openings. Using the ANN model, an open-loop optimal control problem is solved using genetic algorithms (GA) to identify the optimal control scheme, which is then forwarded to a PID controller for implementation. Grimstad et al. (2016) used multivariate splines to approximate complex oil and gas network models developed by GAP software from Petroleum Experts (IPM-Suite, 2024). Then the production optimization problem was solved based on the splines through a MINLP formulation using the Convex Envelopes for Spline algorithm based on branch and bound technique; they successfully applied their method to three different benchmark oil and gas production platforms.

Al Selaiti et al. (2020) developed a data-driven approach to find optimal operating envelope/conditions for gas-lift wells. They constructed ML models using sensor data to map the relationship between well responses (multiphase flow rates, Water Cut (WC), Gas-Oil Ratio (GOR) and reservoir pressure) and control variables such as choke settings and gas injection rates. A sensitivity analysis was performed based on these ML models to identify optimal operating conditions. Al Lawati et al. (2021) developed data-driven methods based on a combination of similarity measures and unsupervised ML techniques to cluster wells with similar production and behavioral profiles. These well clusters were then used to identify the process path most likely to result in optimal production.

The majority of the works presented in the area of SS-RTO of oil and gas production platforms share at least one of the following limitations:

a) The ML techniques have been used only for the goal of simplifying a complex, physics-based simulation model to accelerate the optimization convergence. In other words, ML models are built using data generated by simulation with a physics-based model (assuming that such simulation model represents the real system behavior).
b) The considered physics-based models are relatively simple, i.e. built on simplified assumptions, and are commonly employed to represent small-scale production systems. Moreover, their accuracy is not validated against the actual behavior of the real- system.
c) The relevance of the numerically obtained steady-state optimization results has not been validated with respect to the real field performance.

This paper focuses on the solution of the unique challenges that face the steady-state production optimization of a real-world offshore oil and gas production network located in a western central African country. The network consists of eight wells belonging to two fields, where each well is equipped with a choke valve to control the flow. The MPF, outlet of the different wells, is collected and combined through the gathering system, and then the total flow is delivered to a train of two parallel MultiPhase Pumps (MPPs). The pumps boost the flow pressure to guarantee transportation, through a riser, to the FPSO. A complex and detailed simulation model of the whole network is available, which integrates physics-based models of the different subsystems of the networks, such as models of wells, gathering system (piping and manifolds), MPF pumping station, riser, separator, etc. Moreover, the monitoring data of the real network collected over a period of two years is also available, which include the real-time measurements of key variables, such as the opening of the well's choke valves, flowrates (oil, gas, and water), temperatures, and pressures at different locations of the network, beside the speed and power of the MPPs.

The primary challenge addressed by this work is modeling the behavior of the pumps. On one hand, the pump manufacturer's curves require, as inputs, the total liquid flow at suction, pump speed, and gas liquid ratio. They provide outputs including the pump pressure rise and power. However, it has been observed that the results obtained from

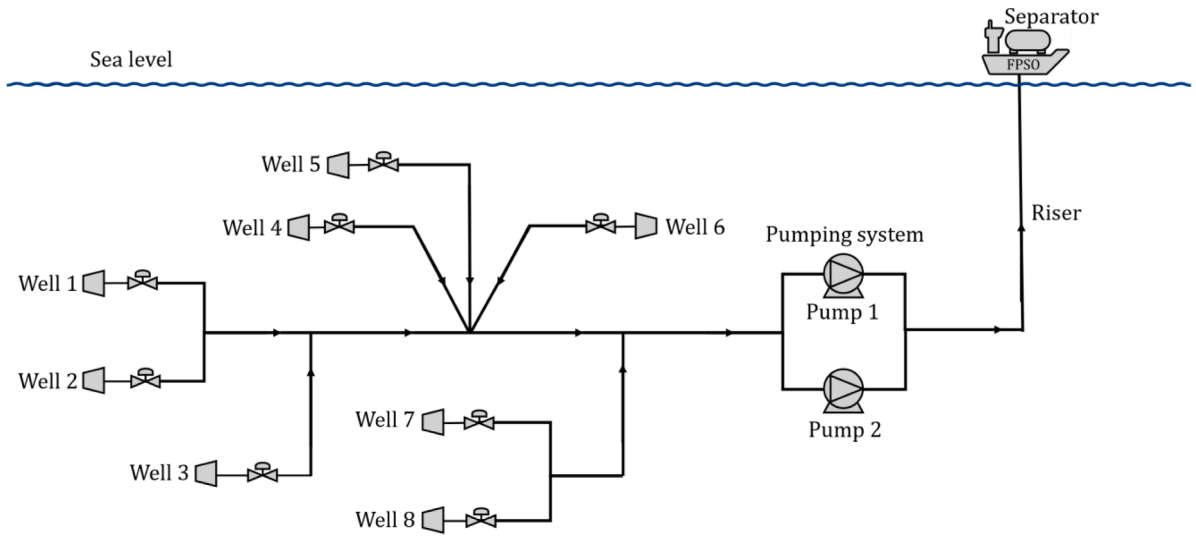


Fig. 2. Schematization of the real production network of the case study.

these curves rarely align with the pressure rise and power data actually measured in the field. Moreover, no single curve is able to accurately describe the pump's behavior under all conditions; instead, different curves are needed depending on the gas-liquid ratio (GLR). This variability complicates their application. On the other side, the available physics-based model of the MPPs is largely inaccurate compared to the real behavior measured from the real pumps. In fact, different modelling approaches have been attempted based on the first principle knowledge, but none of them were able to reliably describe the behavior of the pumps. Actually, multiphase flow metering and modeling are very challenging problems that have been frequently reported in the literature (Andrianov, 2018). In order to overcome the lack of knowledge about the pump behavior, this work proposes to exploit the monitoring data measured from the real pumping station and use them to develop Machine Learning (ML) models able to mimic the pumps behavior. The ML-based pump models are then integrated with the physics-based models of other subsystems (e.g., wells, gathering network, riser, separator, etc.) to constitute an overall hybrid model of the production facility. Another challenge that this work has innovatively treated is the design of the ML-based models of the pumps in such way that makes them compatible and conciliated with the numerical flow of the physics-based models of the other subsystems in the oil and gas production facility.

After its validation, the hybrid model is used for the steady-state production optimization by means of a Differential Evolutionary (DE) algorithm. To assess the robustness and the relevance of the proposed method, the production optimization is carried out at different time periods and in different operational conditions; then, the obtained optimal results are compared to the actual production.

Addressing the three previously mentioned limitations leads to the main contributions of the work, which can be summarized as follows:

i) Development of an accurate ML model for a train of MPPs, built using real-time monitoring data collected over three years of field operation. The model predicts pump power, suction pressure and outlet flow temperature as functions of key operational variables: oil, water and gas inlet flow rates, inlet flow temperature, outlet flow pressure and pump speed. Although the ML model is trained exclusively on field data, its input and output variables are deliberately designed to align with those used in the platform's physics-based models, enabling seamless integration. This contribution addresses the aforementioned first limitation (item (a)): unlike traditional ML models trained on synthetic or simulation data, this model captures the actual behavior of the pumps under real operating conditions. This distinction is critical, as the existing physics-based pump model of the platform lacks the accuracy needed to represent true pump performance.

ii) Development of a hybrid model for a real-world, large-scale oil and gas production facility, which integrates ML-based models of pumps (developed in item i) with high-fidelity, physics-based models of other components and subsystems, including wells, gathering network, risers and separators. These components and subsystems are modeled using the Integrated Production Modelling (IPM) suite — a comprehensive environment for the modeling and simulation of hydrocarbon reservoirs and surface production facilities. The IPM suite leverages fundamental physical principles, including multiphase flow dynamics, heat and mass transfer and pressure equilibrium, to ensure a high-fidelity, robust and accurate representation of the facility's operational behavior. The good prediction accuracy of the hybrid model has been evaluated against the real behavior measured from the field. This contribution addresses the second limitation in item (b).

iii) Steady-state production optimization of a real hydrocarbon facility using the hybrid model developed in item ii), coupled with an evolutionary differential algorithm. The optimization's numerical results are validated with respect to the actual production status of the real field, demonstrating significant performance improvements across multiple time periods. This validation effectively addresses the third identified limitation in item (c).

2. Case description and problem statement

This work considers the steady-state production optimization of a real offshore oil and gas production network located in a west central African country. The production system covers an area of about 3000km^2 and the water depth ranges from about 200 m to more than 1500 m. The network (Fig. 2) consists of eight wells that are connected to two large reservoirs: six wells from the first reservoir and two wells from the second The MPF (oil, water, and gas) is transferred from the reservoirs to the wellheads through the wellbores, which provide a path for the fluid to flow under the pressure difference between the reservoir and the wellbores. The amount of fluid that migrates depends on a number of conditions. These include wellbore pressure, rock properties, average reservoir pressure, fluid properties, flow restrictions near the wellbore and the size and shape of the drainage area (Stanko, 2020). Each wellhead is then equipped with valves that can regulate, the production of hydrocarbons.

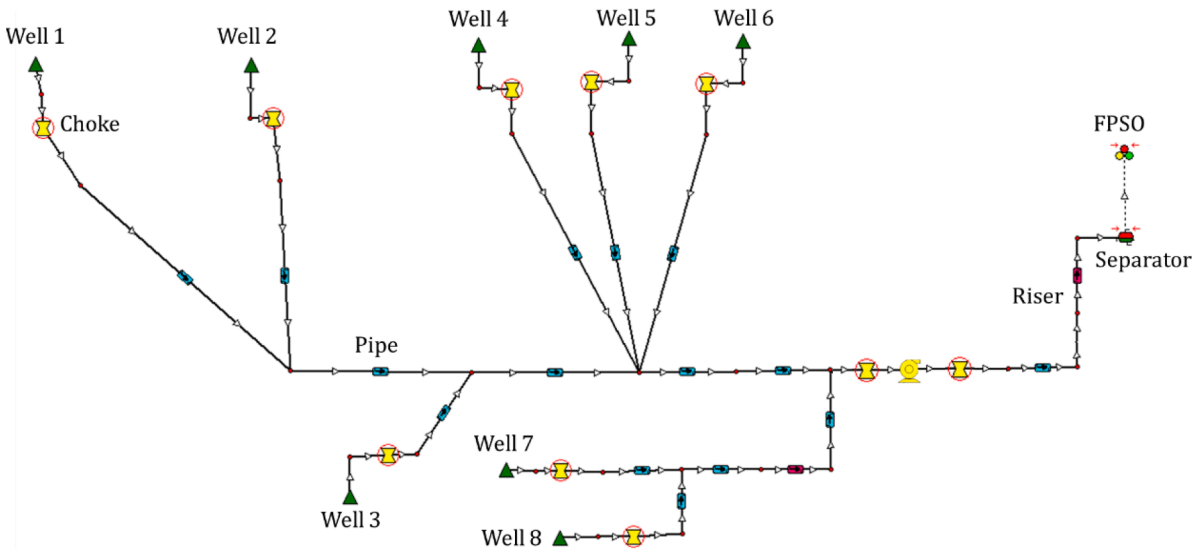


Fig. 3. The network under study, modeled using the IPM suite software.

The outlet flows of the well is collected and combined through the gathering system, which consists of pipelines, valves and pumps. The gathered MPF is transported to a train of two parallel MPPs to increase the flow pressure. Depending on the field operating conditions and maintenance activities, either the two pumps run together or only one of them is in operation. In the case when both pumps work together, each pump handles half of the total flow, while both of them operate at the same speed and consequently the same power. The pressurized flow leaves the pumps and is transported to the FPSO through a riser. A separator with a fixed pressure of 19 bar is installed on the surface of the FPSO, which divides the incoming flow into three phases (oil, water and gas). The crude oil is further treated and then uploaded at regular intervals on a tanker, while gas and water are reinjected in the reservoir for disposal and pressure maintenance.

2.1. *Available models for the facility*

The real network under study is modeled using the Integrated Production Modelling (IPM) software (Petroleum Experts-PE, 2024), which is a popular commercial environment that facilitates the modeling and simulation of an entire oil or gas production system including reservoir, wells and the surface network. Thousands of oil production fields worldwide have been modeled and managed using the IPM software. The IPM suite includes different modeling modules, such as the General Allocation Program (GAP) and the PROSPER environments. To model the wells performance the PROSPER module is used, which is based on a multiphase well and pipeline nodal analysis offering many elements that can be used to describe the vast majority of the physical phenomena occurring in the wells (IPM-Suite, 2024). For the reservoirs, PROSPER allows also to describe their main characteristics such as: pressure, temperature, water cut, gas oil ratio, productivity index etc. (IPM-Suite, 2024).

To model the wells, the following sections are used:

- The Pressure Volume Temperature (PVT) section, which computes the fluid properties, such as temperature pressure, bubble point, gas oil ratio, oil density, oil viscosity, etc. using various correlations provided by the software.
- The Inflow Performance Relationship (IPR) section allows the description of the reservoir deliverability for a given depletion state and assuming that a pseudo-steady state has been reached in the reservoir (Stanko, 2020). The IPR provides the fundamental parameter of the bottom hole pressure (BHP), which is compared to the reservoir pressure to evaluate the driving force for the wellbore inflow.
- The equipment section allows the definition of the well hardware (e.g., casing, tubing, Xmas tree), the deviation survey which is a reflection of the path the well takes to the surface, and the formation temperature profile. This section also manages the Vertical Lift Performance (VLP) curves, which describe the relationship between the top node pressure and the BHP for various flow rates under given conditions (IPM-Suite, 2024).
- The analysis section ensures if the well model behavior matches with the conditions measured in the field.

To model the rest of the production facility (i.e., the gathering system, the MPPs and the separator), the GAP module is used, which is a multiphase network modelling tool. The GAP offers a huge library of physics-based models of different equipment/elements of the network such as pipelines, chokes, compressors, pumps, separators, etc. (IPM-Suite, 2024). The surface network model can be developed by selecting and connecting the required items. For each item/equipment, specific design (e.g., diameter) and operational (e.g., temperature) parameters must be set according to the real case. For example, the main characteristics of a pipeline include its geometry, heat transfer coefficient, fluid heat capacity and environmental parameters such as the ambient temperature. For throttle valves, the main parameters are valve opening diameter, maximum and minimum opening diameter and a correction coefficient. In the GAP environment, the separators represent nodes where the pressure is fixed. In other words, the separator unit in GAP does not necessarily represent an actual separator in the field, but just any fixed pressure points in the system, i.e., a boundary condition. Note that a point in the network, where two or more items are connected, is called a “joint,” where pressure and mass balance must be achieved. The modelled network with Gap results as shown in Fig. 3:

After the development of the surface network model, the aforementioned models of the wells built using the PROSPER are integrated into it to represent the whole production network model. The solution of the overall network model is based on balancing the pressure, mass flow and temperature from all items. The only fixed points are the pressure at the separator and the pressures and temperatures specified in the reservoirs. This is done by solving the following system of equations for each joint (red dots in Fig. 3):

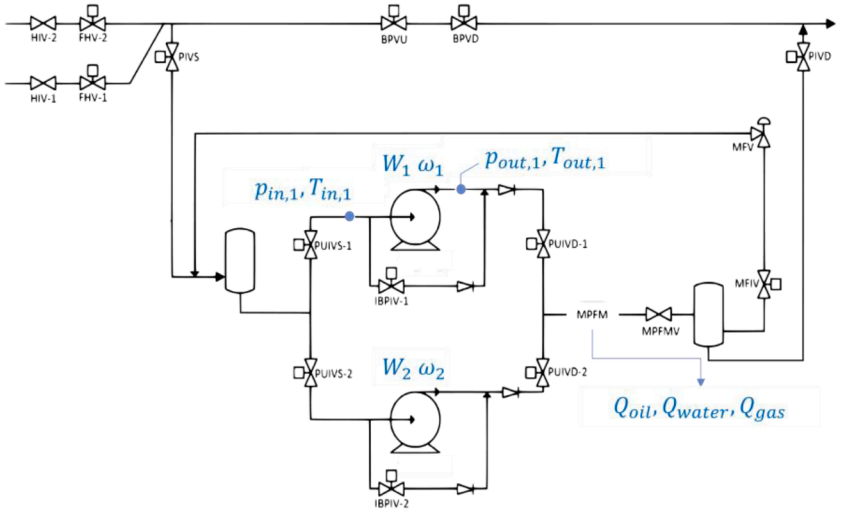


Fig. 4. Schematization of the pumping station, showing the locations where data was collected, except for the valve openings of the wells.

$$\begin{cases} \left(\dot{m}_{oil}+\dot{m}_{gas}+\dot{m}_{water}\right)_{in} = \left(\dot{m}_{oil}+\dot{m}_{gas}+\dot{m}_{water}\right)_{out} \\ p_{us} - p_{ds} - \Delta p = 0 \\ \Delta p = f(\dot{m}, p, T, L, \cdots) \\ p_{end} = C \end{cases} \tag{1}$$

where $\dot{m}$ is the total mass flow $[kg/h]$, p_{us} and p_{ds} are the upstream and downstream pressures of the joint $[bar]$, Δp is the pressure drop across the pipe $[bar]$, can be calculated with correlation models. The pressure p, temperature $T[^{\circ}C]$, and length of the pipe $L[m]$, that with the other parameters contribute to the pressure drop calculation along the pipe. p_{end} is the pressure $[bar]$ at the end point of the network at the separator and C is the fixed pressure value (IPM-Suite, 2024). This system of equations is solved through an iterative procedure, where the convergence is achieved if the mass and pressure balances are preserved at each node of the system within a certain tolerance defined by the user.

To define pipe and fluid temperatures, the "Rough Approximation Temperature Model" was used in GAP. This model was used to reduce the computational time of network resolution. The model assumes that the heat transferred between fluids and the surrounding environment (sea water) by all the different heat transfer mechanisms (such as conduction, convection, and radiation) can be calculated using a global heat transfer coefficient, $U\ \left[\frac{W}{m^2K}\right]$. This means that the heat transferred $H_T[W]$ can be calculated as:

$$H_T = UA\left(T_{surr} - T_{fluid,\ avg}\right) \tag{2}$$

where $A\ [m^2]$ refers to the inner area of the pipe, $T_{surr}\ [K]$ is the average surrounding temperature, $T_{fluid,\ avg}\ [K]$ is the average temperature of the fluid in the pipe.

The heat transferred can be described also with the following equation:

$$H_T = \left(\dot{m}_{oil}C_{p,oil} + \dot{m}_{gas}C_{p,gas} + \dot{m}_{water}C_{p,water}\right)\cdot\left(T_{in} - T_{out}\right) \tag{3}$$

where $m\ C_p$ is mass flow rate of the phase multiplied by its average heat capacity, T_{in} and T_{out} are the temperatures at the inlet and outlet of the pipe. With the resolution of a system of these two equations, T_{out} can be evaluated.

2.2. Available data collected from the real field

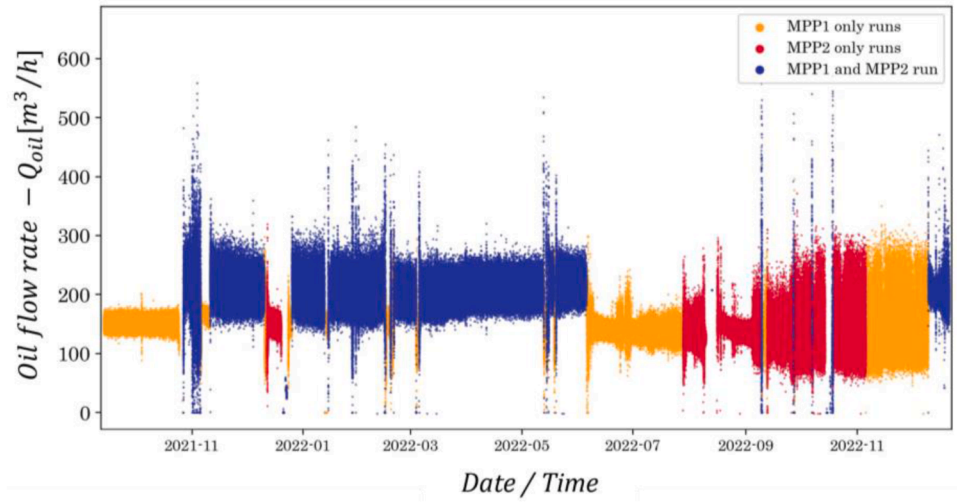
The Eni Digital Oilfield (eDOF) is a system that stores and analyzes, in real time, the operating parameters collected from the field. The authorized user can remotely access the monitoring data history and can acquire them at the preferred frequency (per minute, per day, per month). Using the eDOF, it is possible to obtain the monitoring data of the variables considered in this study (illustrated in Fig. 4), which are:

- The total outlet fluid flowrates of the network: $Q_{oil}, Q_{gas}, Q_{water}\ \left[\frac{m^3}{h}\right]$
- The speed of the pumps: $\omega_1,\ \omega_2\ [rpm]$
- The temperatures at the suction of the pump train $T_{in,1}, T_{in,2}$ [°C] and at the discharge $T_{out,1}, T_{out,2}$ [°C]
- The pressures at the suction of the pump train $p_{in,1}, p_{in,2}$ [bar] and at the discharge $p_{out,1}, p_{out,2}$ [bar]
- The power of the pumps: W_1, W_2 [kW]
- The choke openings: $\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7, \delta_8$ [inches]
- Other data, such as GOR, WC, temperatures and fluids from different branches of the pipes etc.

