EXPLORING THE ADOPTION OF A CONCEPTUAL DATA ANALYTICS FRAMEWORK FOR SUBSURFACE ENERGY PRODUCTION SYSTEMS

A Study of Predictive Maintenance, Multi-Phase Flow Estimation, and Production Optimization

Dissertation

Zur Erlangung des Doktorgrades

der Ingenieurwissenschaften

vorgelegt von

Ramez Maher Abdalla, M.Sc.

aus Alexandria, Ägypten

genehmigt von der

Fakultät für Energie- und Wirtschaftswissenschaften

der Technischen Universität Clausthal

Tag der mündlichen Prüfung:
11.09.2023

Dekan
Prof. Dr. mont. Leonhard Ganzer

Vorsitzender der Promotionskommission
Prof. Dr. mont. Dr. rer. nat. Michael Fischlschweiger

Betreuer
Prof. Dr. Ing. habil. Philip Jaeger

Gutachterin
Prof. Dr. Dan Sui

Eidesstattliche Erklärungen

Hiermit erkläre ich an Eides Statt, dass ich die bei der Fakultät für Energie- und Wirtschaftswissenschaften der Technischen Universität Clausthal eingereichte Dissertation selbständig und ohne unerlaubte Hilfe angefertigt habe. Die benutzten Hilfsmittel sind vollständig angegeben.

Unterschrift

01.06.2023

Datum

Acknowledgments

I would like to extend my heartfelt gratitude to my committee members who provided their invaluable expertise and time to make this thesis possible. A special mention goes to Prof. Dr. Philip Jaeger, my thesis main supervisor, for his unwavering support, encouragement, and patience throughout the entire process. His dedication and countless hours of reflection and reading helped me refine my ideas and produce a quality thesis.

I am deeply grateful to Dipl. Ing. Michael Stundner, Executive Vice President of Technology at Datagration, for his invaluable support and guidance throughout my research journey. His unwavering commitment and enthusiasm for my work has been instrumental in enabling me to achieve the best possible results. I would also like to express my appreciation to Datagration. Their dedication to innovation and excellence has been an inspiration to me, and I am honored to have been a part of it. Without the unwavering support of Dipl. Ing. Michael Stundner and Datagration, this research would not have been possible.

I am also grateful for the collaborative efforts of my colleagues. Their contributions helped me identify areas of improvement and led to a more comprehensive and robust thesis.

I would also like to acknowledge the study participants whose cooperation and contributions have been crucial in the gathering and analysis of data. Their willingness to participate in the study has been vital to the success of this research.

Overall, I hope that the findings of this thesis will provide valuable insights and contribute to the advancement of knowledge in this field.

Dedication

I dedicate my dissertation work to my family, who have provided me with love, encouragement, and unwavering support. Their sacrifices and guidance have been crucial to my academic success, and I am deeply grateful for everything they have done for me.

Abstract

As technology continues to advance and become more integrated in the oil and gas industry, a vast amount of data is now prevalent across various scientific disciplines, providing new opportunities to gain insightful and actionable information. The convergence of digital transformation with the physics of fluid flow through porous media and pipelines has driven the advancement and application of machine learning (ML) techniques to extract further value from this data. As a result, digital transformation and its associated machine learning applications have become a new area of scientific investigation.

The transformation of brownfields into digital oilfields can aid in energy production by accomplishing various objectives, including increased operational efficiency, production optimization, collaboration, data integration, decision support, and workflow automation. This work aims to present a framework of these applications, specifically through the implementation of virtual sensing, predictive analytics using predictive maintenance on production hydraulic systems (with a focus on electrical submersible pumps), and prescriptive analytics for production optimization in steam and waterflooding projects.

In terms of virtual sensing, the accurate estimation of multi-phase flow rates is crucial for monitoring and improving production processes. This study presents a data-driven approach for calculating multi-phase flow rates using sensor measurements located in electrical submersible pumped wells. An exhaustive exploratory data analysis is conducted, including a univariate study of the target outputs (liquid rate and water cut), a multivariate study of the relationships between inputs and outputs, and data grouping based on principal component projections and clustering algorithms. Feature prioritization experiments are performed to identify the most influential parameters in the prediction of flowing rates. Model comparison is done using the mean absolute error, mean squared error, and coefficient of determination. The results indicate that the CNN-LSTM network architecture is particularly effective in time series analysis for ESP sensor data, as the 1D-CNN layers are capable of extracting features and generating informative representations of time series data automatically.