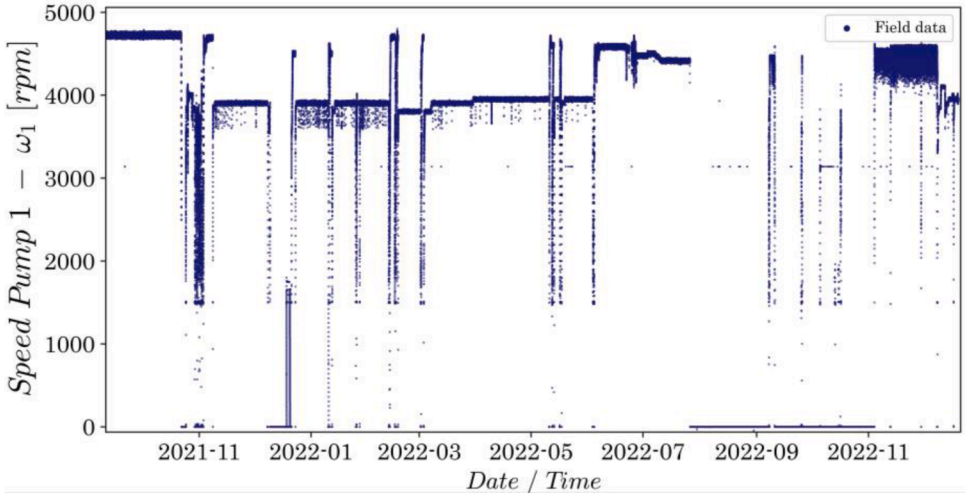
The data considered have been acquired during a period from September 2021 to December 2022 with a sampling period of one minute, which results in a dataset of 700,000 instances. As example, Fig. 5 shows the real measurement of some of the aforementioned variables. Fig. 5-(a) also illustrates three distinct operational regimes: periods during which only one of the pumps is active—either pump-1 (orange) or pump-2 (red)—resulting in reduced oil flow rates, and periods when both pumps operate simultaneously (blue). Note that when both pumps run in parallel they equally share the gathered flowrate from the wells. This configuration leads to approximately identical operating conditions for both pumps: i.e. $T_{in,1} = T_{in,2}$, $T_{out,1} = T_{out,2}$, $p_{in,1} = p_{in,2}$, $p_{out,1} = p_{out,2}$, $\omega_1 = \omega_2$, and $W_1 = W_2$. Also in Fig. 5, it can be noticed that the network is operated most of the time in steady state regime (it can be seen when the signals are stationary); however, there are short periods of transient conditions due to several reasons, such as changing the operating configuration (running either one or two pumps), maintenance, or transition between different conditions (e.g. different oil production targets).

On the other hand, a large amount of noise can be noticed, which is typically encountered in the monitoring data of subsea assets due to many reasons that arise in such harsh environment. Particularly, in our case, this is mainly due to the slugging of the pipes upstream of the pump train, and also the increase of the GOR and WC through time, which exacerbate fluid turbulences.

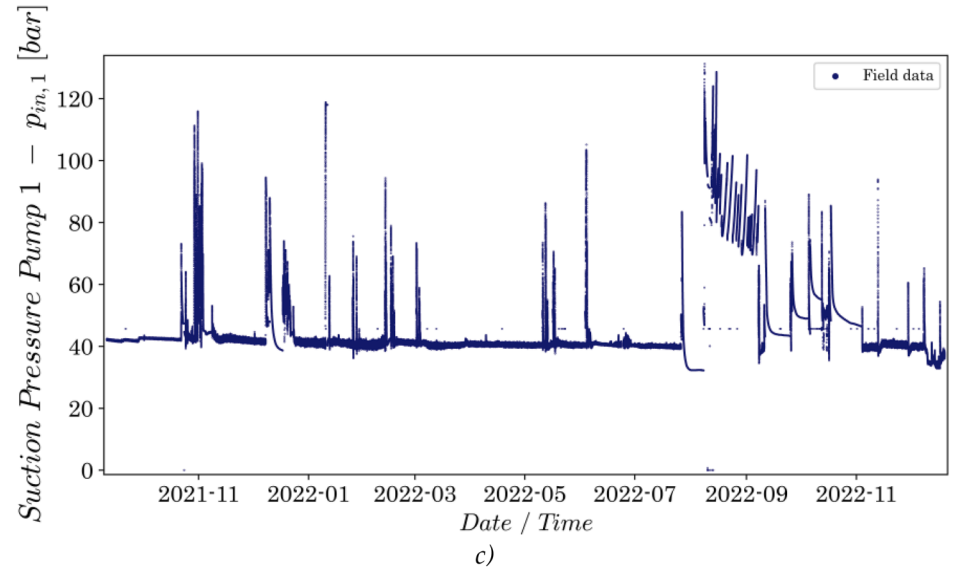
The rest of the data are shown in Fig. 23 of the Annex.



(a)



(b)



c)

Fig. 5. Data collected from the real field: a) Oil flow rate (Q_{oil}), b) Speed of Pump 1 (ω_1), and c) Suction pressure of Pump 1 $p_{in,1}$.

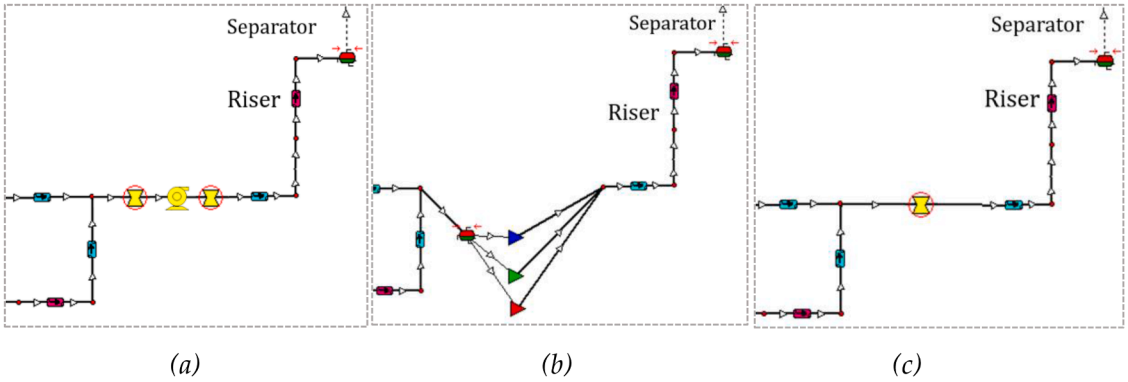


Fig. 6. The three modeling approaches the company used to replicate pump behavior:: (a) using a pump item; (b) using a separator to replicate the characteristic curves; (c) using a valve.

2.3. Work objective: production steady state optimization

The objective of this work is the optimization of the oil production at steady-state conditions. The quantity of the produced oil Q_{oil} depends on the reservoir pressure, the natural drop of the pressure across the whole system (including wellbores, wellheads and gathering network, riser, FPSO). The chokes openings of the wells and MPP represent discontinuities in the facility pressure profile, which can be manipulated by the operator to optimize the oil production. In other words, the optimization variables in this problem include the choke opening of each well $\delta_i,\ i = 1, \ldots,\ 8$ and the speed of the pumps ω_1 and ω_2. Hence, the optimization problem is formulated as follows:

$$\begin{aligned} &\max_{\substack{\delta_i,\ i=1,\cdots,N_w \\ \omega_j,\ j=1,\cdots,N_p}} Q_{oil} = f(\delta_i, \omega_j) \\ &\text{S.T.} \\ &0.5 \le \delta_i \le 4[\text{in}] \\ &3000 \le \omega_j \le 4800[\text{rpm}] \\ &m_{tlr}\% < 1\% \\ &p_{tlr} < 0.1\ bar \end{aligned} \tag{4}$$

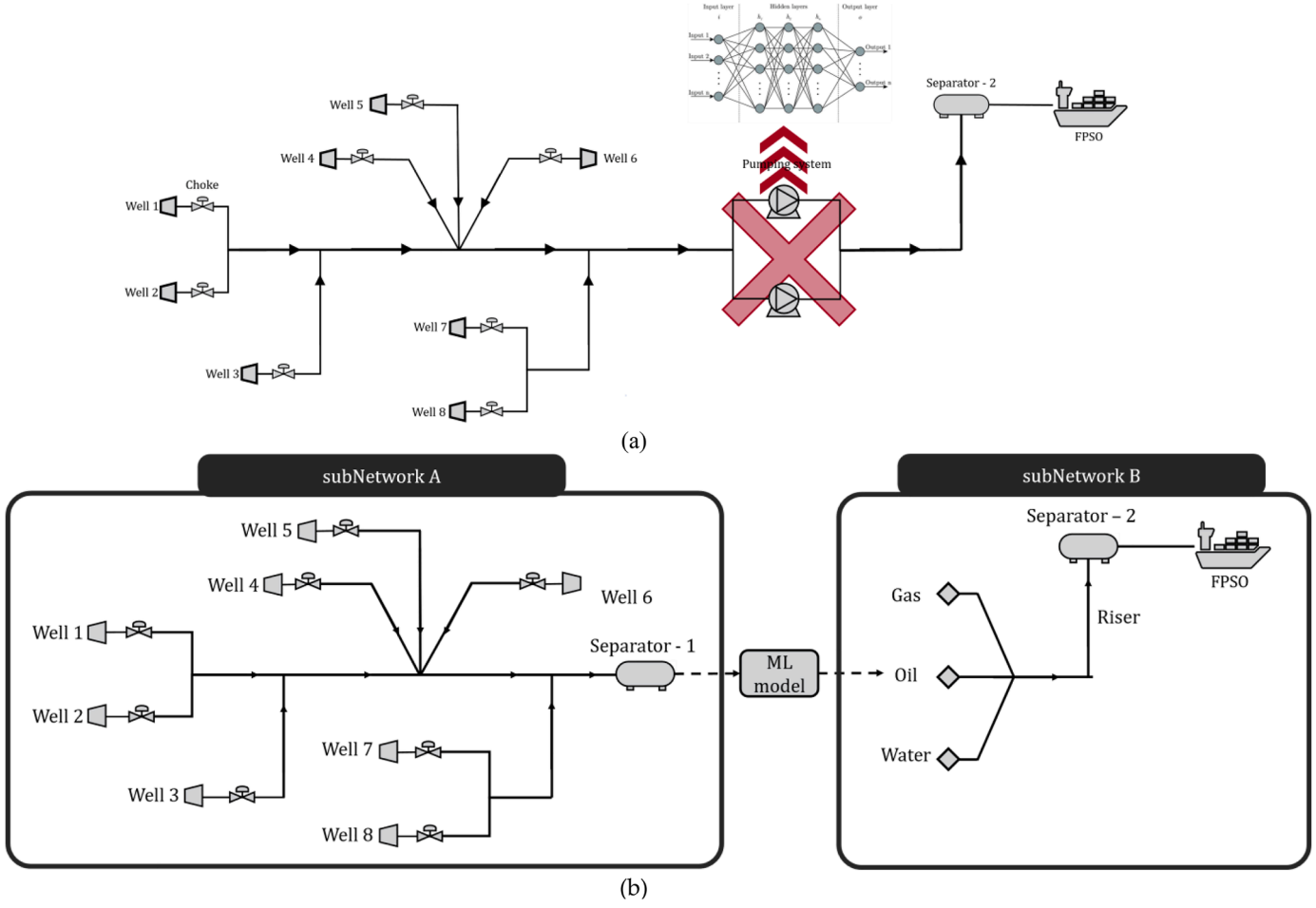


Fig. 7. Representation of (a) the physics-based model of the entire network developed using the IPM suite and of (b) the targeted hybrid model of the network $\mathscr{N}$.

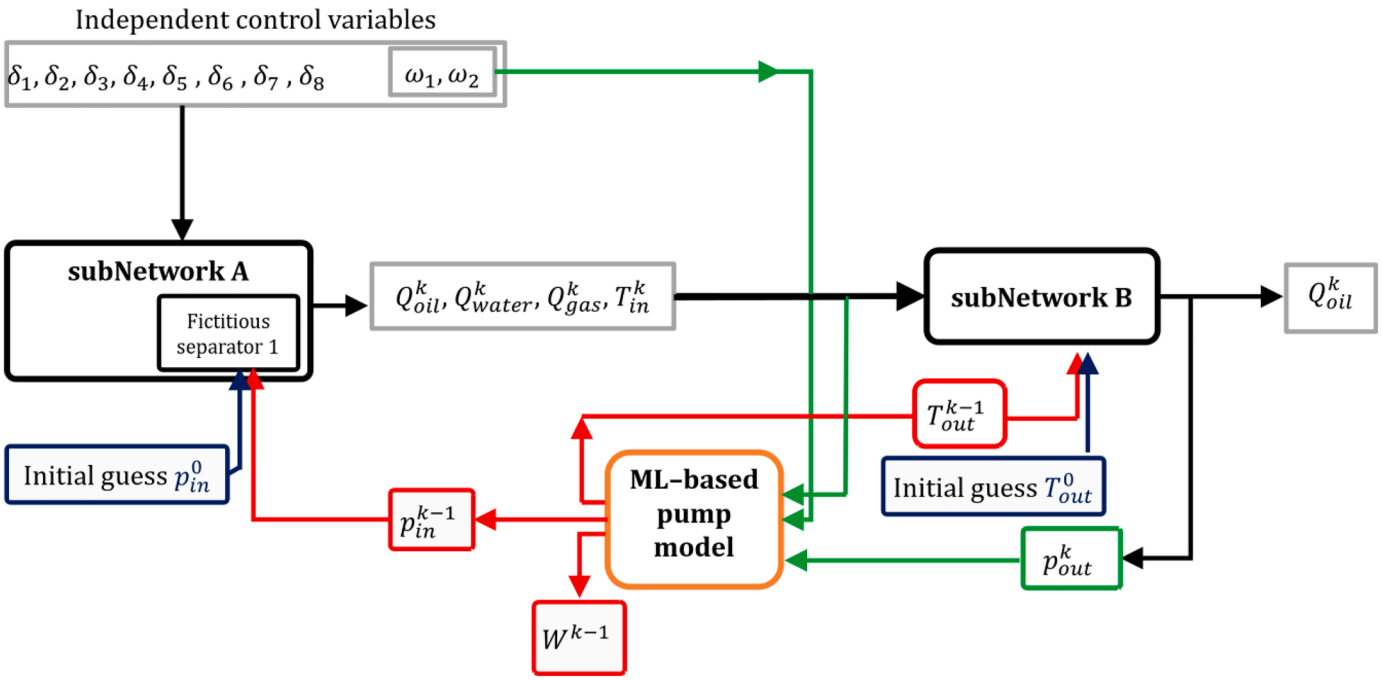


Fig. 8. Representation of the numerical/computational flow in the hybrid model $\mathscr{H}$. The solid black boxes are the Subnetworks modelled using the IPM suite; the blue boxes represent an initial guess of values which are only used to initialize the model before any iteration, i.e., at iteration $k = 0$. The orange box represents the ML-based pump model, whereas the green and red lines represent the inputs and outputs of this ML model, respectively.

where f is the physics-based model of the network described in Section 2.1 and shown in Fig. 3, $N_w = 8$ is the number of wells and $N_p = 2$ is the number of pumps. The symbols m_{tlr} and p_{tlr} are the maximum tolerance values for the mass and the pressure, respectively, which must not be violated by the balance calculations (in Eq.(1)) at each node of the network model during its simulation. In cases when these tolerance values are violated, the network resolution is not valid and the resulting value of Q_{oil} is discarded. Also, it is assumed that the two pumps operate in perfect parallel configuration, where each pump handles half of the total flow with the same speed. Therefore, the optimization variables ω_1 and ω_2 are the same, and from now on, they will be denoted by ω. It is also assumed that the suction temperature and suction pressure, discharge temperature and discharge pressure are the same between the two pumps (slight differences that may occur in practice are neglected as they are mainly due to small flow instabilities and fluctuations).

The main challenge that hinders the achievement of the aforementioned objective (steady state production optimization of the network) is the lack of an accurate physics-based model for the MPPs. In fact, MPPs are fundamental to the system functionality, as their speed (together with the choke openings) determine the oil quantity produced by the entire system. Actually, three trials have been tested for modeling the pumping station and all of them led to very modest prediction accuracy. The first trial (see Fig. 6(a)) exploited an integrated modular model for the pump, which requires some charts and curves to control the relationship among the flow, pressure rise, and velocity. In the second trail (Fig. 6(b)), a separator modular unit was used to represent the pump. In this case, to replicate the suction pressure of the pump train, a fixed pressure is introduced as a boundary condition. This method severely limits the simulation, as the pump performance is completely absent and therefore the power consumption and the pump speed are unknown. In the third modelling trial (Fig. 6(c)), the idea was to use a valve element to ensure a fixed pressure rise to the flow, having set a constant negative pressure drop. This solution is too simple and has the same problems as the previous one.

3. Methodology and techniques

The proposed methodology consists of four main steps. The first step involves designing a conceptual schematic of the hybrid model representing the entire network, explicitly outlining the numerical integration scheme employed to couple the ML-based pump model with the physics-based models of the other network's components and subsystems. Also, this first step includes the identification of the input and output variables for the ML-based pump model to be developed. In the second step, ML-based pump models are constructed using real data collected from the plant, employing ANNs and RF models. The third step entails developing the hybrid model of the production network and validating it against real field data. Finally, the fourth step incorporates the steady-state production optimization based on the hybrid model, using a DE algorithm.

3.1. Conceptual design of the network's hybrid model

As previously mentioned, the available physics-based models of the pumps are inaccurate, which in turn degrade the prediction accuracy of the physics-based model for the entire network. To address this challenge, the goal is to replace the physics-based pump model with a ML-based model $\mathscr{T}_{ML}$, developed using real data measured at the facility (see Fig. 7-(a)). However, since the ML-based pump model and the physics-based model of the network, denoted as f, operate based on fundamentally different mathematical principles and are implemented in distinct numerical environments or languages, it becomes necessary to divide the physics-based model of the full network into two separate subnetwork models (see Fig. 7-(b)). These include a model for the subnetwork upstream of the pump (subnetwork-A), denoted as f^{Sub-A}, and a model for the subnetwork downstream of the pump (subnetwork-B), denoted as f^{Sub-B}. The ML-based pump model is, then, placed and integrated between these two physics-based models of the two subnetworks to eventually constitute the overall network's hybrid model $\mathscr{H}$, such that $\mathscr{H} = f^{Sub-A} \oplus \mathscr{T}_{ML} \oplus f^{Sub-B}$.

Note that in the original physics-based model of the network (Fig. 2),

a single separator is used at the surface of the FPSO, which reflects the real or physical configuration. However, to enable partitioning the physics-based model of the network, f, and to obtain the subnetworks' models f^{Sub-A} and f^{Sub-B} (Fig. 7-(b)), an additional - but fictitious - separator is introduced at the end of subnetwork-A (i.e., Separator-1 in Fig. 7-(b)). This modification is required due to the modeling rules of the IPM suite, which mandate that any model of a production network must terminate with a fixed-pressure unit, typically represented by a separator. Accordingly, the pressure of Separator-1 is set equal to the suction pressure of the pump, since it is positioned just before the pump. On the other hand, the pressure of Separator-2 is set equal to the pressure of the FPSO, which is fixed at 19.0 bar.

Since the models of the subnetworks are detached, it is necessary to pass the flow rate of the fluid from subnetwork-A to subnetwork-B to ensure the mass balance. Therefore, three streams/sources of gas, oil, and water are added as inputs for subnetwork-B (Fig. 7-(b)), which are directly mixed in one stream and used as inlet for the riser and Separator-2. The mixed stream is characterized by 1) the flow rates of the three phases, equal to the outlet flowrate of subnetwork-A, ii) a pressure which is determined by the pressure of Separator-2, and iii) a temperature equals to the temperature of the outlet flow from the pump. The chemical properties of the different phases in the inlet streams to subnetwork-B, (e.g., oil density, water density, water salinity, percentage of N_2, CO_2) are kept exactly the same as those of subnetwork-A.

The final element in the hybrid model is the ML-based pump model: although it will be developed using real data collected from the plant, its functionality must be compatible with the numerical flow of the two subnetworks' models to which it will be integrated. This makes it necessary to understand the numerical flow of the targeted hybrid model shown in Fig. 8.

It is worth noting that only the data collected from Pump 1 are considered in this study, for the following reasons: i) both pumps are of the same commercial type and share identical design and specifications; ii) when operating simultaneously, the two pumps equally divide the gathered flowrate from the wells, resulting in identical operating conditions (e.g., $T_{in,1} = T_{in,2}$; see Section 2.2); and iii) when only one pump is active, the other remains shut down, and since the pumps are identical, the operating behavior remains representative.

For given values of the nine independent control variables, i.e., the eight valve openings $\delta_i,\ i = 1,\ldots,8$ and the pumps speed $\omega_1 = \omega_2$, the hybrid model converges through an iterative loop. At each iteration k:

i. If $k = 0$, make an initial guess of the pressure of the end unit in the model of subnetwork-A (i.e., pressure of Separator-1), which also represents the pump's suction pressure, p^0_{in}; otherwise, consider the pressure value predicted by the ML-based pump model p^{k-1}_{in} in the pervious iteration.
ii. Run the model of subnetwork-A and obtain the volumetric flowrates Q^k_{oil}, Q^k_{gas}, Q^k_{water} and their temperature T^k_{in}.
iii. If $k = 0$, make an initial guess of the temperature of the inlet flow to subnetwork-B, T^0_{out}; otherwise, use the temperature value predicted by the ML-based pump model T^{k-1}_{out} in the pervious iteration,
iv. Use, as input, the values of the flow rates, Q^k_{oil}, Q^k_{gas}, Q^k_{water} and the inlet flow temperature to run the model of subnetwork-B to obtain the outlet pressure p^k_{out}.
v. Use the values ω_1, ω_2, Q^k_{oil}, Q^k_{gas}, Q^k_{water}, T^k_{in}, and p^k_{out} as input for the ML model to obtain its outputs p^k_{in}, T^k_{out}, and W^k, which will be used to setup the next iteration $(k+1)$ starting from step i) to obtain the new values of the volumetric flowrates Q^{k+1}_{oil}, Q^{k+1}_{gas}, Q^{k+1}_{water}, and so on.
vi. Stop the iteration if the flow rates difference between two successive iterations k and $k+1$ is less than 1.0 %, such as

$$\left(\frac{|Q^{k+1}_{oil}-Q^k_{oil}|}{Q^k_{oil}}+\frac{|Q^{k+1}_{water}-Q^k_{water}|}{Q^k_{water}}+\frac{|Q^{k+1}_{gas}-Q^k_{gas}|}{Q^k_{gas}}\right)\times 100\% \le 1\%, \quad \text{otherwise}$$

continue the iterations.

The final output of interest of the hybrid model is the amount of produced oil, Q^k_{oil}.

The previous detailed illustration of the network's operational logic, alongside the computational and numerical flow of the hybrid model $\mathscr{H}$, will play a crucial role in guiding the optimal design of the ML-based pump model $\mathscr{T}_{ML}$, as discussed in the following section.

3.2. Development of ML-based models for the pump train

It is important to emphasize that, in process operation and control, the development of an efficient and practically useful ML model for a process or unit depends on two main factors. The first is the intended function of the ML model, such as performance monitoring, or serving as a simplified approximation to accelerate complex computations. The second key factor is the functional logic and operational constraints of the systems with which the ML model will be integrated or expected to interact. For example, if the purpose of the ML-based pump model is monitoring—and assuming the pump freely operates without any imposed operational rules or constraints—its design would logically be structured as follows: the input features would include the flow rates of oil (Q_{Oil}), water (Q_{Water}), and gas (Q_{gas}); the suction pressure (p_{in}); the inlet temperature (T_{in}); and the pumps rotational speed (ω), while the model's predicted outputs would then consist of the discharge pressure (p_{out}), outlet temperature ($\widehat{T}_{out}$) and power consumption (W). However, this is not the case in our problem. The objective here is to leverage real-world field data to develop a ML-based pump model, $\mathscr{T}_{ML}$, that fulfills the following two key requirements:

- It represent an accurate mapping of the pump's behavior, which will be integrated with other physics-based models (f^{Sub-A} and f^{Sub-B}) to eventually obtain an exhaustive hybrid model $\mathscr{H}$ supporting the "steady-state" optimization of the entire network. Since the hybrid model simulates the steady-state behaviour of the facility, the ML-based pump model must perform in steady-state as well. In other words, the ML model will not capture dynamics among its inputs and outputs, but rather, it should approximate a static mapping at certain time instance t.
- It must also adhere to the network's operational logic and constraints, described in Section 3.1, and consequently meets the computational/numerical flow of the physics-based models of the two subNetworks, f^{Sub-A} and f^{Sub-B}, see Fig. 8. This logic implies that the FPSO is required to operate at a fixed and predefined inlet pressure—this pressure is, in fact, equal to the discharge pressure of the pump train. Therefore, the ML-based pump model must be designed to comply with this operational constraint, ensuring that its discharge pressure exactly matches the predefined inlet pressure of the FPSO. In other words, the discharge pressure should be included as one of the input features of the ML-based pump model, while the suction pressure should be treated as one of the predicted output features

Guided by these key requirements, the ML-based pump model, $\mathscr{T}_{ML}$, is formulated as shown in Eq.(5), where the input variables are $[Q_{Oil},\ Q_{Water},\ Q_{gas},\ \omega,\ p_{out},\ T_{in}]$ and the corresponding outputs are $[p_{in},\ W,\ T_{out}]$. Note that the suction pressure, p_{in}, is one of the primary factors—alongside the choke opening of the wells—that governs the amount of flow drawn from the field.

$$\widehat{p}_{in}(t), \widehat{W}(t), \widehat{T}_{out}(t) = \mathscr{T}_{ML}\left(Q_{Oil}(t),\ Q_{Water}(t),\ Q_{gas}(t), \omega(t), p_{out}(t)\ ,\ T_{in}(t)\right) \tag{5}$$

3.2.1. Data processing: filtering and scaling

As noted in Section 2.2, a significant amount of noise is present in the data—an issue commonly encountered in monitoring data from subsea assets due to various factors inherent to such harsh environments. Particularly, in our case, this is mainly due to the slugging of the pipes upstream of the pump train, and also the increase of the GOR and WC through time, which exacerbate fluid turbulences.

To mitigate the high level of noise described in Section 2.2, the data were processed using the Savitzky-Golay (SG) filtering (Schafer, 2011). This filtering technique was chosen for its ability to smooth out noise while preserving essential features and variations in the time series data (Jardim and Morgado-Dias, 2020). The Savitzky-Golay filter fits an n^{th}-order polynomial to the time series data of one variable within a moving window using a least-squares regression. The central point of the fitted polynomial curve is, then, used as the new smoothed data point. Consider a dataset consisting of a univariate time series for observed variable j, with $j = 1,\ldots,N$, and N represents the total number of variables included in the study (see Section 2.2). The time series data are sampled at a uniform frequency, with a constant sampling interval denoted by Δt. To derive the smoothed estimate $\hat{x}_{t,j}$ of a raw observation $x_{t,j}$ measured at an arbitrary time instant t, a local least squares polynomial fitting technique is employed. Specifically, a polynomial of order $\mathscr{C}$ is fitted over a symmetrical sliding window comprising $\mathscr{W} = 2\mathscr{K} + 1$ consecutive samples, where $x_{t,j}$ is located at the center of the window. This configuration ensures that the smoothing process utilizes $\mathscr{K}$ observations preceding and $\mathscr{K}$ observations succeeding the target point $x_{t,j}$ (Luo, et al., 2005) .

For a given data point $x_{t,j}$, the SG filter can be mathematically expressed as follows (Schafer, 2011):

$$\begin{bmatrix} x_{t-\mathscr{K}\Delta t,j} \\ \vdots \\ x_{t,j} \\ \vdots \\ x_{t+\mathscr{K}\Delta t,j} \end{bmatrix} = \begin{bmatrix} b_0 + b_1(t-\mathscr{K}\Delta t) + b_2(t-\mathscr{K}\Delta t)^2 + \cdots + b_{\mathscr{C}}(t-\mathscr{K}\Delta t)^{\mathscr{C}} \\ \vdots \\ b_0 + b_1(t-0\Delta t) + b_2(t-0\Delta t)^2 + \cdots + b_{\mathscr{C}}(t-0\Delta t)^{\mathscr{C}} \\ \vdots \\ b_0 + b_1(t+\mathscr{K}\Delta t) + b_2(t+\mathscr{K}\Delta t)^2 + \cdots + b_{\mathscr{C}}(t+\mathscr{K}\Delta t)^{\mathscr{C}} \end{bmatrix} = \begin{bmatrix} \alpha_0 + \alpha_1(-\mathscr{K}) + \alpha_2(-\mathscr{K})^2 + \cdots + \alpha_{\mathscr{C}}(-\mathscr{K})^{\mathscr{C}} \\ \vdots \\ \alpha_0 + \alpha_1(\mathscr{K}) + \alpha_2(\mathscr{K})^2 + \cdots + \alpha_{\mathscr{C}}(\mathscr{K})^{\mathscr{C}} \end{bmatrix}$$

$$\begin{bmatrix} x_{\,t-\mathscr{K}\Delta t,j} \\ \vdots \\ x_{t,j} \\ \vdots \\ x_{\,t+\mathscr{K}\Delta t,j} \end{bmatrix} = \begin{bmatrix} 1 & -\mathscr{K} & \cdots & -\mathscr{K}^{\mathscr{C}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathscr{K} & \cdots & \mathscr{K}^{\mathscr{C}} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{\mathscr{C}} \end{bmatrix} = \mathscr{J}\boldsymbol{\alpha} \tag{6}$$

where $\alpha_0,\ldots,\alpha_{\mathscr{C}}$ are the coefficients of the polynomial curve. The smoothed values of the samples within the window, denoted by $\hat{\mathbf{x}}$, can be calculated as in Eq.(7).

$$\hat{\mathbf{x}} = \mathscr{J}\left(\mathscr{J}^{\mathrm{T}}\mathscr{J}\right)^{-1}\mathscr{J}^{\mathrm{T}}\boldsymbol{x} \tag{7}$$

Scaling

In order to transform the raw data of the input variables to features of the same distribution/scale, the previously filtered data are normalized using the following formula:

$$x'_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j},\ i = 1,..,n,\ j = 1,\cdots,N \tag{8}$$

where $x'_{i,j}$ is normalized data, $x_{i,j}$ is the original value of the sample, n is the number of samples, N is the number of variables involved in the study, μ_j and σ_j are the mean and the variance of the $j-th$ variable, respectively.

Note that both the scaling and filtering steps are applied independently to the univariate time series data of each variable j, where $j = 1,\ldots,N$, regardless of the variable's role in the ML model (whether as an input or output). This is because both scaling and filtering are just data preprocessing steps, and no ML model has been trained or developed at this stage.

3.2.2. ML models: artificial neural network and random forest

Once the data are properly filtered and scaled, they can be used to build the ML model expressed in Eq.(5). Two types of supervised ML techniques for regression are adopted in this work to represent the function $\mathscr{F}_{ML}$ in Eq. (5), which are ANNs and RF. The purpose is to test the methodology performance using ML models belonging to two different classes, namely parametric and non-parametric models.

3.3. Development and validation of the network's hybrid model

Once the ML-based pump model is built and proven to possess satisfactory prediction accuracy, it is integrated to the physics-based models of subNetworks A and B as described in Section 3.1, Figs. 7, and 8, to form the hybrid model $\mathscr{H}$ of the entire network. The predictions of $\mathscr{H}$ must then be validated against the behavior of the real network over selected time periods from the facility's operating history (e.g., a week or month), following the procedure outlined below:

- Prepare the real input values required to run the hybrid model $\mathscr{H}$. These values include:
  - Field measurement of the manipulated/control variables over the selected validation period, which involve the valves opening $\delta_i,\ i = 1,\ldots,8$ and the pump speed ω.
  - The PVT, IPR and VLP curves (PROSPER files) that describe the state and conditions of the reservoir and the wells throughout the selected validation period.
- Run the hybrid model to obtain the simulated output over the specific validation period, which are the flow rates $\widehat{Q}_{oil},\ \widehat{Q}_{water}, \widehat{Q}_{gas}$, the pressures $\widehat{p}_{in}, \widehat{p}_{out}$ and the pump power $\widehat{W}$.
- Evaluate the hybrid model's prediction accuracy by comparing the simulated outputs to their corresponding values measured from the real network $Q_{oil},\ Q_{gas}, Q_{water}, p_{in}, p_{out},\ W$.

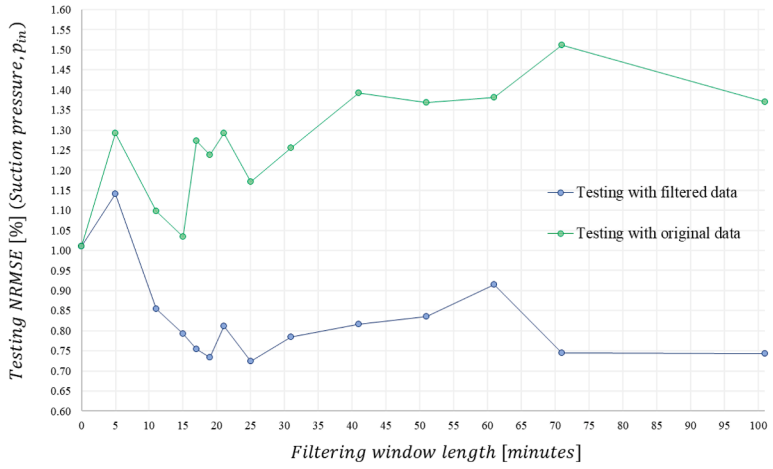
In this study, the Normalized Root Mean Square Error (NRMSE) serves as the accuracy metric for evaluating the performance of the hybrid model. For instance, when assessing the prediction of the oil flow rate, it can be computed as in Eq. (9), where $Q_{oil,max}$ and $Q_{oil,min}$ are the maximum and minimum values of the oil flowrate in the dataset, respectively, and n denotes the number of patterns use for the hybrid model validation:

$$NRMSE_{Q_{oil}} = 100 \times \sqrt{\frac{\sum_{i=1}^{n}\left(Q_{oil,i} - \widehat{Q}_{oil,i}\right)^2}{n}} \times \frac{1}{Q_{oil,\max} - Q_{oil,\min}} \tag{9}$$

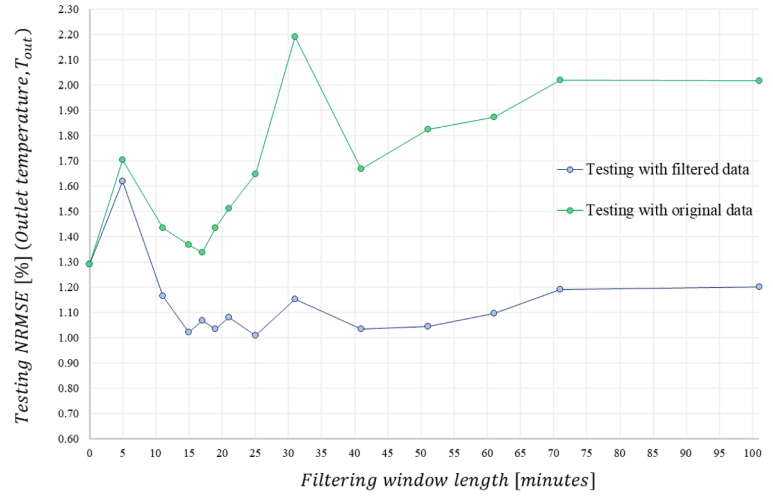
3.4. Steady-State production optimization

Once the hybrid model $\mathscr{H}$ is validated, it can be used as the basis for the network's production optimization. Therefore, the optimization problem formulated in Section 2.3 and Eq.(4) can be reformulated into Eq. (10), where the hybrid model $\mathscr{H}$ takes the place of the physics-based model f.

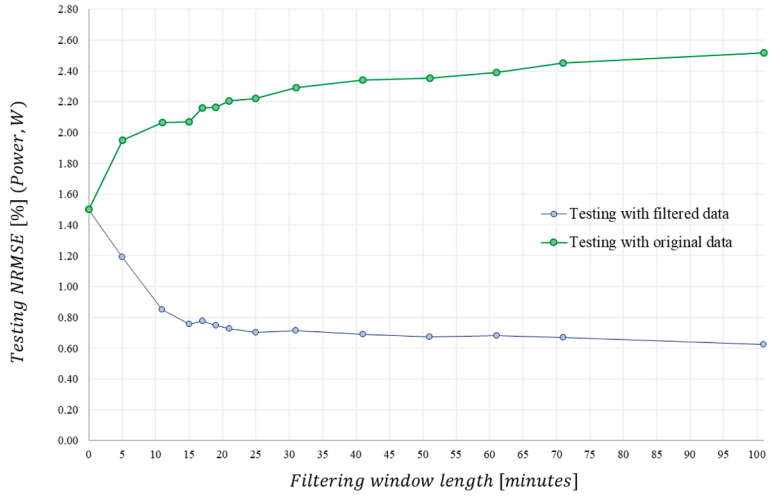
Given the complex structure of the hybrid model, the use of deterministic optimization algorithms becomes challenging due to the difficulty in calculating derivatives. As a result, derivative-free optimization algorithms are an obvious choice. Therefore, this work adopts the popular DE algorithm, which is a class of genetic algorithms that simulates the mechanisms of natural biological evolution (Huang and Chen, 2013). The DE algorithm has been shown to exhibit superior global convergence and robustness compared to classical genetic algorithms in numerous studies. More details about the DE algorithm can be found in



(a)

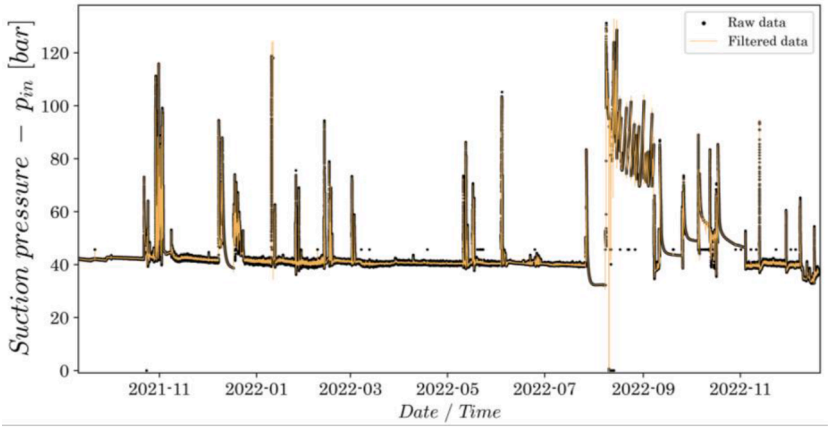


(b)

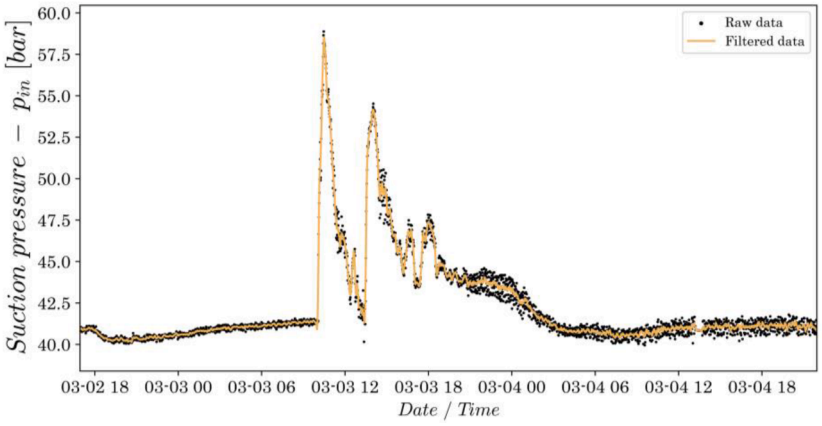


(c)

Fig. 9. NRMSE of the ANN versus the filtering window length: (a) Suction pressure p_{in}, (b) Discharge Temperature T_{out}, and (c) Power W.



(a)



(b)

Fig. 10. Application of the SG filter with a window length of 15 min to the suction pressure data: (a) the entire signal and (b) a zoom-in view of the 3th March 2022.