Subsequently, the study presented herein a methodology for implementing predictive maintenance on artificial lift systems, specifically regarding the maintenance of Electrical Submersible Pumps (ESPs). Conventional maintenance practices for ESPs require extensive resources and manpower, and are often initiated through reactive monitoring of multivariate sensor data. To address this issue, the study employs the use of principal component analysis (PCA) and extreme gradient boosting trees (XGBoost) to analyze real-time sensor data and predict potential failures in ESPs. PCA is utilized as an unsupervised technique and its output is further processed by the XGBoost model for prediction of system status. The resulting predictive model has shown to provide signals of potential failures up to seven days in advance, with an F1-score greater than 0.71 on the test set.

In addition to the data-driven modeling approach, The present study also incorporates model-free reinforcement learning (RL) algorithms to aid in decision-making in production optimization. The task of determining the optimal injection strategy poses challenges due to the complexity of the underlying dynamics, including nonlinear formulation, temporal variations, and reservoir heterogeneity. To tackle these challenges, the problem was reformulated as a Markov decision process and RL algorithms were employed to determine actions that maximize production yield.

The results of the study demonstrate that the RL agent was able to significantly enhance the net present value (NPV) by continuously interacting with the environment and iteratively refining the dynamic process through multiple episodes. This showcases the potential for RL algorithms to provide effective and efficient solutions for complex optimization problems in the production domain.

In conclusion, this study represents an original contribution to the field of data-driven applications in subsurface energy systems. It proposes a data-driven method for determining multi-phase flow rates in electrical submersible pumped (ESP) wells utilizing sensor measurements. The methodology includes conducting exploratory data analysis, conducting experiments to prioritize features, and evaluating models based on mean absolute error, mean squared error, and coefficient of determination. The findings indicate that a convolutional neural network-long short-term memory (CNN-LSTM) network is an effective approach for time series analysis in ESPs. In addition, the study implements principal component analysis (PCA) and extreme gradient boosting trees (XGBoost) to perform predictive maintenance on ESPs and anticipate potential failures up to a seven-day horizon. Furthermore, the study applies model-free reinforcement learning (RL) algorithms to aid decision-making in production optimization and enhance net present value (NPV).

Kurzfassung

Als die Technologie weiter fortschreitet und immer stärker in der Öl- und Gasindustrie integriert wird, steht eine enorme Menge an Daten in verschiedenen Wissenschaftsdisziplinen zur Verfügung, die neue Möglichkeiten bieten, informationsreiche und handlungsorientierte Informationen zu gewinnen. Die Konvergenz der digitalen Transformation mit der Physik des Flüssigkeitsflusses durch poröse Medien und Pipeline hat die Entwicklung und Anwendung von maschinellem Lernen (ML) vorangetrieben, um weiteren Mehrwert aus diesen Daten zu gewinnen. Als Folge hat sich die digitale Transformation und ihre zugehörigen maschinellen Lernanwendungen zu einem neuen Forschungsgebiet entwickelt.

Die Transformation von Brownfields in digitale Ölfelder kann bei der Energieproduktion helfen, indem verschiedene Ziele erreicht werden, einschließlich erhöhter betrieblicher Effizienz, Produktionsoptimierung, Zusammenarbeit, Datenintegration, Entscheidungsunterstützung und Workflow-Automatisierung. Diese Arbeit zielt darauf ab, ein Rahmenwerk für diese Anwendungen zu präsentieren, insbesondere durch die Implementierung virtueller Sensoren, Vorhersageanalytik mithilfe von Vorhersagewartung für die Produktionshydraulik-Systeme (mit dem Schwerpunkt auf elektrischen Unterwasserpumpen) und präskriptiven Analytik für die Produktionsoptimierung in Dampf- und Wasserflutprojekten.