Annex.

$$\left.\begin{array}{c} \max\limits_{\substack{i=1,\cdots,N_w \\ j=1,\cdots,N_p}} Q_{oil} = \mathscr{H}(\delta_i, \omega_j) \\ S.T. \\ \mathscr{H} = f^{Sub-A} \oplus \mathscr{F}_{ML} \oplus f^{Sub-B} \\ 0.5 \leq \delta_i \leq 4 \text{ [in]} \\ 3000 \leq \omega_j \leq 4800 \text{ [rpm]} \\ m_{tlr}\% < 1\% \\ p_{tlr} < 0.1 \ bar \end{array}\right\} \quad (10)$$

In Eq. (10), $\mathscr{H} = f^{Sub-A} \oplus \mathscr{F}_{ML} \oplus f^{Sub-B}$ represents the network's hybrid model, where f^{Sub-A} is the physics-based model of subNetwork-A, $\mathscr{F}_{ML}$ is the ML-based pump model, and $f^{Sub-B}$is the physics-based model of subNetwork-B.

4. Application and results

In order to maintain clarity and reinforce the logical progression of ideas, the application section presents and discusses the obtained results following the same sequence as the methodological steps outlined in Section 3. It is important to note that no additional results are provided corresponding to Step 1 in Section 3.1, as this step focuses solely on the "conceptual" and high-level design of the network's hybrid model, which has already been comprehensively discussed. As such, the application section begins with the development of the ML-based pump models, where the practical implementation of the methodology is initiated.

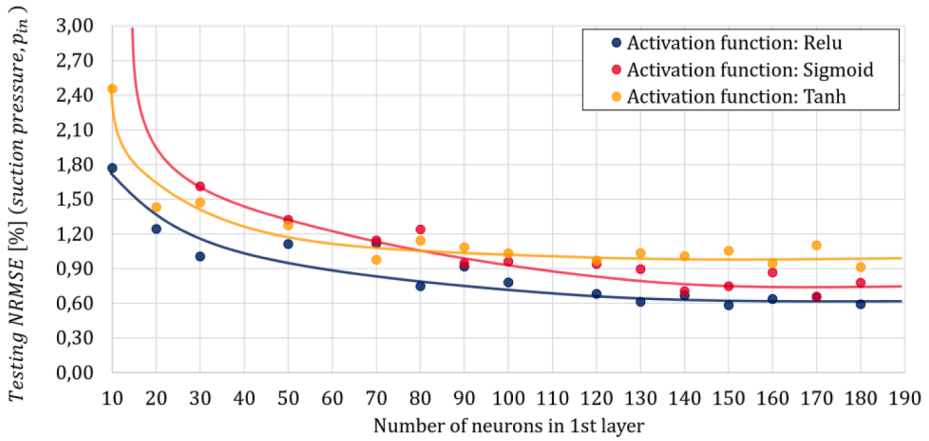
4.1. Development of ML-based models for the pump train

4.1.1. Data processing: filtering and scaling

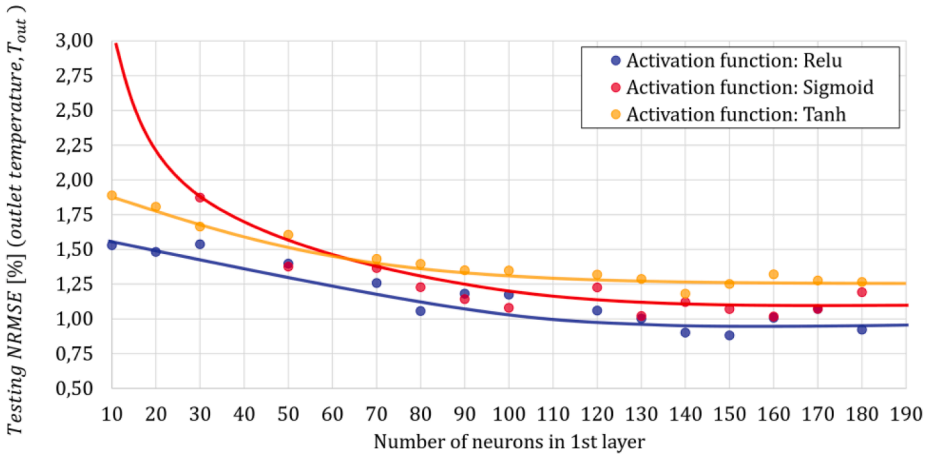
The entire dataset (refer to Section 2.2) was analyzed and a total of 1978 patterns were identified as containing NaN (Not a Number) values. Given that these patterns represent a negligible portion of the overall dataset—which comprises approximately 700,000 patterns—they were excluded from the analysis.

With respect to the filtering, there is no deterministic method for the choice of the filtering window length. As a general hypothesis, the window length should neither be too small to preserve large amount of noise in the data, nor too large to distort the latent behavior. We propose a procedure to formalize this general hypothesis: the optimal window length is the one that produces an ML-based pump model—trained on data filtered using that window—which achieves the most consistent and balanced prediction accuracy. This accuracy is evaluated in two scenarios: first, using the unfiltered (raw) testing data subset; and second, using the filtered version of the same testing subset. This dual evaluation enables the assessment of the model's robustness and generalization ability across different data preprocessing conditions. In other words, the proposed numerical procedure determines the optimal filtering window length that best balances data smoothness with the preservation of important information

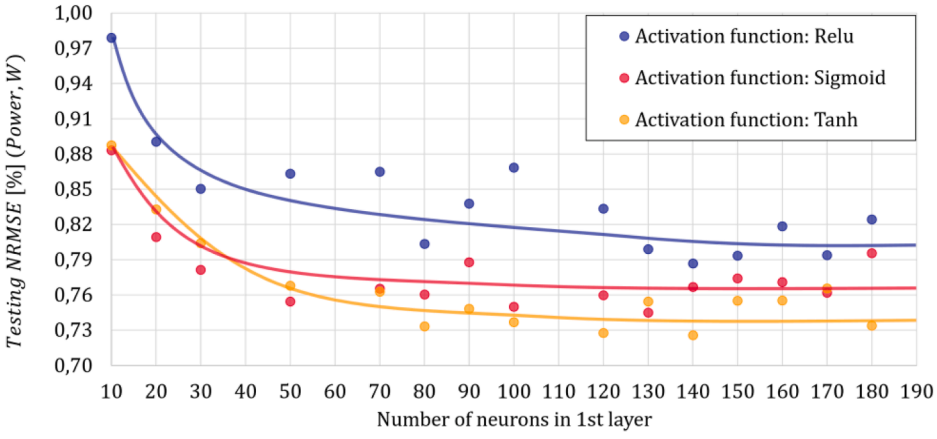
An overestimated range for the window length $w \in \boldsymbol{W}$, $\boldsymbol{W} = [0 - 100]$ minutes is considered, where the values within this range are not



(a)



(b)



(c)

Fig. 11. NRMSE of testing of the three ANNs (blue, red and yellow) versus the number of neurons in the first hidden layer: (a) Suction Pressure p_{in}, (b) Discharge Temperature T_{out}, and (c) Power W.

equally spaced so as to cover a wide range of parameter values with minimum computational cost. It is important to note that the first point on each curve, corresponding to $w = 0$, represents the scenario in which SG filtering is entirely omitted during the ML model development, and the unprocessed raw data is utilized instead.

The procedure is as follows:

i) Set j=1.
ii) Set the SG filtering window length $w = \boldsymbol{W}(j)$ minutes and filter the whole dataset.
iii) Randomly split the dataset into training, validation and testing subsets with split ratios of 70 %, 10 %, and 20 %, respectively.
iv) Train a feedforward ANN to approximate the relation in Eq.(5) using the filtered training and validation subsets. The ANN

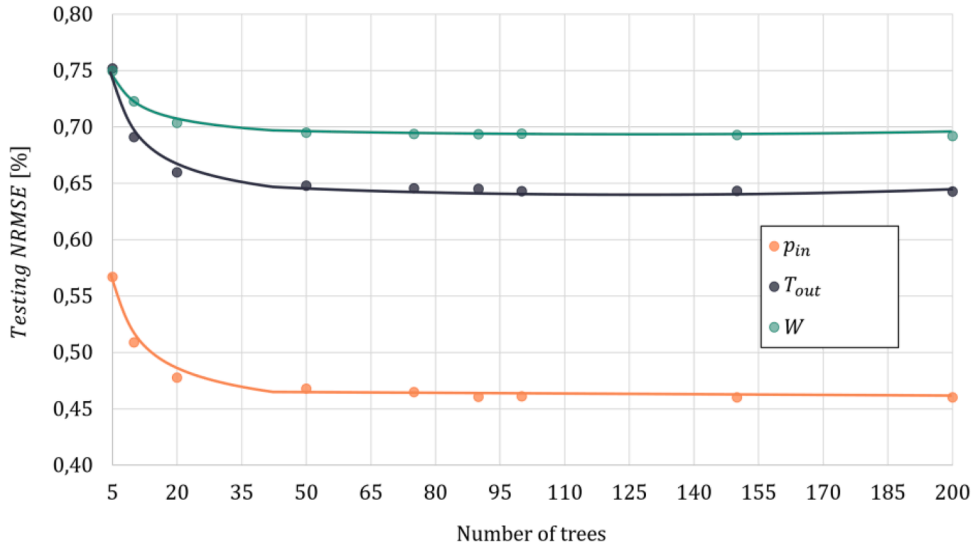


Fig. 12. NRMSE of testing of the RF versus the number of trees.

structure is designed with two hidden layers including 150 and 75 neurons, respectively, with Sigmoid activation function. The training is done using the Adam optimization algorithm, for a maximum of 300 training epochs. Also, to minimize the risk of over-fitting, early stopping is forced if the validation error is not decreasing over 10 successive training epochs.

v) Asses the accuracy of the trained ANN two times: one time using the raw (unfiltered) values of the testing subset, and the other time using their filtered values. Then, report the NRMSE for each of the three outputs (p_{in}, T_{out}, W).

vi) Stop if j= length (***W***), otherwise set $j = j + 1$ and go to step ii).

The results of this procedure are presented in Fig. 9, where each subplot illustrates the NRMSE of each output as a function of the filtering window length, for both testing subset cases: filtered (blue) and unfiltered (green). As expected, the NRMSE is consistently higher when testing on the original (unfiltered) data compared to its filtered counterpart, since the ANN was trained and validated using filtered data subsets. Initially, increasing the window width leads to a reduction in NRMSE, reaching a minimum around a window length of 15 to 20 min. Beyond this point, the error begins to increase again. Consequently, a window length of 15 min was selected as the optimal value. To illustrate the effectiveness of the SG filter at this chosen window length, Fig. 10-(a) displays a comparison between the raw suction pressure data (black) and its filtered version (yellow), while Fig. 10-(b) provides a magnified view for better visualization.

After the filtering, scaling of all the variables has been performed as described in Section 3.2.1.

4.1.2. Artificial neural networks

General guidelines in the literature indicate that a neural network for a regression with two or three layers is sufficient to approximate a wide range of engineering problems (Hagan, et al., 2014). In this work, an ANN including two hidden layers is decided to minimize the risk of overfitting. A sensitivity analysis has been performed to select the best number of neurons in the first hidden layer and the type of transfer function in the hidden layers. As a rule of thumb, the number of neurons in the second hidden layer is set equal to half of that in the first hidden layer. The motivation is to have a squeezing structure that gradually compresses and condenses the latent information from one layer to the next.

First, the number of neurons in the first hidden layer $\mathscr{L}$ is assumed to vary within the range $\mathscr{L} = [10 - 180]$, and the values within this range are not equally spaced so as to cover a wide range of parameter values with minimum computational cost. The following procedure has been applied, independently for each of the considered types of transfer functions (ReLU, Sigmoid, and Tanh):

i) Set j=1.

ii) Set $\mathscr{L} = \mathscr{L}(j)$ neurons, and hence the number of neurons in the second hidden layer is set equal to $\mathscr{L}/2$.

iii) Randomly split the dataset into training, validation and testing subsets with split ratios of 70 %, 10 %, and 20 %, respectively.

iv) Train the ANN to approximate the relation in Eq.(5) using the training and validation subsets. The training is done using the Adam optimization algorithm, with a maximum of 300 training epochs. Also, to minimize the risk of over fitting, early stopping is forced if the validation error is not decreasing over 10 successive training epochs.

v) Assess the performance of the trained ANN using the testing subset, and, then, report the testing NRMSE for each of the three outputs (p_{in}, T_{out}, W).

vi) Stop if j= length ($\mathscr{L}$); otherwise, set $j = j + 1$ and go to step ii).

Fig. 11 shows the NRMSE of the testing of the three ANNs versus the number of neurons in the first hidden layer, where each ANN is characterized by a different color. In general, as the number of neurons increases, the NRMSE decreases until it becomes stable between 120–150 neurons. For the suction pressure (Fig. 11-(a)) and the discharge temperature (Fig. 11-(b)), the ReLU function achieves the best performance, specially at smaller number of neurons, however at higher number of neurons, the Sigmoid transfer function shows competitive performance. For the power (Fig. 11-(c)), the Tanh function shows the best performance with a very slight difference to the Sigmoid type. Therefore, the ANN with 150 neurons in the first hidden layer and a Sigmoid transfer function is selected.

4.1.3. Random forest regression

Similarly, for the random forest model, a sensitivity analysis has been carried out for its performance with respect to the number of trees $\mathscr{T}$. Considering the range $\mathscr{T} \in \mathscr{T}$, $\mathscr{T} = [5 - 200]$, the following procedure is implemented:

i) Set j=1.

ii) Set $\mathscr{T} \in \mathscr{T}(j)$.

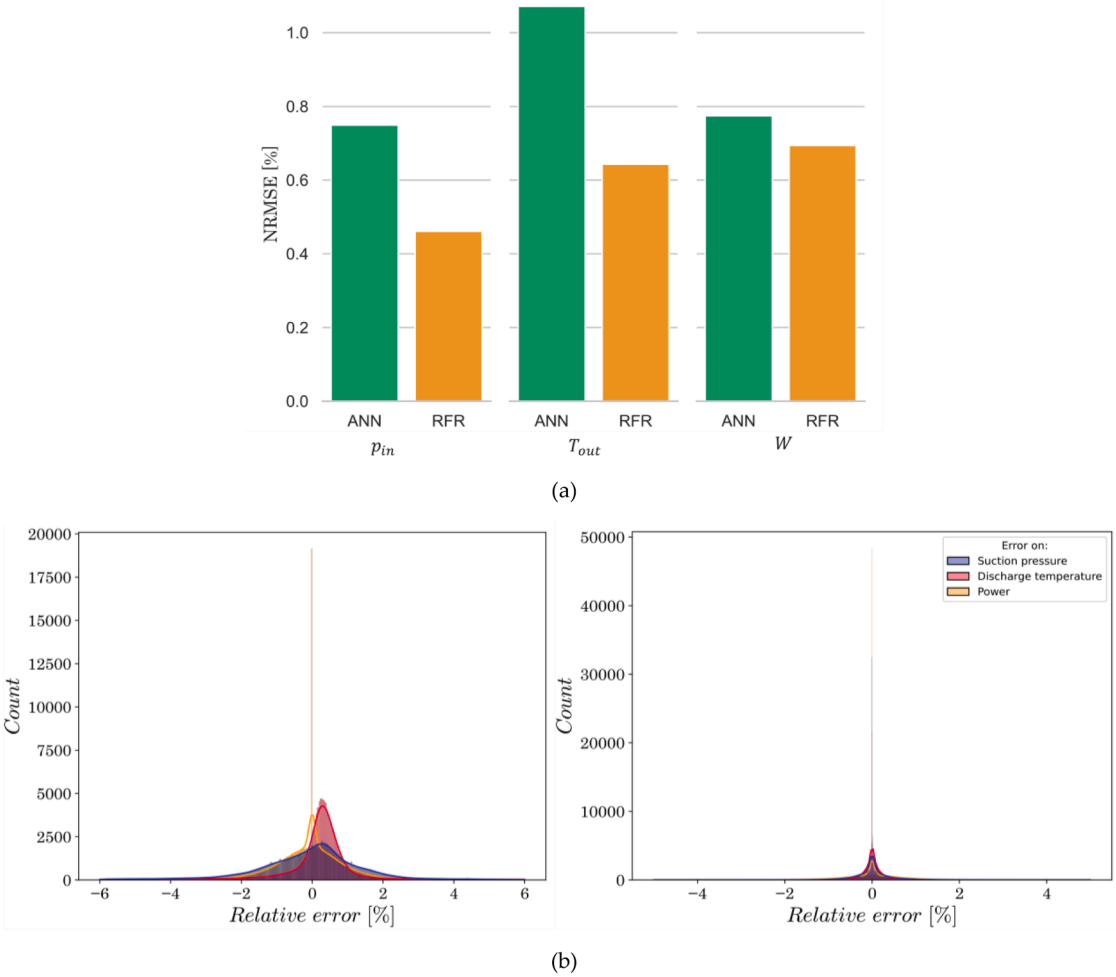


Fig. 13. Comparison between the performances of the best ANN ($\mathscr{L}$ = 150 neurons with Sigmoid activation function) and of the best RF ($\mathscr{T}$=100 trees) models using the testing subset: (a) NRMSE and (b) distributions of the relative errors.

iii) Randomly split the dataset into training, validation and testing subsets with split ratios of 70 %, 10 %, and 20 %, respectively,
iv) Train RF model to approximate the relation in Eq.(5) using the training and validation subsets.
v) Assess the performance of the trained RF using the testing subset, and, then, report the testing NRMSE for each of the three outputs (p_{in}, T_{out}, W).
vi) Stop if $\mathscr{t}$ = length ($\mathscr{T}$), otherwise set $\mathscr{t} = \mathscr{t} + 1$ and go to step ii).

After analyzing Fig. 12, a number of 100 trees has been chosen, since the NRMSE plateau has been reached at this value for all three outputs.

Fig. 13-(a), numerically, compares the performances of the best ANN and the best RF models identified through the previous analysis. The two models achieve very good performances (a NRMSE less than 1 % in most cases), although the RF model shows a slightly higher accuracy. Fig. 13-(b) qualitatively checks the performances of the two models by showing that their relative prediction errors follow normal distribution with a mean value centered close to zero. For a test pattern, the relative error is calculated as the difference between the predicted and the true values divided by the true value.

4.2. Development and validation of the network's hybrid model

The hybrid model of the network is constructed by integrating the ML-based pump model (either the ANN or the RF developed in Section 3.2) into the physics-based models of subNetworks A and B, as described in Section 3.1. The hybrid model is then validated following the steps detailed in Section 3.3. For this purpose, two distinct time periods were selected: April 2021 and November 2021. These periods were chosen based on the following criteria: i) both pumps were operating simultaneously during these months, ii) the reservoir and well conditions differed significantly between the two periods, enabling a more robust evaluation of the hybrid model's performance, and iii) comprehensive PROSPER files (including PVT, IPR, and VLP curves) are available, accurately characterizing the reservoir and well conditions throughout the selected time frames.

To reduce the computational cost of the validation procedure, the field data collected during the two selected periods were downsampled to a rate of one measurement every two hours for all variables involved.

Fig. 14 presents the simulation results of the hybrid model of the network, which incorporate either an ANN-based pump model (green) or a RF-based pump model (orange), compared against the real field data (grey circles) collected during November 2021. Table 1 provides the corresponding numerical evaluation of these comparisons in terms of the NRMSE. The visual comparison for the water and the gas flow rates is shown in Fig. 27 in the Annex. Both Fig. 14 and Table 1 confirm the high prediction accuracy of the network's hybrid model, whether using the ANN or RF as a ML-based pump model. Moreover, the hybrid model demonstrates satisfactory performance in capturing the plant's transient behavior, particularly evident in Fig. 14 during the first ten days of November 2021.

It is important to emphasize that the prediction error values presented in Table 1 are significantly overestimated. This overestimation arises because the real field data used to compute the accuracy index

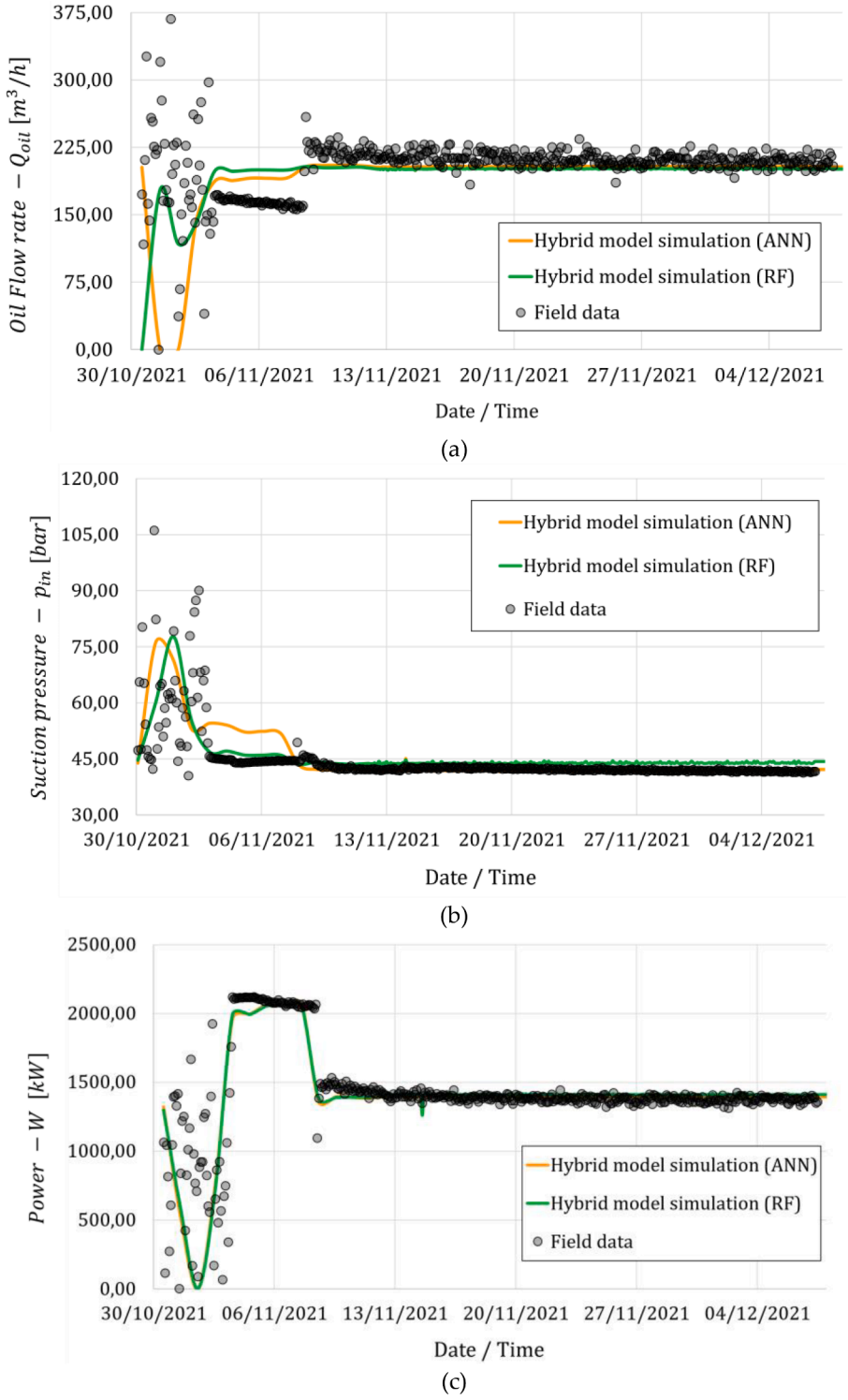


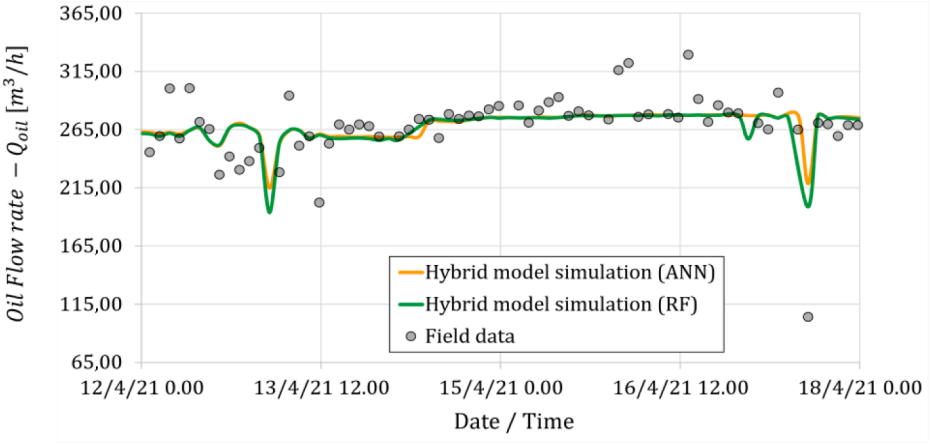
Fig. 14. Comparison between the real field data (grey circles) and the outcomes of the hybrid model of the network for the period of November 2021: (a) Oil flow rate Q_{oil}, (b) Suction pressure p_{in}, and (c) Power W.

Table 1
Prediction accuracy of the hybrid model over the period of November 2021, evaluated in terms of NRMSE (%).

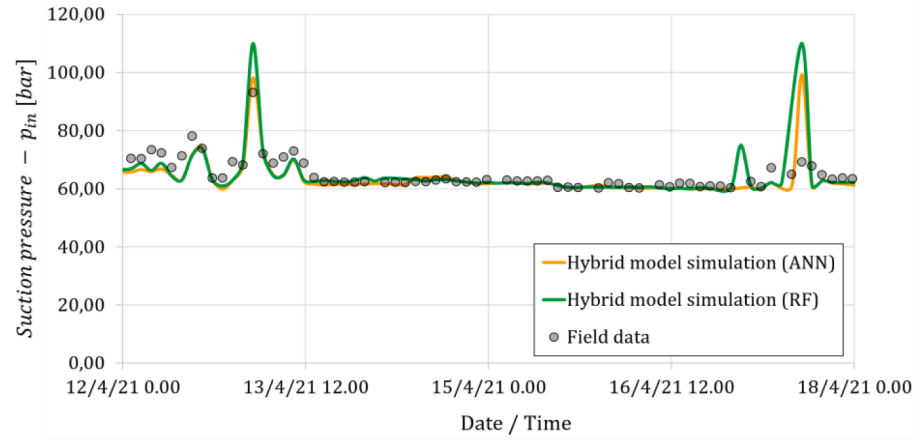
	Q_{oil}	Q_{water}	Q_{gas}	p_{in}	W
ANN	2.06 %	2.70 %	1.93 %	0.15 %	1.22 %
RF	2.90 %	2.43 %	1.97 %	0.62 %	1.55 %

contain substantial noise due to instrumentation faults and system instability - especially in the flowrates of oil Q_{oil}, water Q_{water} and gas Q_{gas}.

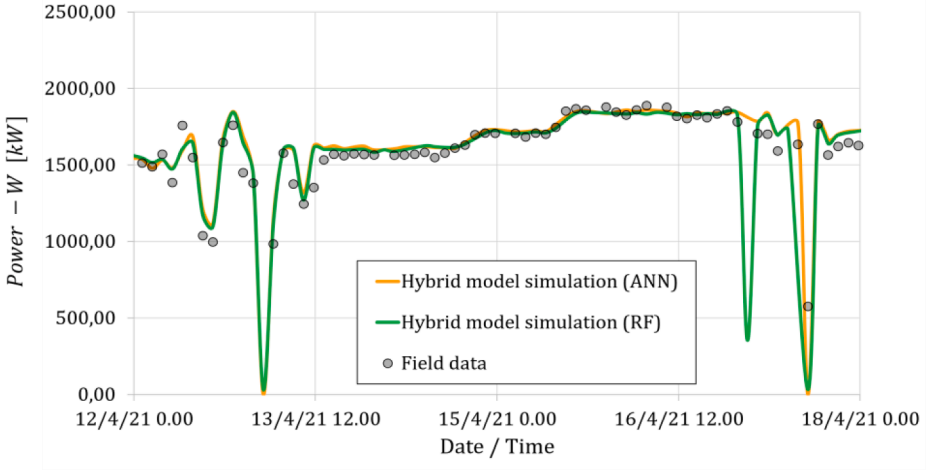
An additional validation of the network's hybrid model was performed using real field data collected in April 2021. It is important to note that this period lies outside the original dataset range described in Section 2.2, which spans from September 2021 to December 2022. Therefore, the ML-based pump models (both ANN and RF) were



(a)



(b)



(c)

Fig. 15. Comparison between the real field data (grey circles) and the simulation of the hybrid model of the network for the period of April 2021: (a) Oil flow rate Q_{oil}, (b) Suction pressure p_{in}, and (c) Power W.

Table 2
Prediction accuracy of the hybrid model over the period of April 2021, evaluated in terms of NRMSE (%).

	Q_{oil}	Q_{water}	Q_{gas}	p_{in}	W
ANN	3.43 %	5.02 %	4.43 %	1.00 %	4.56 %
RF	3.46 %	5.13 %	4.01 %	1.03 %	3.71 %

retrained using an updated dataset covering April 2021 to December 2022, following the exact procedure outlined in the methodology section.

The validation results of the network's hybrid model for April 2021 are presented in Figs. 15, 28 (in the Annex), and Table 2. These results further confirm the hybrid model's ability to replicate real field behavior, even under varying reservoir and well conditions. Similar to the results from November 2021, the prediction errors for oil, gas, and water flow rates are noticeably higher than those for power and suction pressure. This is due to the relatively higher levels of noise in the real

Table 3
Optimal values of the decision variables, the resulting objective function and the computational burden for the periods of November and April, using the network hybrid model with either the ANN-based or RF-based pump model.

Decision variables and objective function [unit]	Optimization for the period of November-2021		Optimization for the period of April-2021	
	RF	ANN	RF	ANN
δ_1 [in]	3.561	3.603	3.317	3.914
δ_2 [in]	0.569	0.501	⊥	⊥
δ_3 [in]	2.595	3.142	2.500	1.598
δ_4 [in]	2.627	3.437	0.558	0.515
δ_5 [in]	3.701	3.399	3.545	3.441
δ_6 [in]	0.706	1.091	0.562	0.661
δ_7 [in]	3.086	3.343	2.792	3.646
δ_8 [in]	3.085	3.915	3.780	3.052
ω [rpm]	3910.39	4407.93	4479.52	4793.83
Optimal objective, Q_{oil} [m^3/h]	234.51	240.05	333.74	334.63
Number of generations consumed before termination	11	14	9	12
Computational time [h]*	17	22	10	14

⊥ In the actual field operations, the choke opening of well 2 was closed during April 2011. To reflect this condition in the hybrid model, we set $\delta_2 = 0$. As a result, δ_2 was excluded from the set of decision variables in the optimization problem for that period.

* A machine with 16 GB of RAM and a 2.3 GHz processor.

measurements of the flowrates.

4.3. Steady-State production optimization

4.3.1. Optimization results

The optimal daily production management of the network is achieved by solving the optimization problem formulated in Eq. (10), which is based on a well-validated hybrid model and employs the DE algorithm to seek the optimal solution.

Initially, a series of random trial-and-error optimization runs were conducted to determine the key parameters of the DE algorithm. The mutation rate was set to 0.5, the recombination rate to 0.7 and the population size to 135 individuals. Such relatively small population size was selected due to the relatively high computational cost of simulating the hybrid model. Each simulation run (i.e. a functional evaluation of a single individual or candidate solution) takes approximately 29 to 41 s, depending on convergence speed and conditions, resulting in several hours of computation for a single optimization run. To offset the limited population size, a generous upper limit of 1000 generations was set as the stopping criterion, along with a strict objective function tolerance of 0.01.The bounds for the decision variables were determined based on the operator's experience and equipment capacity. Specifically, the valve openings for the eight wells are set within the range [0.5–4] inches, representing the minimum and maximum allowable valve settings. Similarly, the pump speeds are allowed to vary within the range [3000–4800] rpm, corresponding to the standard operating range of the pumps.

To assess the method in a robust manner, we conducted production optimization of the network over two distinct periods - April 2021 and November 2021- during which the network exhibited different conditions and states. Consequently, the PROSPER files (PVT, IPR and VLP curves) integrated into the hybrid model were updated for each period to capture the varying reservoir and well conditions. Additionally, in each of these periods, the optimization was repeated twice, corresponding to the use of the hybrid model integrating either ANNs or RF as the pump model.

Table 3 presents the optimization results, including the optimal values of the decision variables, the corresponding objective function values, and the computational burden for the periods of November and April. The results are obtained using the network's hybrid model with either the ANN-based or RF-based pump model. These results highlight that the network's hybrid model incorporating the ANN-based pump model yields better objective values, in both periods. However, this comes at the cost of higher computational demands compared to the RF-based pump model. This may be attributed to the fact that RF, as a non-parametric model, has limited extrapolation capabilities beyond its training domain. Consequently, when the optimizer explores combinations of decision variables that lead to new conditions outside the RF model's training range, the predictions can become unreliable (Li et al., 2018). In contrast, the ANN, as a type of parametric model, possesses stronger extrapolation capabilities. This enables it to better support the optimization search and may encourage the optimizer to explore broader regions of the search space. Finally, it is worth noting that, in both periods, the results consistently recommend that the valve openings for wells 2 and 6 (i.e., δ_2 and δ_6) should be kept at near their minimum limits.

Fig. 16 illustrates the evolution of the optimal objective value across generations for the November 2021 optimization run, using both the ANN-based and RF-based pump models. Similarly, Fig. 17 presents the same results for the April 2021 run. In all cases, the optimal objective value either stabilizes or changes only slightly in the final generations, indicating convergence of the optimization process.

Fig. 18 shows the box-plot of the objective function values across generations for the November 2021 optimization run. The middle line of each-box represents the population median for that generation, whereas the box edges correspond to the first and third quartiles. The horizontal whiskers indicate the minimum and maximum values (excluding outliers), and the scattered points denote the outliers. As observed, the increase in the population median over generations indicates good convergence and highlights how the recombination phase in the DE algorithm progressively generates better offspring.

To delve deeper into the obtained results, we present the evolution of relevant variables during the network optimization for the period of November 2021 and offer expert insights. Fig. 19 shows the progression of the suction pressure p_{in}, a variable that, although not classified as a decision variable, plays a crucial role in the overall functioning of the

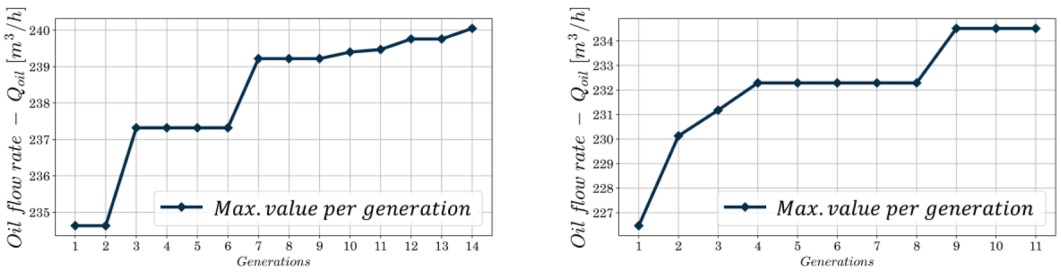


Fig. 16. Evolution of the optimal objective value Q_{oil} across generations for the optimization runs in April and November 2021: (a) ANN-based pump model, and (b) RF-based pump model.

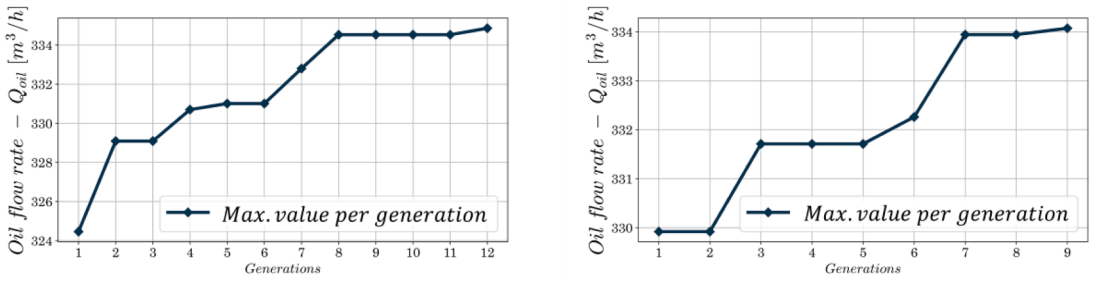


Fig. 17. Evolution of the optimal objective value Q_{oil} across generations for the optimization runs in April and November 2021: (a) ANN-based pump model, and (b) RF-based pump model.

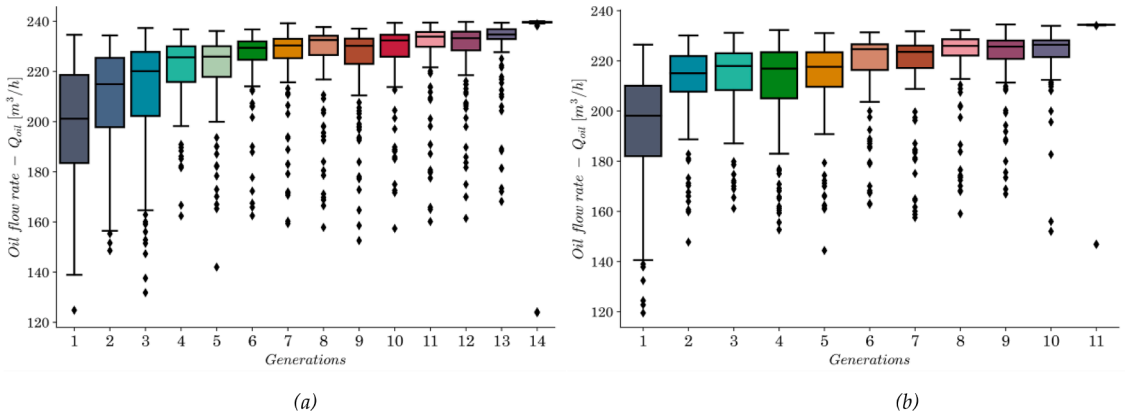


Fig. 18. Box plots showing the distribution of population objective values Q_{oil} across generations during the November 2021 optimization run: (a) ANN-based pump model, and (b) RF-based pump model.

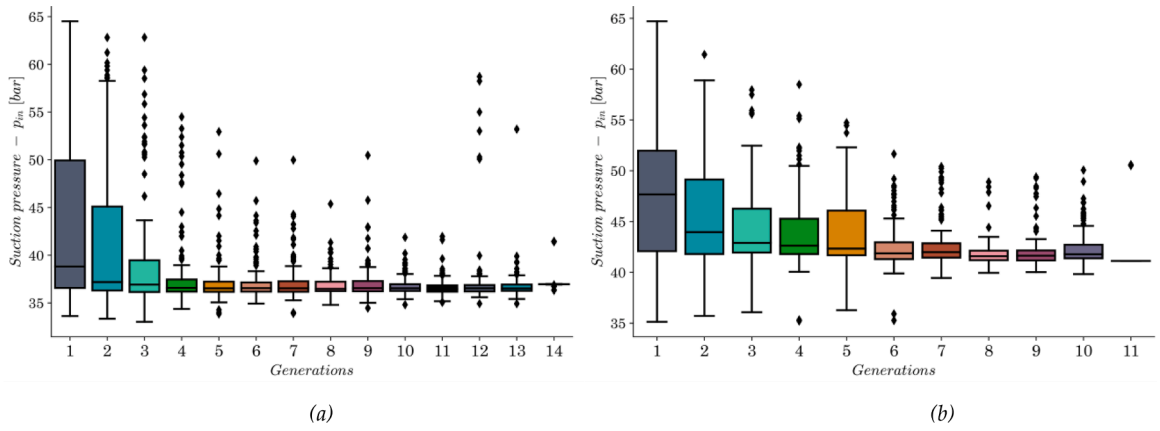


Fig. 19. Box plots showing the distribution of suction pressure p_{in}over the generations of optimization for the November 2021 run: (a) ANN-based pump model, and (b) RF-based pump model.

network. It can be observed that the suction pressure has a strong inverse correlation with the total oil flowrate extracted from the network (i.e., the objective function). Recall that the optimal production when using the ANN-based pump model (240.05 m^3/h) is slightly higher than when using the RF-based pump model (234.51 m^3/h), see Table 3. This is likely because the suction pressure in the latter case (Fig. 19-(a)) reaches lower values than in the former case (Fig. 19-(b)).

Fig. 20 shows that, for the November period, the optimization search is mainly focused on medium level pumps speed ω (see the median of the box-plots), whereas the final optimal speed never reaches to the maximum allowable limits. One could deduce that this is against the theoretical assumption, which is minimizing the suction pressure to extract more oil flowrate requires the operation of the pumps at the maximum allowable speed. However, as mentioned earlier in Sections 3.1 and 3.2, the pump does not operate freely, nor is it the only factor influencing the pressure distribution across the network. The opening of the check valves also plays a pivotal role, giving rise to a combined effect from both components.