In Bezug auf virtuelle Messungen ist eine genaue Schätzung von Mehrphasenströmen für die Überwachung und Verbesserung von Produktionsprozessen entscheidend. Diese Studie präsentiert einen datengetriebenen Ansatz zur Berechnung von Mehrphasenströmen mithilfe von Sensormessungen in elektrischen untergetauchten Pumpbrunnen. Es wird eine ausführliche exploratorische Datenanalyse durchgeführt, einschließlich einer Ein Variablen Studie der Zielausgänge (Flüssigkeitsrate und Wasseranteil), einer Mehrvariablen-Studie der Beziehungen zwischen Eingaben und Ausgaben sowie einer Datengruppierung basierend auf Hauptkomponentenprojektionen und Clusteralgorithmen. Feature Priorisierungsexperimente werden durchgeführt, um die einflussreichsten Parameter in der Vorhersage von Fließraten zu identifizieren. Die Modellvergleich erfolgt anhand des mittleren absoluten Fehlers, des mittleren quadratischen Fehlers und des

Bestimmtheitskoeffizienten. Die Ergebnisse zeigen, dass die CNN-LSTM-Netzwerkarchitektur besonders effektiv bei der Zeitreihenanalyse von ESP-Sensordaten ist, da die 1D-CNN-Schichten automatisch Merkmale extrahieren und informative Darstellungen von Zeitreihendaten erzeugen können.

Anschließend wird in dieser Studie eine Methodik zur Umsetzung von Vorhersagewartungen für künstliche Hebesysteme, insbesondere bei der Wartung von Elektrischen Untergetauchten Pumpen (ESP), vorgestellt. Conventional maintenance practices for ESPs require extensive resources and manpower, and are often initiated through reactive monitoring of multivariate sensor data. Um dieses Problem zu lösen, wird die Verwendung von Hauptkomponentenanalyse (PCA) und Extreme Gradient Boosting Trees (XGBoost) zur Analyse von Echtzeitsensordaten und Vorhersage möglicher Ausfälle in ESPs eingesetzt. PCA wird als unsupervised technique eingesetzt und sein Ausgang wird weiter vom XGBoost-Modell für die Vorhersage des Systemstatus verarbeitet. Das resultierende Vorhersagemodell hat gezeigt, dass es Signale von möglichen Ausfällen bis zu sieben Tagen im Voraus bereitstellen kann, mit einer F1-Bewertung größer als 0,71 im Testset.

Diese Studie integriert auch Model-Free Reinforcement Learning (RL) Algorithmen zur Unterstützung bei Entscheidungen im Rahmen der Produktionsoptimierung. Die Aufgabe, die optimalen Injektionsstrategien zu bestimmen, stellt Herausforderungen aufgrund der Komplexität der zugrundeliegenden Dynamik, einschließlich nichtlinearer Formulierung, zeitlicher Variationen und Reservoirstrukturheterogenität. Um diese Herausforderungen zu bewältigen, wurde das Problem als Markov-Entscheidungsprozess reformuliert und RL-Algorithmen wurden eingesetzt, um Handlungen zu bestimmen, die die Produktion optimieren. Die Ergebnisse zeigen, dass der RL-Agent in der Lage war, den Netto-Barwert (NPV) durch kontinuierliche Interaktion mit der Umgebung und iterative Verfeinerung des dynamischen Prozesses über mehrere Episoden signifikant zu verbessern. Dies zeigt das Potenzial von RL-Algorithmen, effektive und effiziente Lösungen für komplexe Optimierungsprobleme im Produktionsbereich zu bieten.

Contents

List of Figures

List of Tables

1 Introduction

The field of digital energy is rapidly expanding, resulting in a massive increase in data storage. This exponential growth in data has made data analytics an essential part of the subsurface energy industry, including both the upstream and downstream sectors. The emergence and growth of big data, artificial intelligence and machine learning have driven the adoption and growth of data analytics across various sectors of the supply chain.

The subsurface energy industry is a complex and dynamic system that requires constant monitoring and optimization. Data mining techniques can help extract valuable insights from the vast and diverse data collected from the field, such as predicting and preventing failures. By applying machine learning and data analytics to these data, it is possible to improve the performance and reliability of field operations, plan maintenance and repairs. The digital energy field is a critical enabler for smart and integrated management of subsurface energy resources.

Machine learning plays a significant role in the oil and gas industry, particularly in three main areas: real-time data extraction, predictive analytics and production optimization. The industry produces a colossal amount of data that can be analyzed in real-time using machine learning to develop diagnostic applications. Predictive analytics can help forecast oil production and determine the optimal way to design production systems. Some examples of these applications include advisory systems for injection rates, pumping strokes per minute, well spacing and testing different fracking techniques.

These three areas show how machine learning can help oil and gas companies make data-driven decisions, both for the present and for the future. As the industry embraces new technologies, enhancing workflow efficiency will be essential and machine learning has many potential applications that can contribute to this goal. These include real-time drilling, reservoir engineering, oil and gas production and procurement, downtime prevention and well testing, among others.

By utilizing machine learning, employees in the oil and gas industry can become more proficient at their jobs and improve the quality of their work. However, this transition will require careful consideration of the challenges involved. For instance, one of the most significant challenges is the need to

hire data scientists who can extract knowledge from industrial data. Given that the industry already faces a shortage of skilled workers and a lack of qualified candidates in some regions, this may prove difficult for some companies.

Another challenge is the massive amounts of data that oil and gas companies handle every day, necessitating the use of powerful machines to analyze these data in a reasonable amount of time. Additionally, training machine learning algorithms to recognize valuable data patterns specific to the oil and gas industry will require a significant investment of time and effort. Nevertheless, the benefits of using accurate models to make data-backed decisions are considerable and can potentially provide a significant competitive advantage for oil and gas companies.

The realization that machine learning techniques can be successfully deployed for oil and gas companies is still occurring in the industry. As more energy companies investigate ML use cases, an increase in its adoption will be seen. This will certainly change how oil and gas companies operate, and it is just one example of how data science is changing how business is done.

Oil and gas companies are already using machine learning to help with prediction models, produce better results from their resources and optimize their processes. Machine learning has the power to revolutionize the oil and gas industry. This forward momentum is crucial for the sector solely responsible for keeping the rest of society moving.

Over the next few years, the oil and gas industry is expected to increasingly leverage machine learning technologies to optimize their operations. The future of this industry lies in the convergence of multiple technological domains, of which machine learning is just one component. Early adoption of these technologies can help companies improve their resource allocation and gain deeper insights into their operations.

The purpose of this research is to develop a comprehensive framework for data analytics applications in the subsurface energy systems industry. The framework encompasses all three stages of data analysis, namely descriptive, predictive and prescriptive analytics.

The research presents three specific applications that have been developed using this framework. First, a virtual flow meter has been implemented on an electrical submersible pump, utilizing a descriptive analytics approach. This application provides a detailed understanding of the pump's performance by analyzing the pump's data and generating insights on its flow rates.

Secondly, a predictive maintenance application has been developed for the electrical submersible pump, utilizing a predictive analytics approach. This application can predict the likelihood of pump failures, based on historical data and machine learning algorithms and can recommend maintenance actions to avoid future failures.

Lastly, a prescriptive analytics approach using reinforcement learning has been used to develop an optimal policy for steam injection rate. This application utilizes machine learning algorithms to learn from historical data and recommend the optimal steam injection rate for efficient energy production.

This thesis consists of eight chapters. Chapter 1 provides an overview of the subject of the study. Chapter 2 reviews relevant literature on real-time, predictive and prescriptive analytics of petroleum production systems. Chapter 3 describes the problem statement, objectives and methodology. Chapter 4 discusses data-driven multiphase flow estimation through artificial lift systems, focusing on electrical submersible pumps. Chapter 5 presents predictive maintenance of electrical submersible pumps through pre-workover event prediction. Chapters 6 and 7 introduce actor-critic reinforcement learning for decision-making in energy systems optimization, including a proof of value for steam injection and waterflooding optimization. Chapter 8 showcases model accuracy and validation. Finally, Chapter 9 concludes the thesis and provides recommendations for future research.

2 Background and Related Work

2.1 Introduction

Oil production engineering uses field data and models to optimize well and surface production facilities. This chapter gives an overview of using machine learning in production modeling. It covers predicting well flow rates, detecting anomalies with predictive maintenance, and optimizing energy production systems. These techniques improve decision-making and lead to greater efficiency and success in production engineering.