Fig. 29 in the Annex presents similar results for the evolution of key variables (Q_{oil}, p_{in}, ω) throughout the generation during the optimization run for the period of April 2021.

4.3.2. *Comparing the optimization results to actual production state*

It is also insightful to compare the optimal production rates

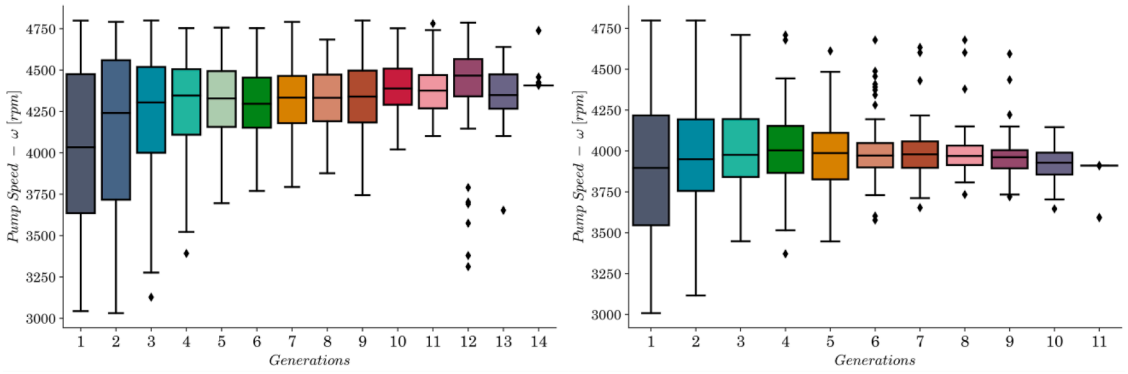


Fig. 20. Box plots showing the distribution of pump speed ω over the generations of optimization for the November 2021 run: (a) ANN-based pump model, and (b) RF-based pump model.

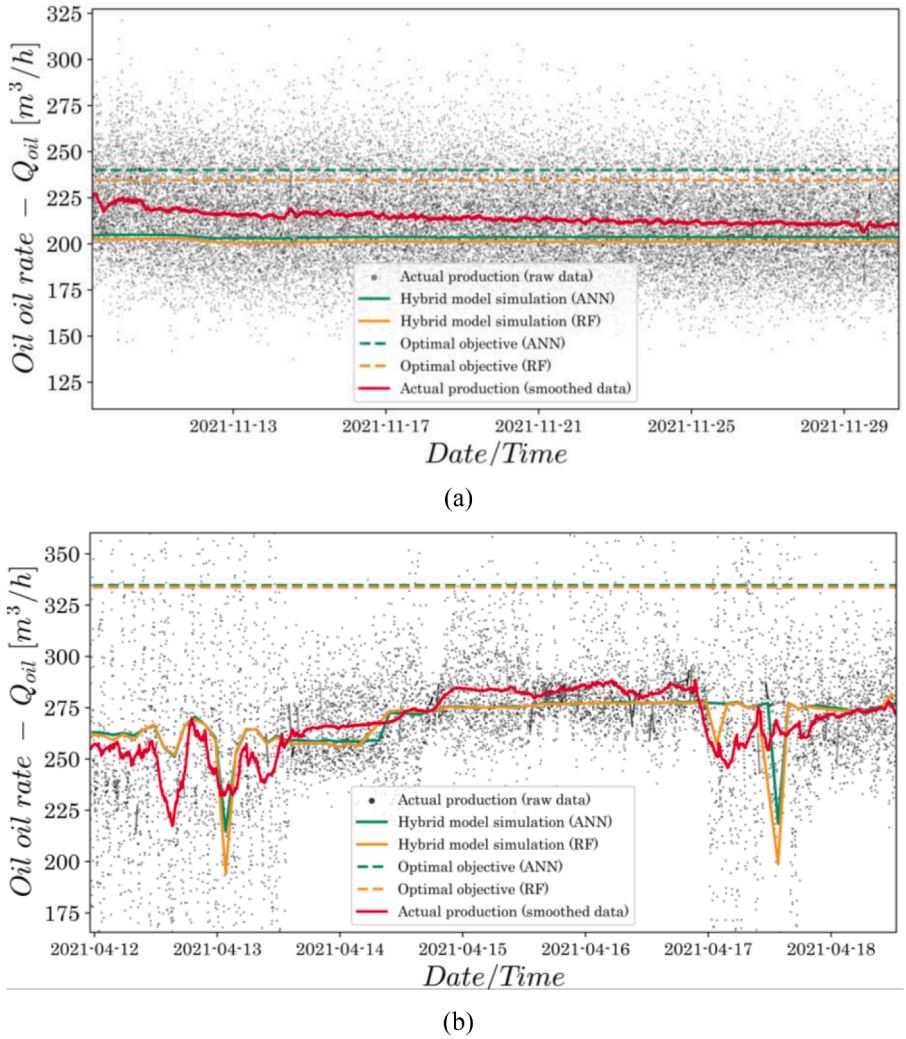


Fig. 21. Comparison of the optimal oil production rates obtained through optimization (green and orange dashed lines) with the actual field production rates (black dots), their smoothed values (red lines), and their simulated behavior (solid green and orange lines) during the periods of (a) November 2021 and (b) April 2021.

previously obtained through optimization (i.e., in Section 4.3.1) with the actual production rates of the network during April 2021 (Fig. 21-(a)) and November 2021 (Fig. 21-(b)). In this Figure: i) the black dots represent the raw oil flowrate data measured from the field, which contain a significant amount of noise due to flow instabilities, as well as measurement and metering limitations; ii) the solid red lines show the smoothed values of these raw measurements, obtained using the SG filter with a relatively large window of 8 hours, in order to generate a representative average for comparison; iii) the solid green and solid orange lines correspond to the simulation results of the hybrid model

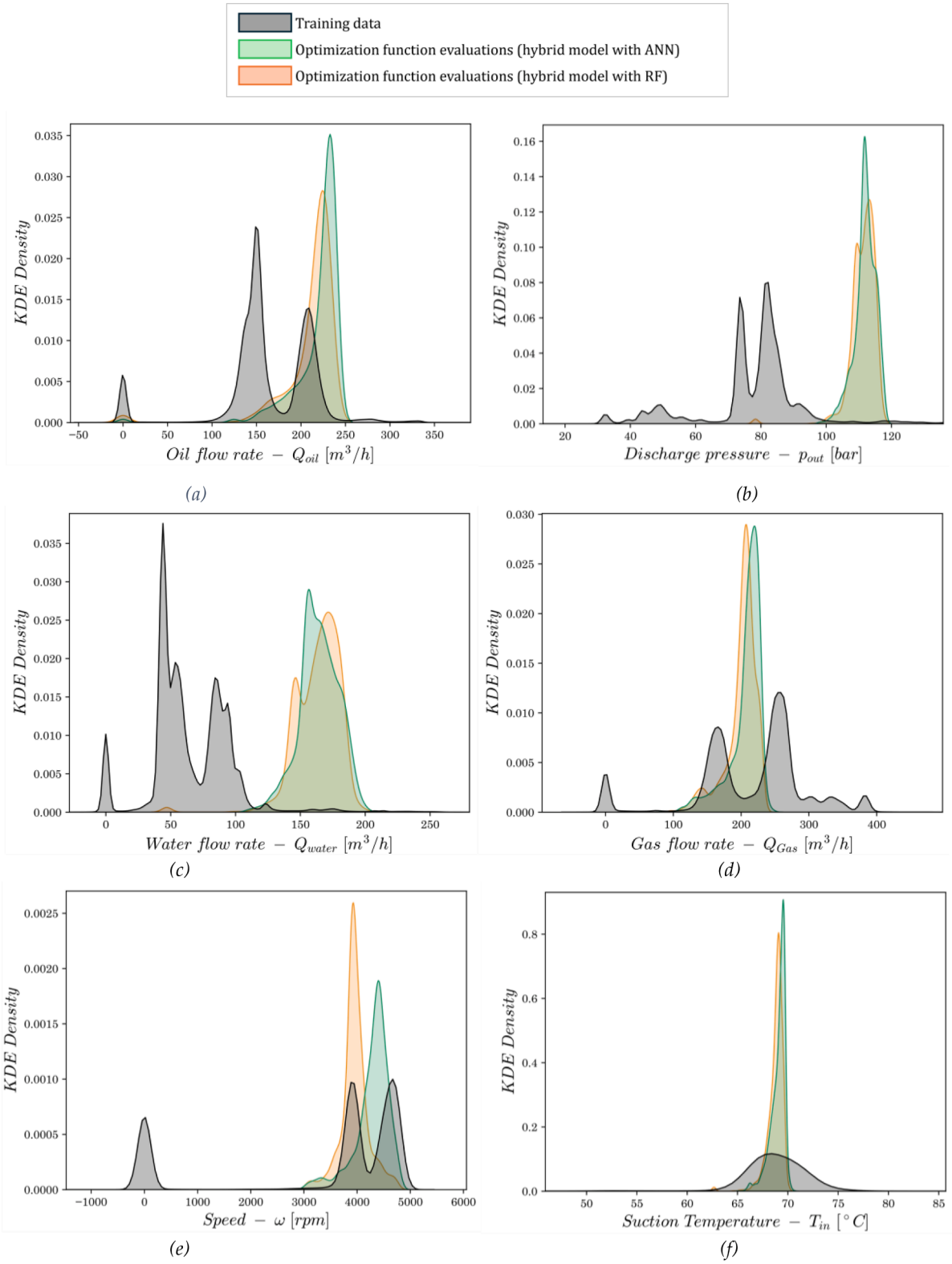
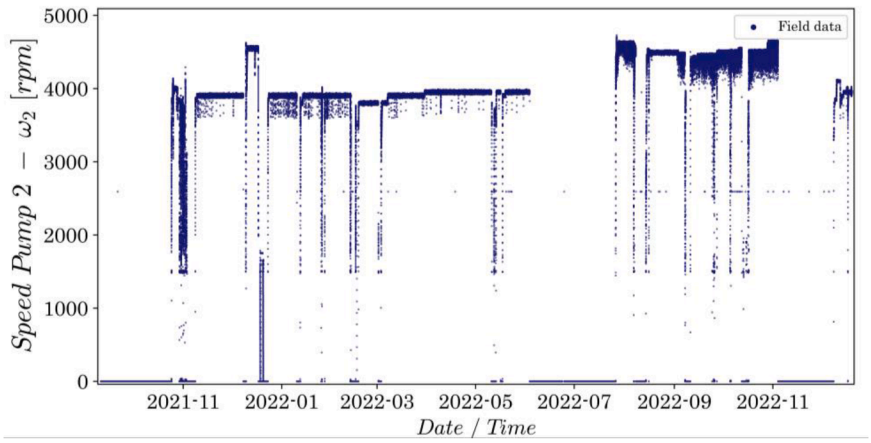


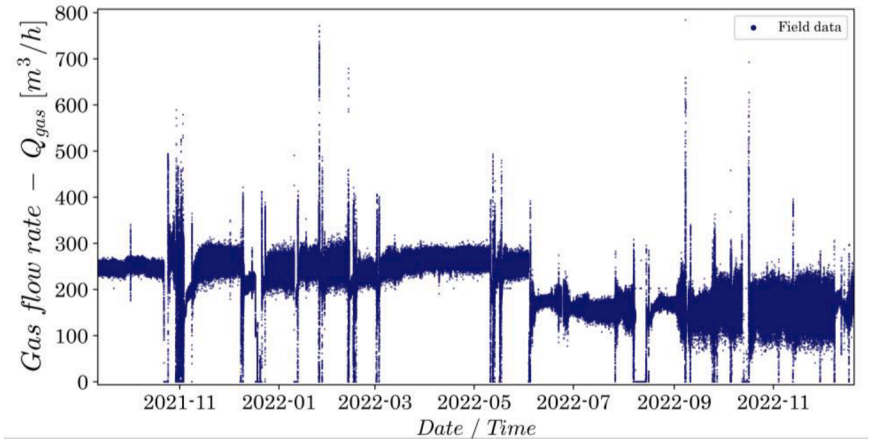
Fig. 22. Distribution of real field data (light grey) used to train the ML-based pump model, compared to the input regions explored during the optimization search in the period of November 2021. The explored regions are shown in light green for the ANN-based pump model and in light orange for the RF-based pump model. Subplots correspond to the following input variables: (a) Q_{oil}, (b) p_{out} (c) Q_{water}, (d) Q_{gas}, (e) ω, and (f) T_{in}.

incorporating the ANN-based and RF-based pump models, respectively; iv) the dashed green and dashed orange lines represent the optimal production rates determined through optimization for the ANN and RF models, respectively.

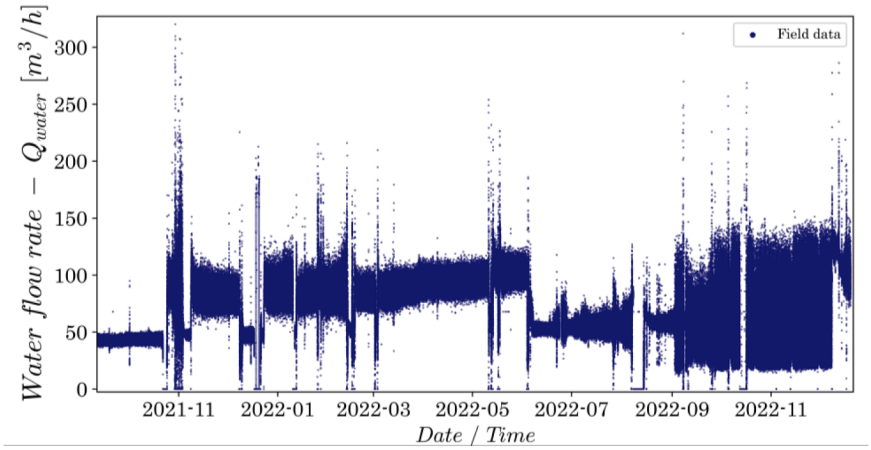
The Figure demonstrates that in the period of November 2021 the optimization resulted in an improvement in the production rate of 14.3 % (in the case of ANN) and 11 % (in the case of RF), whereas in the period of April 2021, the optimization led to a greater increase of 21.5 %



(a)

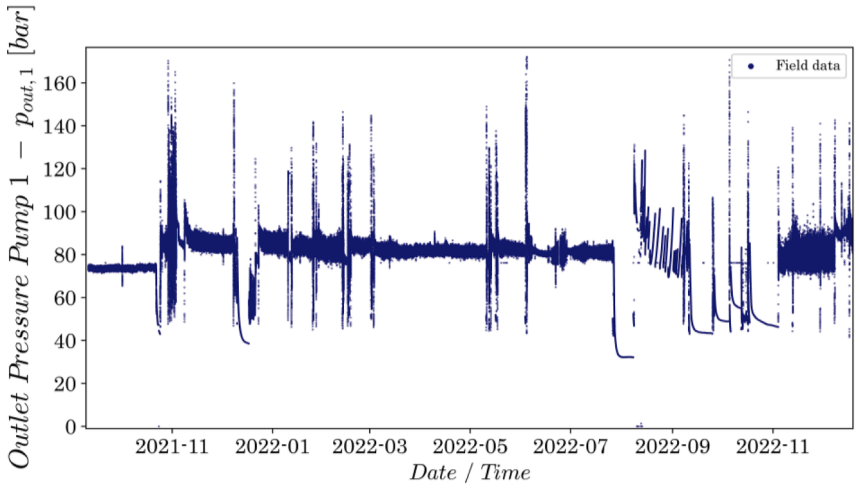


(b)

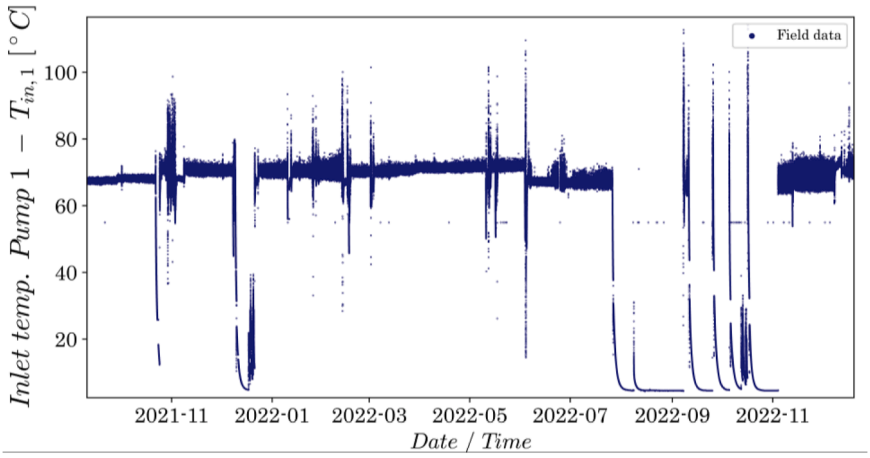


(c)

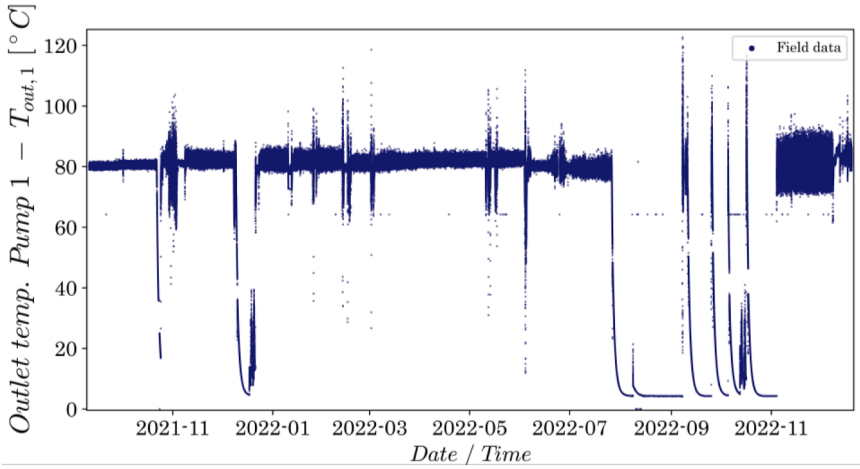
Fig. 23. Data collected from the real field: a) Speed of pump 2, ω_2, b) Gas flow rate, Q_{gas}, c) Water flow rate, Q_{water}, d) Outlet pressure of pump 1, $p_{out,1}$, e) Inlet temperature of pump 1, $T_{in,1}$, f) Outlet temperature of pump 1, $T_{out,1}$, and g) Power pump 1, W_1.



(d)

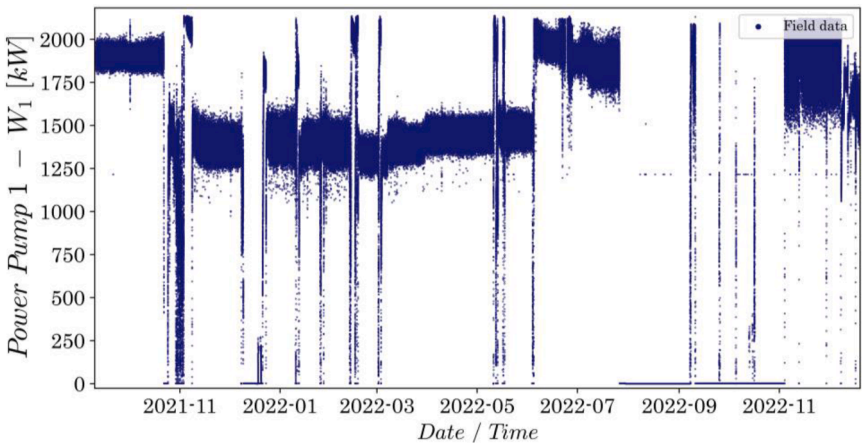


(e)



(f)

Fig. 23. (*continued*).



(g)

Fig. 23. (*continued*).

(in the case of ANN) and 21.1 % (in the case of RF).