Advances in digital transformation over the last couple of decades enabled technology framework to continuously optimize oil and gas fields (Cramer and Goh 2009). Continuous optimization requires the integration of field hardware (e.g., downhole sensors, remotely activated completions, and surface facilities). It also requires data analysis computer algorithms to be applied to the data for decision-making, virtual sensing, pattern recognition, predictive maintenance, and physical model integration.

Understanding production systems is essential in data analysis applications. Production engineering is an ongoing process that seeks to enhance technical and financial performance in the oil and gas industry by maximizing production system capacity while minimizing costs and effort. This process involves data acquisition, analysis, decision-making, and execution. The potential solutions To address common questions in production engineering involve data modeling, mechanistic modeling, or hybrid systems (physics-informed modeling). These questions include:

- Is the asset (reservoir, well, or facility) operating at its full potential production rate?
- What does the surveillance data indicate regarding the asset's operational health?
- What are the limiting factors and critical values impacting production?
- What is the most likely cause of production loss or deferral?
- What is the most effective action to arrest the decline and restore production?
- Will the chosen action be successful, profitable, and sustainable if implemented?

Answering these questions in production engineering is crucial to understand the operational status of the field and identify necessary measures to optimize production performance. This study aims to address these questions using data modeling or hybrid systems (such as physics-informed modeling), which provide insights for virtual sensing, predictive maintenance, and decision-making support.

This chapter presents a comprehensive literature review of production data-driven modeling methodologies. It categorizes production data into three main categories based on their nature of occurrence. The focus then shifts to the application of machine learning for virtual sensors, specifically virtual flow metering for electrical submersible pumped wells. It also explores the use of machine learning algorithms for predicting failures and monitoring through historical production data. Finally, it covers data-driven decision-making and flow control. A summary of all applications is presented in the concluding remarks.

2.2 Mapping production and reservoir data

A large variety and volume of data are captured through routine surveillance programs and assimilated. Accordingly, subject matter experts (SMEs) analyze all acquired data to assess the system status and establish the optimum operating envelope of wells and facilities. Table 2.1 presents a categorization of the production data. It could be categorized into static, Fluid Properties and time-dependent data. The static data represents the wellbore and completion data, which includes tubing size, casing size, casing setting depth, friction, perforation depth, etc. The second category, which is the reservoir fluid data, includes the various properties of the produced fluids. Finally, the time-dependent data is the data that changes over time.

The third category of data in production systems is time data, which comprises operating parameters that frequently change. Time data parameters can be broadly classified into two categories : environmental parameters and manipulating parameters. Environmental parameters are those parameters that respond to the dynamic system, but they cannot be directly controlled. Manipulating parameters, on the other hand, are those parameters that can be directly adjusted based on the decisions of subject matter experts (SMEs).

For instance, in an electrical submersible pumped well system, pump intake, discharge, motor temperature, pump frequency, choke opening, etc. are parameters. In such a system, pump intake pressure, temperature and

Table 2.1: Production data categorization

Data Categories	Production Engineering Component	
Static Data	Wellbore Data Completion Class A	Tubing Size, Artificial Lift used with its relevant characteristics pump type and its characteristic curve Perforation Depth
Reservoir Fluid Data	Fluid Properties Class B	Interfacial Tension Formation Volume Factor Density Viscosity
Time Data	Operating Parameters Class C	It depends on well completion (Gas Lift, Electrical Submersible Pumps, Sucker Rod Pumps, Natural Flow, Gas Nat. Flow, Plunger Lift...). These parameters could be controlled parameters, manipulated parameters or disturbance parameters Example Electrical submersible pump (ESP): Pump head Motor temperature Frequency Voltage ...

motor temperature are considered environmental parameters, while pump frequency and choke opening are manipulating parameters.

In reservoir engineering applications, artificial intelligence-based models are deployed to solve a large spectrum of problems in both forward- and inverse-looking manners (Ertekin and Q. Sun 2019). Table 2.2 lists the three categories of data to be processed, which include reservoir characteristics, project design parameters and field response data.

A forward-looking model utilizes the reservoir characteristics and project design parameters as input to predict the field response. A well-developed forward-looking model can be employed as an AI-based predictor to obtain quick assessments of certain project development strategies. Instead of rigorously solving the system of flow transportation equations, the forward-looking AI models generate predictions by developing some nonlinear relations between the input and output parameters. Therefore, the computational cost is much less intensive compared to high-fidelity numerical models (Cornelio et al. 2021; Hadi et al. 2019; Kubota and Reinert 2019; Noshi et al. 2019; Z. Zhao and D. Wang 2021).

Meanwhile, the AI models can be structured with two inverse-looking applications. Unlike the forward-looking models, the inverse AI models always use field response data (for example, fluid production and pressure measurement data) as input. The first objective of these models is called

Table 2.2: Reservoir data categorization

Data Categories	Reservoir Engineering Component	
Reservoir Characteristics Data	Geophysical data	Seismic survey data Well log data
Class A	Petrophysical data	Permeability distributions Porosity distributions Formation Depth Reservoir Pressure Reservoir Temperature Fluid contact
	Fluid Properties	Fluid Composition PVT data
	Rock/Fluid interaction data	Relative Permeability data Capillary Pressure data
Project Design Parameters Class B	Injection/Production well specification Well pattern design Well spacing Well architecture design EOR (Enhanced Oil Recovery) design parameters	
Field responses data Class C	Fluid Production data Pressure data Project economics	

Table 2.3: Structures of forward and inverse-looking AI models

Model Objective	Inputs	Output
Forward-looking models	A and B	C
Inverse History-matching models	C over B	A
Inverse Project Design models	C and A	B

"history-matching models," which use project design parameters and field historical data as input to characterize fluid and rock properties (Sengel and Turkarslan 2020).

The second objective of the inverse-looking applications aims at finding the engineering design strategy that fulfills the desired project outcome such as the hydrocarbon recovery (J.L. Guevara et al. 2021; N.. Sibaweihi et al. 2019). For projects with considerable capital and operational costs, such as the drilling of maximum reservoir contact (MRC) wells and large-scale chemical flooding, the implementation of an inverse design model would reasonably guide and place the project strategy on the right trajectory and significantly reduce project risks.

2.3 Increase well performance through virtual sensing

Proper estimation of multiphase flowrates in oil and gas production systems is an essential tool for monitoring and optimizing production systems. Hence, one of the routine tests of wells is production testing. It is usually conducted as a scheduled test to monitor liquid rates, water cuts and gas oil ratio (GOR). Production testing is easily conducted using the test separator to compare the actual production rate with the theoretical one. It is the most common form of production and reservoir surveillance. However, this technique has its limitations. The main limitation of this test is its insufficient resolution or repeatability to identify trends in liquid and water-cut rates over short periods of time. Another potential problem could be the duration issue. It is often the case in low-flow rates and deep wells that require several time-consuming partial or complete liquid holdup periods.

Later, an alternative solution to production testing was developed. This solution is called multi-phase physical flow rates (MPFMs). This technology depends on the idea of indirectly estimating multi-phase flowrates without separating the phases. This is done by tracking supplementary measurements of fluid phase properties such as velocity and phase fractions inside the device. These meters are usually installed at the wellhead so that the multiphase flowrates of a particular well can be tracked in real-time. One disadvantage of MPFMs is that they have higher capital costs (CAPEX) and operating costs (OPEX), as they also require frequent production calibration.

Subsequently, the technology of Virtual flow meter (VFM) is developed as an attractive technology in the oil and gas industry because of its low OPEX and CAPEX required. Many technologies have been developed to estimate rates and pressures from other indirect measurements (e.g., virtual metering or soft sensors). It depends on either analytical or data-driven models for real-time calculations of phase production. In analytical models, the near-well area, wells, pipelines, and production chokes must be simulated while data-driven VFM depends on the available measured data. Depending on the measurement data available, the production system can either be represented as a whole from the reservoir to the processing plant, or it can be divided into sub-models. Figure 2.1 shows a schematic graph of well sub-models and relevant data. Also, optimization algorithms may be used to modify flowrates and other tuning parameters. This is mainly to stabilize estimation models by reducing the discrepancy between model predictions and actual measurements.