4.3.3. Analysis of ML-Based pump model exploration during optimization

Before delving into this analysis, it is essential to recall the three following main facts of the modeling:

i) The ML-based pump model, $\mathscr{T}_{ML}$, takes as input the multiphase flow rates entering the pump train -Q_{Oil}, Q_{Water}, Q_{gas}- along with the pump speeds ω, discharge pressure p_{out} and inlet temperature T_{in} (see Eq.(5)). Since the ML-based pump model is developed using real field data, as described in Section 2.2, its input space corresponds to the variability range (i.e., the distribution) of that training data.
ii) The hybrid model $\mathscr{H}$ of the whole network embeds this trained ML-based pump model, $\mathscr{T}_{ML}$, along with physics-based models of other systems and components in the network, denoted as f^{Sub-A} and f^{Sub-B}, such that $\mathscr{H} = f^{Sub-A} \oplus \mathscr{T}_{ML} \oplus f^{Sub-B}$ (see Section 3.1 and Fig. 8). The input or control variables of the hybrid model, which also serve as the decision variables of the optimization problem, include the valve openings of the wells, δ_i, where $i = 1, = 1, \ldots, N_w = 8$, and the pumps speeds, ω.
iii) During the optimization search, the network's hybrid model $\mathscr{H}$ is used to simulate different combinations of the main control or decision variables explored by the ED optimizer.

By putting these facts together and referring to Fig. 8 in Section 3.1, it becomes clear that during the simulation of the network's hybrid model, the ML-based pump model receives inputs that are not direct optimization decisions—except for the pump speed. Instead, these inputs for the ML model are internal (or hidden) states computed by other physics-based components in the hybrid model, specifically f^{Sub-A} and f^{Sub-B}, as illustrated by the green arrows in Fig. 8. As a result, the inputs to the ML-based pump model during optimization are not fully controllable and instead depend on the specific combination of decision variables explored by the optimizer throughout the search process.

As a result, an insightful analysis is to compare the regions or spaces of the ML-based pump model's input explored during the optimization process performed in Section 4.3.1 with its nominal domain defined by the distribution of the real training data used to construct it (see Section 4.1). Given the large size of the training dataset (500,000 samples) compared to the relatively small number of function evaluations performed during optimization (a maximum of 2000), the Kernel Density Estimation (KDE) technique (Weglarczyk, 2018) is employed for the comparison. KDE is particularly suitable for this task as it normalizes all distributions such that their integrals equal one, enabling meaningful comparisons.

Fig. 22 displays the KDE distributions of the real field data used to train the ML-based pump model (shown in light grey), representing its nominal input domain. These are compared to the KDE distributions of the input space of the ML-based pump model explored during the optimization search, shown in light green for the ANN model and light orange for the RF model. Notice that for the input variables Q_{oil}, Q_{gas}, ω and T_{in} (Fig. 22-(a, d, e, f)), the optimization process led to the exploration of input spaces of the ML-based pump model (light green and orange) that are localized within the main bulk of the training data's nominal domain (light grey). In contrast, for the p_{out} and the Q_{water} variables (Fig. 22-(b, c)), the optimization search explores regions of the ML model's input space that, although still within the training data's nominal domain, correspond to areas with very low training data density. This could explain the improved optimization results achieved with the ANN-based pump model, as artificial neural networks generally exhibit stronger generalization and extrapolation capabilities compared to random forests.

5. Conclusions

This work addresses a practical challenge involving the efficient optimization of a real-world offshore oil and gas production network located in a central African country. Accurate physics-based models are available for most subsystems and units in the network (e.g., wells, gathering system, riser and the FPSO); however, they are not attainable for the MPPs, even though these units are crucial for network operation as they control the outlet quantity of produced oil. Despite various modeling trials employing different approaches, the developed models for the MPPs were unable to accurately describe the behavior of the real pumps in the field.

To tackle this challenge, the work leveraged real monitoring data collected from the pumps over a period of more than two years of operation. These data were used to develop machine learning models, based on ANN and RF, to approximate pump behavior. Sensitivity analysis was performed to adjust the hyperparameters of the ML models, such as the number of neurons for the ANN and the number of trees for the RF. The ML-based pump models were tested and shown to predict the real behavior of the pump with very good accuracy (a maximum NRMSE of 1.10 % for ANN and 0.7 % for RF). Subsequently, the ML-based pump models were integrated with the available physics-based models of the rest of the network to create the final hybrid model.

To assess its accuracy, the hybrid model was validated using real data collected from the field over two different periods of operation: April 2021 and November 2021. The validation results demonstrate that the

hybrid model can simulate the network's behavior with very good accuracy (with a maximum NRMSE of 4.43 % and 2.90 % for April 2021 and November 2021, respectively). Finally, network production optimization was conducted using a differential evolutionary algorithm and relying on the hybrid model. The optimization was also performed for the two aforementioned periods of April and November 2021. The optimal results were compared to the actual production of the field in both periods, revealing a significant enhancement in production by 12 % and 21 % for April 2021 and November 2021, respectively.

In this work, we relied on random cut-and-try optimization runs to estimate reasonable parameter values for the DE algorithm, such as population size, mutation rate and recombination rate. This approach was necessary due to the high computational burden of simulating the hybrid model, which prevented a comprehensive study to identify the optimal parameter settings. Although the optimization results appear promising and support the relevance of the observed convergence behavior, a key limitation of this study is the absence of a full sensitivity analysis and a thorough investigation of the DE algorithm's parameters.

As a first direction for future work, we aim to address the aforementioned limitation by reducing the complexity of the network's hybrid model through the use of surrogate models. This approach will enable a more detailed and comprehensive sensitivity analysis of the DE optimization algorithm's parameters at a significantly lower computational cost. The second direction will involve the development of an online control system for the oil and gas network, which requires constructing a dynamic model of the system and, consequently, a ML-based dynamic model of the pump capable of predicting its future behavior over multiple time steps under both steady-state and transient operating conditions. A third avenue for future research is the integration of an economic objective function for steady-state plant optimization, taking into account factors such as oil prices and operational costs, including the energy consumption associated with pump operation.

CRediT authorship contribution statement

Luca Trevisan: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ahmed Shokry:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Marco Montini:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Eric Moulines:** Writing – review & editing, Conceptualization. **Enrico Zio:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

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

Annex

Brief description of the ANN model

An ANN is a structure involving a certain number of layers: one input layer, one output layer and at least one hidden layer (Hagan, et al., 2014; NEGASH and YAW, 2020). Each layer is composed of a specific number of nonlinear processing units, called neurons. To process the information among each other's, the neurons are interconnected, where the importance of each connection is calibrated by a weight value. A generic n-th neuron in a generic ℓ-th hidden layer $\mathbb{L}_\ell$ is shown in Fig. 24.

The inputs to this neuron in Fig. 24 are the outputs of each neuron in the preceding layer $\mathbb{L}_{\ell-1}$, besides an independent bias $b_n^{\mathbb{L}_\ell}$ (orange arrows). Then, the output this generic neuron can be expressed as follows:

$$a_n^{\mathbb{L}_\ell} = f\left(b_n^{\mathbb{L}_\ell} + \Sigma\, a_m^{\mathbb{L}_{\ell-1}}\, \psi_{m^{\mathbb{L}_{\ell-1}},\, n^{\mathbb{L}_\ell}}\right),\ n = 1,\cdots,\mathcal{N}^{\mathbb{L}_\ell},\ m, = 1,\cdots,\mathcal{N}^{\mathbb{L}_{\ell-1}}$$

where $b_n^{\mathbb{L}_\ell}$ is the bias of n-th neuron in the layer $\mathbb{L}_\ell$, $a_m^{\mathbb{L}_{\ell-1}}$ is the output of the m-th neuron in the previous layer $\mathbb{L}_{\ell-1}$, $\psi_{m^{\mathbb{L}_{\ell-1}},\, n^{\mathbb{L}_\ell}}$ is the weight value

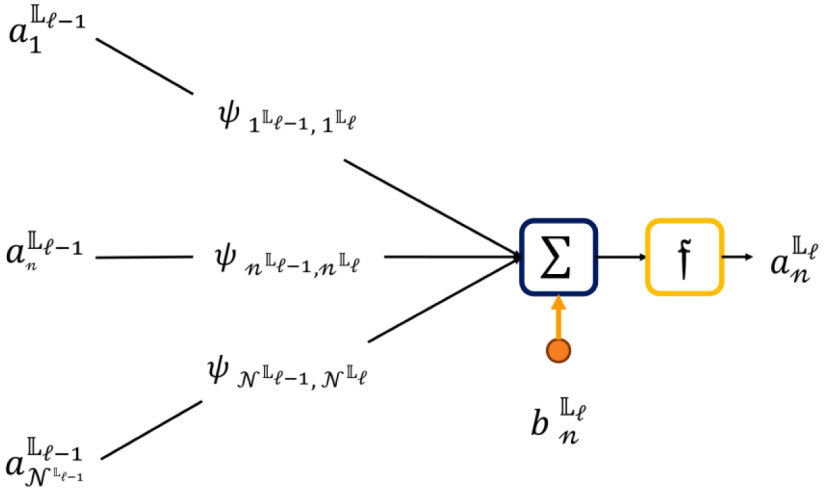


Fig. 24. Illustration of the generic n-th neuron located in the generic ℓ-th hidden layer $\mathbb{L}_\ell$.

assigned to the connection between these two neurons, $\mathcal{N}^{\mathbb{L}_\ell}$ is the number of neurons in layer $\mathbb{L}_\ell$, $\mathcal{N}^{\mathbb{L}_\ell-1}$ is the number of neurons in layer $\mathbb{L}_{\ell-1}$, and $\hat{\mathfrak{f}}$ is an activation function that can be either (Katz, et al., 2020):

- Linear transfer function, which acts as a mirror returning, as output, the same value of its input.
- Sigmoid transfer function, which maps the input to a value between 0 and 1. The fact that this function is smooth, continuous (differentiable), monotonic and bounded leads to increasing the efficiency of that back propagation.
- Hyperbolic tangent function (Tanh), which maps the input to a value between -1 and 1, and possess most properties of the Sigmoid type.
- Rectifying Linear Unit (ReLU) transfer function, which can be seen as a version of the linear transfer function. It is often used for its computational simplicity, because it has a sparse activation and better gradient propagation with respect to the sigmoid.

The linear transfer function is commonly used for the neurons of the output layer, while the transfer function of the neurons of the hidden layers can be Tanh, Sigmoid or ReLU. Fig. 25.

Training a neural network involves a nonlinear optimization problem in which optimal values of the weights and the biases are determined by minimizing a cost function, which is usually related to the errors between the outputs predicted by the network and their target values.

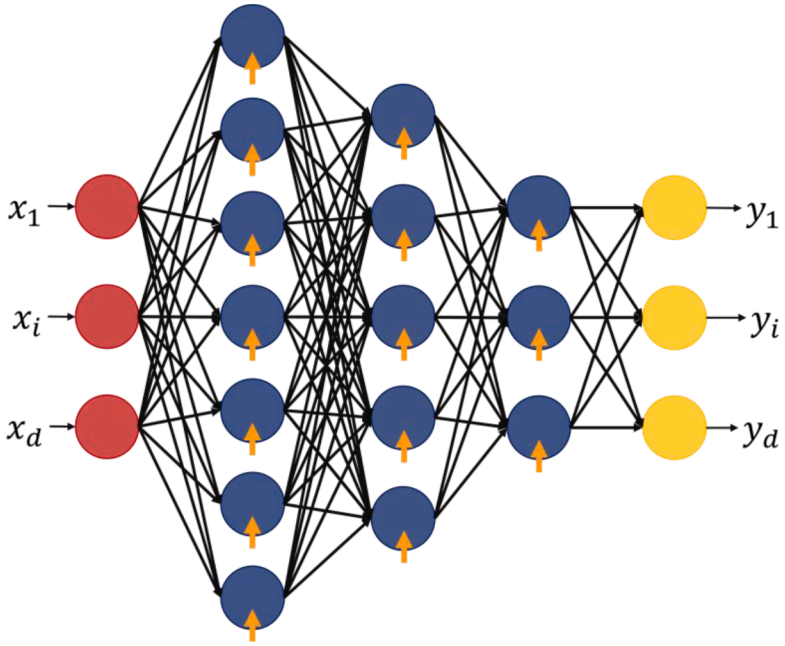


Fig. 25. A feed forward ANN with 3 hidden layers that maps the input $x \in R^d$ to the output $\in R^u$.

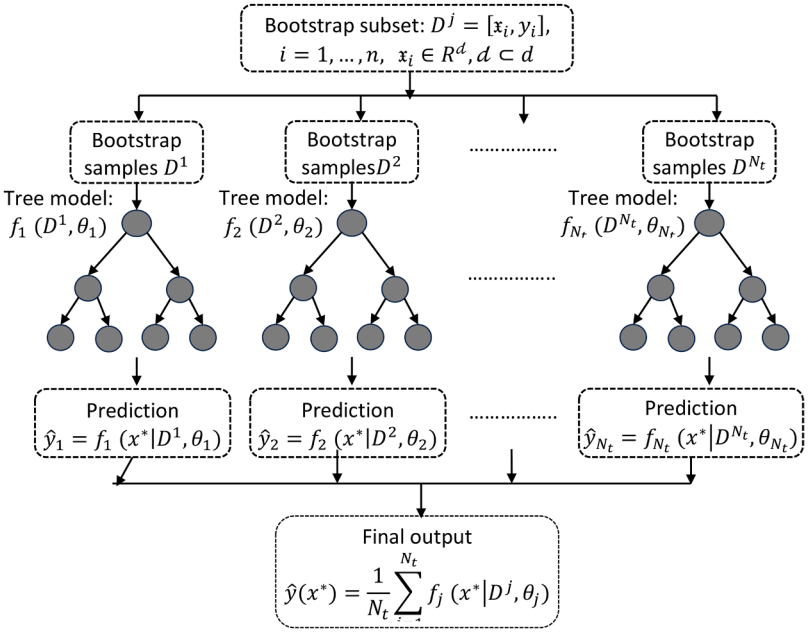


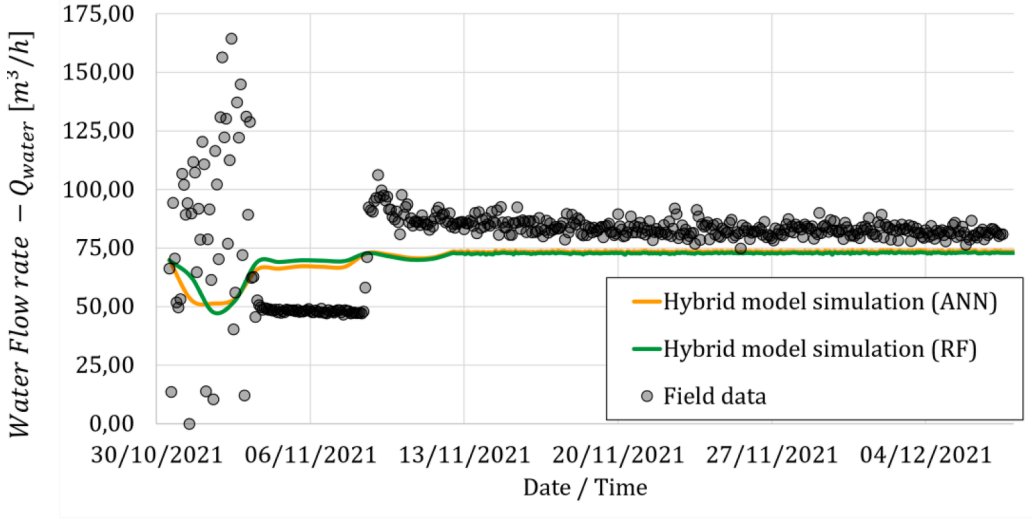
Fig. 26. Illustration of the random forest model for regression.

Defining the best structure (e.g., number of layers and involved neurons) and configuration (e.g., type of transfer function) of an ANN is a challenging task. However, general guidelines in the literature indicates that a neural network for a regression with two or three layers is sufficient enough to approximate a wide range of engineering problems (Hagan, et al., 2014). In this work, an ANN including two hidden layers is decided, with a rule of thumb that the number of neurons in the second hidden layer equals to half of that in the first hidden layer. The reason is to have a squeezing structure that gradually compresses and condenses the latent information from one layer to the next. Then a two-stage sensitivity analysis has been performed to select the best number of neurons and the type of transfer function in the hidden layers. To minimize the risk of over fitting early stopping procedure is considered. Thus, if the error on the validation set is not decreasing over certain 10 of successive training epochs, the training is stopped.

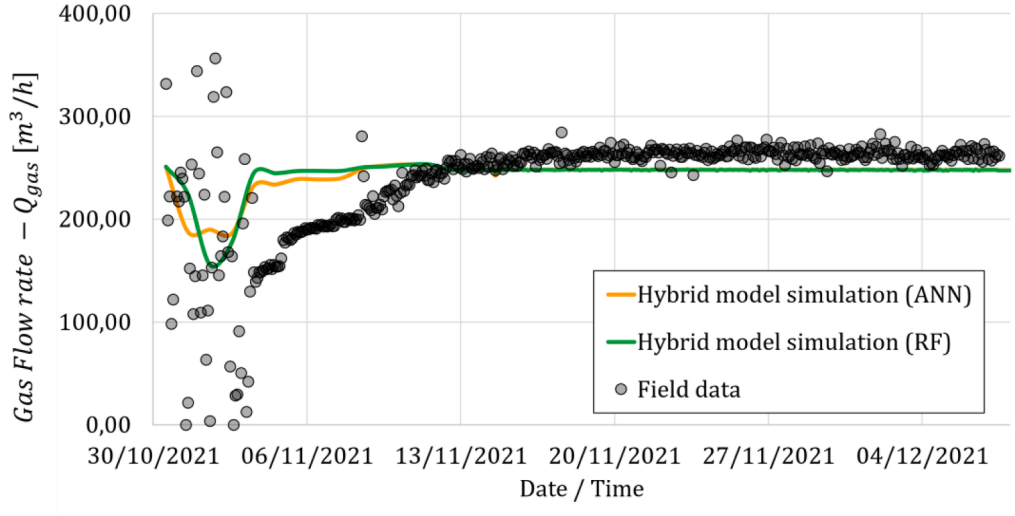
Brief description of the Random Forest model

The random forest (RF) model for regression (Fig. 26) is an ensemble of a number of N_t de-correlated decision tree (DT) models (Li et al., 2018). Given the original dataset D, $D = [x_i, y_i]$, $i = 1,\ldots,n$, where $x \in R^d$, $x \in R^1$, and each DT model, $f_j\ (D^j, \theta_j)$, $j = 1,\ldots N_t$ is build using a subset of the original dataset, D^j, that includes a randomly selected group of input features, such that $D^j = [\underline{x}_i, y_i], i = 1,\ldots,n$, $\underline{x}_i \in R^{d'}, d' \subset d$. A DT model for regression predicts the value of the target variable by learning a tree-like structure consisting of decision and leaf nodes. A decision node has two or more splits/branches of a certain input feature, while a leaf node represents the final output of those decisions and does not contain any further branches. The training of a DT recursively partitions/splits the input features at certain threshold values to create the decision nodes, while the goodness of a split (i.e., which feature is to be split, and at which threshold value) is evaluated considering the mean square error of target prediction (Li et al., 2018).