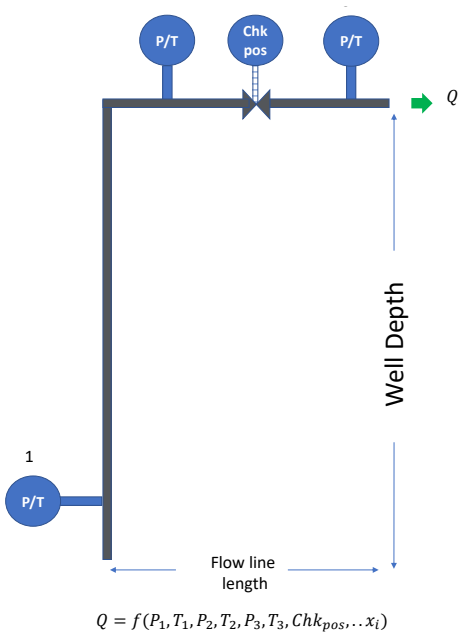


Figure 2.1: Schematic graph of well sub-models

For mechanistic models, many conservation equations take a dynamic form. However, due to the steady-state or quasi-steady-state nature of the optimization problem formulation, an optimization solver can only discover a solution for a single point in time or can use the solution from the previous step as a first estimate for predicting the current time step. In addition to dynamic optimization, Kalman filter approaches and other state estimation methods may be employed to develop a dynamic VFM (Borden et al. 2016). The disadvantages of the aforementioned VFM technique are the computationally costly nature of dynamic optimization for first-principles VFM systems (G. Falcone et al. 2001). Additionally, a high level of knowledge is required for setup and use, as well as the difficulty of tuning some of these models in a reliable way for actual field data.

Another approach is to focus on data-driven VFM, enabled by machine learning algorithms. It is based on gathering field data and mathematically adapting it to the production system's physical parameters, such as wellbore and choke geometry, flowline wall thickness, etc., without providing an explicit description of those parameters. The data-driven model can conduct quick and precise real-time metering if the model has been properly trained and the exposed conditions fall within the training range. This method can build models more affordably than mechanistic models since it does not require as much in-depth physics formulation of the systems.

In summary, virtual sensors are a convenient alternative to physical sensors, using available data during known conditions to predict other measure-

ments in instrumented wells (Vinogradov and Vorobev 2020). For example, establishing a machine learning model to relate (Well head pressure (WHP)), (Well head temperature (WHT)), and (Flow line pressure (FLP)) for a particular choke diameter can serve as a replacement for physical sensors when required. However, since most oil and gas wells undergo non-stationary processes where boundary conditions change with the field's life, the validity of the virtual sensor model may be limited to a specific period.

Data-driven virtual flow metering (VFM) is particularly useful when sufficient measured data, including frequent well tests (approximately 8-12 per year), permanent wellhead and flowline sensors (pressure, temperature) are available. The upcoming subsections will introduce various attempts at VFM for different production systems, encompassing an overview of the system, mechanistic virtual flow metering attempts and virtual sensing applications.

2.3.1 Surrogate sensing for ESP Wells

Electric submersible pumps are currently widely employed on many artificially lifted wells with high water cut, and offshore oil wells due to its simple structure and high efficiency (Takacs 2018). Among all the artificial lift systems, ESP is preferred because it can produce high volumes at higher temperatures and reach deeper depths. The development of sensors and data acquisition systems make it possible for ESP systems to continuously record the intake pressure and temperature, pump head, discharge pressure and temperature, motor temperature, motor current, leakage current, vibration, and so on. Those data would be recorded at regular intervals and transmitted to surface Remote Terminal Units (RTUs) (Carpenter 2019). Figure 2.2 shows a schematic graph of sub-modules and sensors deployed on the electrical submersible pumped well system.

Thanks to that advancement in communication, well technology, and field equipment, a lot of attempts are made for the estimation of flow rates in real-time. In the following, a summary of the applications of soft sensing both mechanistic and surrogate modeling are given.