The final prediction of a trained RF model is calculated as the average of the predictions of the N_t DT models, such as $\hat{y}(x^*) = \frac{1}{N_t} \sum_{j=1}^{N_t} f_j\ (x^* | D^j,\ \theta_j)$, where x^* is a testing point and θ_j is the set of hyperparameters of the $j-th$ DR model. Notice that a key parameter for the RF model is the number of DT

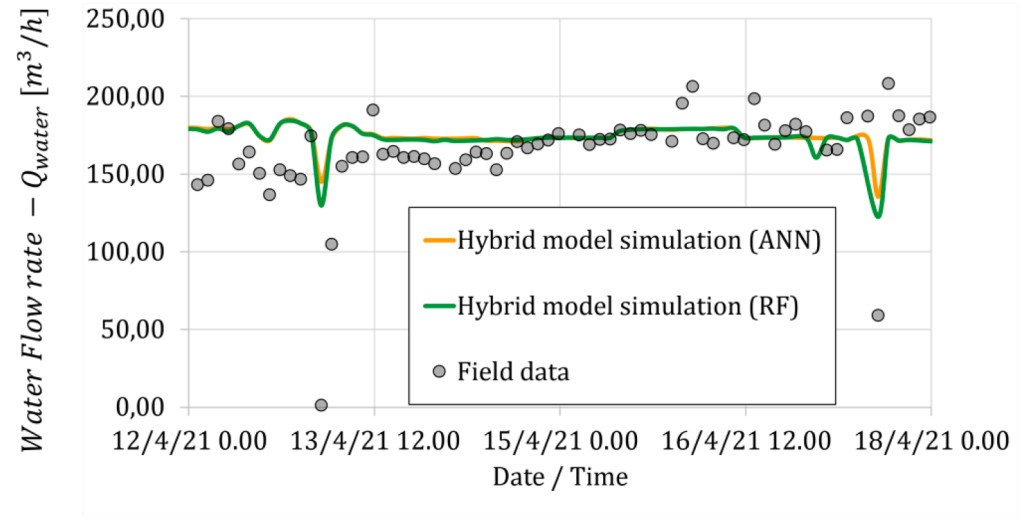


(a)

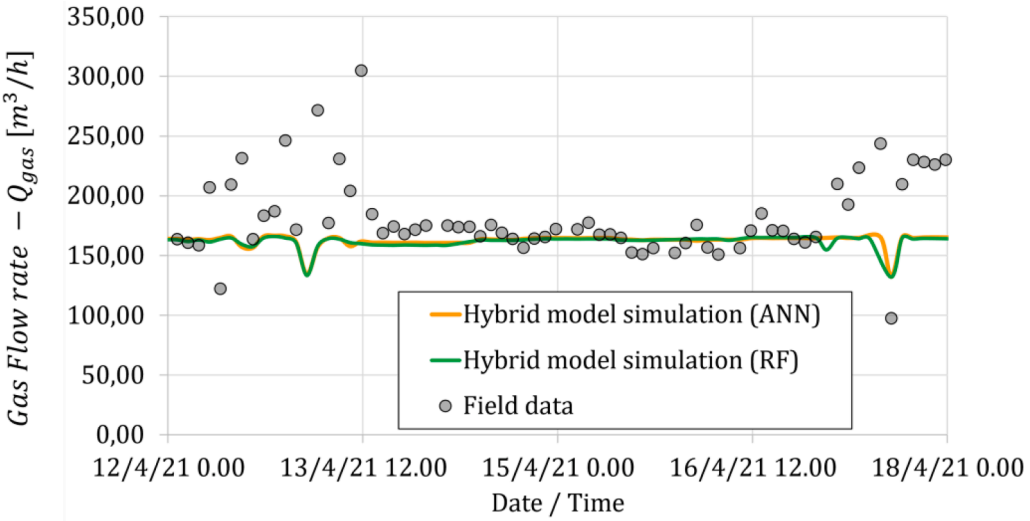


(b)

Fig. 27. Comparison between the real field data (grey circles) and the simulation of the hybrid model of the network for the period of November 2021: (a) Water flow rate Q_{water}, and (b) Gas flow rate Q_{gas}.



(a)



(b)

Fig. 28. Comparison between the real field data (grey circles) and the simulation of the hybrid model of the network for the period of April 2021: (a) Water flow rate, Q_{water}, and (b) Gas flow rate, Q_{gas}.

models, N_t. As this number increases, the prediction accuracy is likely to increase, however this is combined with a rise in the required computational cost. In this work, a sensitivity analysis us conducted to select the best the number of DT models (Svetnik, et al., 2003).

Brief description of DE algorithm

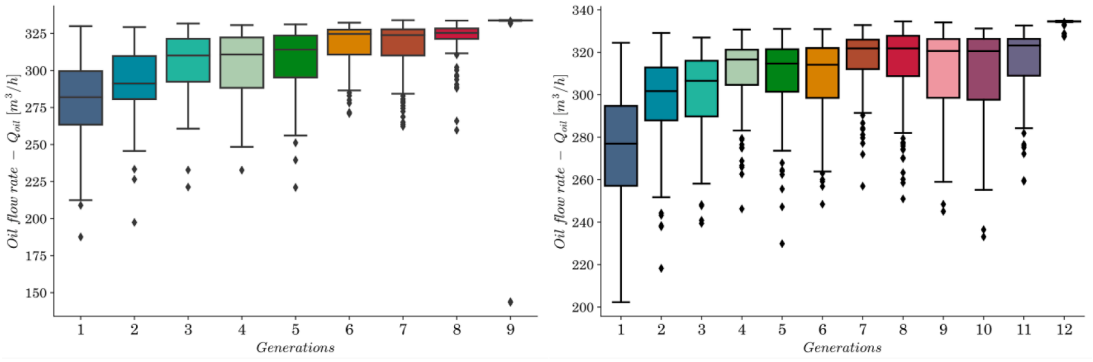
We consider the general single-objective optimization problem illustrated in Eq. (10) function $\mathcal{J}\left(\vec{X}^{\rightarrow}\right)$, where $\mathcal{J}: \Omega \subseteq \Re^{D} \rightarrow \Re$ that is $\mathcal{J}\left(\vec{X}^{\rightarrow^*}\right) < \mathcal{J}\left(\vec{X}^{\rightarrow}\right)$ for all $\vec{X}^{\rightarrow} \in \Omega$, with Ω being a non-empty set $\Omega = \Re^{D}$ (Aranha et al., 2021):

$$\min_{\forall \vec{X} \in \Omega} \mathcal{J}\left(\vec{X}\right) \tag{1a}$$

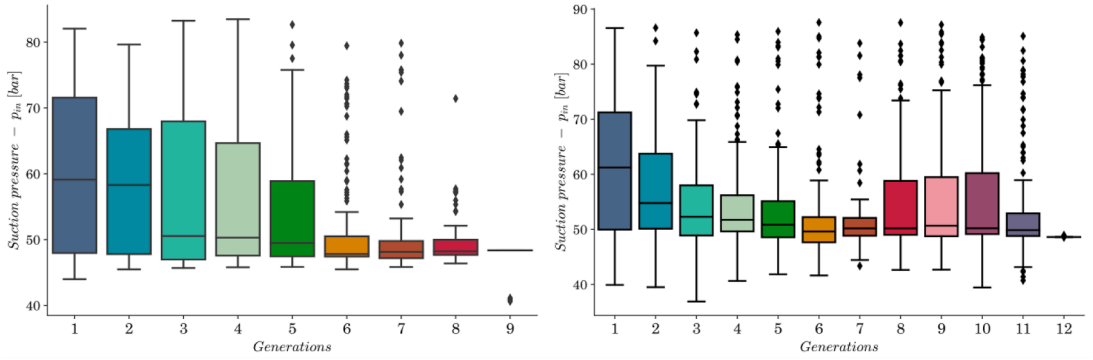
The DE algorithm seeks for the optimal solution $\vec{X}^{\rightarrow^*}$ through a maximum number of generations $\mathbb{G}$, where each generation includes a population of Np candidature solutions $\vec{X}_i^g$ with $i = 1, \ldots, Np$ and $g = 1, \ldots, \mathbb{G}$. The initial generation/population (i.e., $g = 1$) is randomly generated and then the evolution from a generation to the next is done through mutation, crossover and selection processes (Price et al., 2005).

i). Population initialization

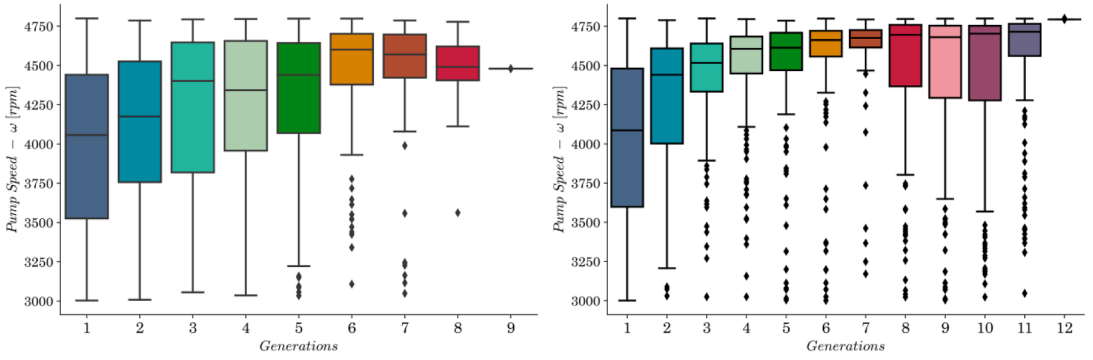
The initial generation, $\vec{X}_i^1 = \left[x_{i,1}^1, \ldots, x_{i,j}^1, \ldots, x_{i,D}^1\right]$, $i = 1, \ldots, Np$, $j = 1, \ldots, D$ is randomly generated considering to a uniform distribution in order to



(a)



(b)



(c)

Fig. 29. Box plots showing the evolution of (a) the objective function Q_{oil}, (b) suction pressure p_{in}, and (c) pump speed ω over the generations of the optimization run for April 2021. Results are shown for both the ANN-based pump model (left column) and the RF-based pump model (right column).

cover the search space of the variables, as follows:

$$x_{i,j}^{1} = x_{j,min} + rand_{i,j} \cdot (x_{j,max} - x_{j,min})$$

Where $x_{j,min}$ and $x_{j,min}$ are the minimum and maximum bounds for the j-th decision variables, $j = 1,\ldots,D$, while $rand_{i,j}$ is a random number uniformly distributed in the range $[0,1]$.

ii). Mutation

The mutation creates a mutant or a donor vector $\vec{V}_i^g$ for each target vector $\vec{X}_i^g$:

$$\vec{V}_i^g = \vec{X}_{r_{i,1}}^g + SF \cdot \left(\vec{X}_{r_{i,2},}^g - \vec{X}_{r_{i,3}}^g \right)$$

where $r_{i,1},\ r_{i,2}, r_{i,3} \in \{1,\ldots,NP\}$, $i \neq \{r_{i,1},\ r_{i,2}, r_{i,3}\}$ are randomly chosen and mutually exclusive integer indices, while SF is a scaling factor.

Then, each element (gene) of the doner vector $\overrightarrow{V}_i^g = \left[v_{i,1}^g, \ldots, v_{i,j}^g, \ldots, v_{i,D}^g\right]$ is scaled by applying a sigmoid function (see Eq.(11)) to ensure that the mutation operator is in the range $[0,1]$:

$$v_{i,j}^g = \frac{1}{1+e^{v_{i,j}^g}}, j = 1, \cdots D \tag{2a}$$

iii). Crossover

In the crossover step, the donor $\overrightarrow{V}_i^g$ and the target $\overrightarrow{X}_i^g$ vectors exchange their genes and generate the trail vector $\overrightarrow{U}_i^g$. In this work, the binomial crossover operator is considered for the exchange:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & if\ rand_{i,j} \le Cr\ or\ j = j_{rand} \\ x_{i,j}^g, & otherwise \end{cases} \tag{3a}$$

Where $rand_{i,j}$ is a random number uniformly generated for each of the $j-$ th components/genes of $i-$ th vector, $0< Cr < 1$ is the crossover rate that represents the probability that a gene of the donor vector will survive in the gene of the trail, and j_{rand} is a random integer uniformly sampled the set $[1, \ldots, D]$.

iv). Selection

the objective of this step is to choose the NP surviving vectors among the set of $2 \times NP$ vectors formed by the trail, $\overrightarrow{U}_i^g$, and the target, $\overrightarrow{X}_i^g$, $i = 1, \ldots, NP$ chromosomes. This is simply performed considering the values of the fitness function $\mathscr{J}$:

$$\overrightarrow{X}_i^{g+1} = \begin{cases} \overrightarrow{U}_i^g & if\ \mathscr{J}\left(\overrightarrow{U}_i^g\right) \le \mathscr{J}\left(\overrightarrow{X}_i^g\right) \\ \overrightarrow{X}_i^g & if\ \mathscr{J}\left(\overrightarrow{U}_i^g\right) \ge \mathscr{J}\left(\overrightarrow{X}_i^g\right) \end{cases} \tag{4a}$$

Data availability

The data that has been used is confidential.

References

Al Lawati, M. et al., 2021. AI for Production Forecasting and Optimization of Gas Wells: a Case Study on a Gas Field in Oman. s.l., s.n.

Al Selaiti, I. et al., 2020. Robust data driven well performance optimization assisted by machine learning techniques for natural flowing and gas-lift wells in Abu Dhabi., s. n., pp. Paper presented at the, Virtual, October 2020.

Andreasen, A., 2020. Applied process simulation-driven oil and gas separation plant optimization using surrogate modeling and evolutionary algorithms. Chem. Eng. 4, 11.

Andrianov, N., 2018. A Machine Learning Approach for Virtual Flow Metering and Forecasting. IFAC-PapersOnLine.

Aranha, C., Martín-Vide, C., Vega-Rodríguez, M.A., 2021. Theory and Practice of Natural Computing. Springer Cham, Switzerland.

Aspentech, 2024. Aspen HYSYS: maximize safety, sustainability and profits by optimizing the entire site in one environment using industry-validated simulation and time-saving workflows [Online] Available at: https://www.aspentech.com/en/products/engineering/aspen-hysys.

Biegler, L., 2010. Nonlinear programming: concepts, algorithms, and applications to chemical processes. MOS-SIAM Ser. Optim.

Bishnu, S.K., Alnouri, S.Y., Al-Mohannadi, D.M., 2023. Computational applications using data driven modeling in process Systems: a review. Digit. Chem. Eng. 8, 100111.

Brioschi, S., et al., 2017. Take on Challenges in Deep-Water Production Optimization: A Real Successful Application of an Innovative Integrated Modelling Tool. Ravenna, Italy, s.n.

Cadei, L., et al., 2020. Machine Learning Advanced Algorithm to Enhance Production Optimization: An ANN Proxy Modelling Approach. Dhahran, Kingdom of Saudi Arabia, s.n.

Camponogara, E., et al., 2018. Derivative-Free Optimization of Offshore Production Platforms Sharing a Subsea Gas Network. IFAC-PapersOnLine 51, 185–190.

Carpio, R.R., et al., 2021. Short-term oil production global optimization with operational constraints: a comparative study of nonlinear and piecewise linear formulations. J. Pet. Sci. Eng. 198, 0920–4105.

Codas, A., Camponogara, E., 2012. Mixed-integer linear optimization for optimal lift-gas allocation with well-separator routing. Eur. J. Oper. Res. 217, 222–231.

Codas, A., et al., 2012. Integrated production optimization of oil fields with pressure and routing constraints: the Urucu field. Comput. Chem. Eng. 46, 178–189.

Epelle, E.I., Gerogiorgis, D.I., 2019. Mixed-Integer Nonlinear Programming (MINLP) for production optimisation of naturally flowing and artificial lift wells with routing constraints. Chem. Eng. Res. Des. 152, 134–148.

Fadda, G., 2017. PhD thesis. Universit`a degli Studi Di Cagliari.

Fetanat, A.T.M., 2024. Evaluation of carbon capture technologies in the oil and gas industry using a socio-technical systems perspective-based decision support system under interval type-2 trapezoidal fuzzy set. Digit. Chem. Eng. 12, 100164.

Giorgio, S.S.V., et al., 2014. Risk analysis for uncertainties management of integrated productyion optimisation and field potential evaluation. In: International Petroleum Technology Conference. Kuala Lumpur, Malaysia.

Giorgio, V., Danilo, A., Marco, D., Almatasem, S.. Integrated production optimization and surface facilities management through advanced optimization techniques. https://doi.org/10.2118/156798-MS.

Gongbo, L., et al., 2023. Dynamic optimization method of subsea production scheme based on mixed mutation flower pollination algorithm. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4410427.

Grimstad, B., Foss, B., Heddle, R., Woodman, M., 2016. Global optimization of multiphase flow networks using spline surrogate models. Comput. Chem. Eng. 84, 237–254.

Hagan, M.T., Demuth, H.B., Beale, M.H. &. Jesus, O., 2014. Neural network design. 2nd ed. s.l.:M.T. Hagan.

Hoffmann, A., Stanko, M., 2017. Short-term model-based production optimization of a surface production network with electric submersible pumps using piecewise-linear functions. J. Pet. Sci. Eng. 158, 570–584.

Huang, Z., Chen, Y., 2013. An improved differential evolution algorithm based on adaptive parameter. J. Control Sci. Eng.

Hülse, E.O., Camponogara, E., 2017. Robust formulations for production optimization of satellite oil wells. Eng. Optim. 49, 846–863.

Hülse, E.O., et al., 2020. Introducing approximate well dynamics into production optimization for operations scheduling. Comput. Chem. Eng. 136, 106773.

IPM-Suite, 2024. GAP: MULTIPHASE NETWORK MODELLING AND OPTIMISATION [Online] Available at: https://www.petex.com/pe-engineering/ipm-suite/gap/.

IPM-Suite, 2024. PROSPER:MULTIPHASE WELL AND PIPELINE NODAL ANALYSIS [Online] Available at: https://www.petex.com/pe-engineering/ipm-suite/prosper/.

Jardim, R., Morgado-Dias, F., 2020. Savitzky–Golay filtering as image noise reduction with sharp color reset. Microprocess. Microsyst. 74, 103006.

Katz, J., Pappas, I., Avraamidou, S., Pistikopoulos, E.N., 2020. Integrating deep learning models and multiparametric programming. Comput. Chem. Eng. 136.

Koroteev, D., Tekic, Z., 2021. Artificial intelligence in oil and gas upstream: trends, challenges, and scenarios for the future. Energy AI 3, 2666–5468.

Krishnamoorthy, D., Foss, B., Skogestad, S., 2018. Steady-state real-time optimization using transient measurements. Comput. Chem. Eng. 115, 34–45.

Li, Y., et al., 2018. Random forest regression for online capacity estimation of lithium-ion batteries. Appl. Energy 232, 197–210.
Luguesi, C., Camponogara, E., Seman, L.O., González, J.T., Leithardt, V.R.Q., 2023. Derivative-free optimization with proxy models for oil production platforms sharing a subsea gas network, 11. IEEE Access, pp. 8950–8967. https://doi.org/10.1109/ACCESS.2023.3239421.
Luo, J., Ying, K., He, P., Bai, J., 2005. Properties of Savitzky–Golay digital differentiators. Digit. Signal Process. 15, 122–136.
Marchetti, A., Ferramosca, A., González, A., 2014. Steady-state target optimization designs for integrating real-time optimization and model predictive control. J. Process Control 24, 129–145.
Matias, J., Oliveira, J.P., Le Roux, G.A., Jäschke, J., 2022. Steady-state real-time optimization using transient measurements on an experimental rig. J. Process Control 115, 181–196.
Mendoza, J.H., et al., 2021. Soft computing tools for multiobjective optimization of offshore crude oil and gas separation plant for the best operational condition. In: International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). Mexico City, Mexico.
Mohammadzaheri, M., et al., 2016. An intelligent approach to optimize multiphase subsea oil fields lifted by electrical submersible pumps. J. Comput. Sci. 15, 50–59.
Negash, B.M., Yaw, A.D., 2020. Artificial neural network based production forecasting for a hydrocarbon reservoir under water injection. Pet. Explor. Dev. 47, 383–392.
Petroleum Experts-PE, 2024. Integrated Production Modelling software (IPM) [Online] Available at: https://www.petex.com/products/ipm-suite/ [Accessed 2024].
Price, K., Storn, R.M., Lampinen, J.A., 2005. Differential evolution: a practical approach to global optimization. Natural Computing Series (NCS). Springer, Berlin, Heidelberg.
Ray, T., Sarker, R., 2007. Genetic algorithm for solving a gas lift optimization problem. J. Pet. Sci. Eng. 59, 84–96.
Scaramellini, S., Cerri, P., Bianco, A., MAsi, S., 2015. Short-Term Production Operation Management: Continuos Application of an Innovative Production Optimization Tool. Houston, Texas, USA, s.n.
Schafer, R.W., 2011. What Is a Savitzky-Golay Filter? [Lecture Notes]. IEEE Signal Process. Mag. 28, 111–117.
Selvan, K.K., Mundra, M., Panda, R., 2022. Steady-state and transient dynamics for sweetening of LPG process. Digit. Chem. Eng. 4, 100035.
Silva, T.L., Camponogara, E., 2014. A computational analysis of multidimensional piecewise-linear models with applications to oil production optimization. Eur. J. Oper. Res. 232, 630–642.
Slb, P.S., 2024. Pipesim Flow Modeling:the foundation for steady-state multiphase flow analysis [Online] Available at: https://www.software.slb.com/products/pipesim.
Stanko, M., 2020. Petroleum Production Systems: Compendium. NTNU, Trondheim, Norway.
Stanko, M., Golan, M., 2015. Exploring the Potential of Model-Based Optimization in Oil Production Gathering Networks with ESP-Produced High Water Cut Wells. Department of Petroleum Engineering, NTNU.
Svetnik, V., et al., 2003. Random forest: a classification and regression tool for compound classification and QSAR modeling. J. Chem. Inf. Comput. Sci. 43, 1947–1958.
Wan, X., et al., 2023. Optimization of operational strategies for rich gas enhanced oil recovery based on a pilot test in the Bakken tight oil reservoir. Pet. Sci. 20, 2921–2938.
Weglarczyk, S., 2018. Kernel density estimation and its application. ITM Web of Conferences. XLVIII Semin. Appl. Math. 23.
Yin, X., et al., 2022. A machine learning-based surrogate model for the rapid control of piping flow: application to a natural gas flowmeter calibration system. J. Nat. Gas Sci. Eng. 98, 104384.