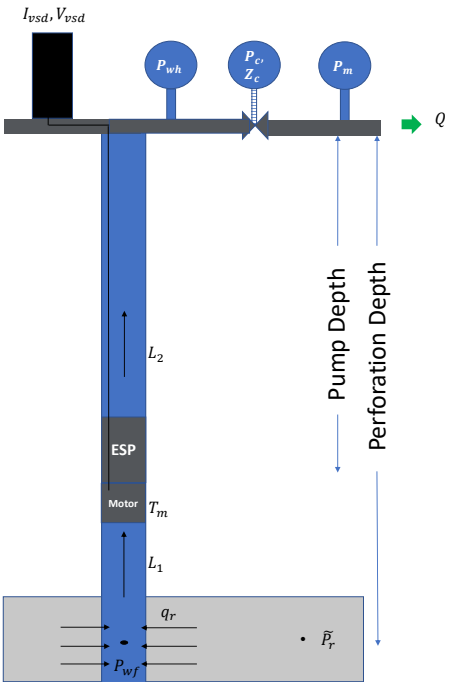


Figure 2.2: A schematic of ESP well

Table 2.4: Literature Study for the soft sensing on ESP wells

Author	Model summary
(L.. Camilleri, El Gindy, Rusakov, and Adoghe 2015; L.. Camilleri, El-Gindy, et al. 2016; L. Camilleri et al. 2016; L.. Camilleri, El Gindy, Rusakov, Ginawi, et al. 2017; Lawrence Camilleri and W. Zhou 2011)	In these papers, ESP models with different modifications are discussed, as well as field case studies where ESP first principles models serve as virtual flow meters. A hybrid method has been used to measure the flow rate without needing a test separator or multi-phase flowmeter. The drop of the pressure in the tubing provides measurements of the average density of the fluid, which is then converted to a water cut. A comparison was conducted between the calculated and the measured flow rates for a shale oil well equipped with an ESP to validate the calculation.
(Haouche, Adrien Tessier, et al. 2012a; Haouche, Adrien. Tessier, et al. 2012b)	The VFM model is a combination of three main units: the reservoir unit, the electrical submersible pump unit and the production tubing unit. A density correction factor is used to take into account the effects of gas on the operational performance of the submersible pump.

Table 2.4: Literature Study for the soft sensing on ESP wells

Author	Model summary
(Binder et al. 2015)	The authors of the study utilized a moving horizon estimator for flow rate estimation in a well equipped with an ESP, incorporating input from sensors such as bottomhole, downhole, and pump pressure sensors, as well as pump parameters. The method demonstrated high accuracy and was recommended for industrial applications.
(David Zhu et al. 2016)	In this research, singular spectrum analysis (SSA) was used on a raw production dataset without any pre-processing or transformation of the original series. They investigated the decomposition of the original series into a summation of the principal independent and interpretable components, such as slowly varying trends, cycling components, and random noise.
(Krikunov et al. 2019)	A hybrid physical-machine learning prediction model was developed. It utilized a range of motor frequencies. A numeric optimization model was created to suggest multi-well operating modes.
(K. Zhu et al. 2020; Dandan Zhu et al. 2021)	In these studies, a mechanistic model was developed to predict pump boosting pressure. The objectives were to forecast oil-water emulsion rheology and how it will affect ESP pressure boosts and describe the pump leak impact under the conditions of a gas-liquid flow.
(Sabaa et al. 2022)	This study aims to develop artificial neural network models to predict flow rates of ESP artificially lifted wells. Each data set included measurements for wellhead parameters, fluid properties, ESP downhole sensor measurements and variable speed drive (VSD) sensor parameters. The models consisted of four separate neural networks to predict oil, water, gas and liquid flow rates.

2.3.2 Surrogate sensing for SRP Wells

Beam pumping, or the sucker-rod lift method, is the oldest and most widely used type of artificial lift for most wells. A sucker-rod pumping system is made up of several components, some of which operate on the surface and others underground, down in the well. The surface-pumping unit, which drives the underground pump, consists of a prime mover (usually an electric motor) and a beam fixed to a pivotal post. The post is called a Sampson post, and the beam is normally called a walking beam.

Several sensors can provide measurements of sucker rod pump operations. One of the main measurements is the load on the pump, which forms what are called dynamometer cards. Dynamometers are diagnostic cards that measure the load on the top rod (polished rod) and plot this load in relation to the polished rod position as the pumping unit moves through each stroke cycle. The load-position plot of the polished rod is known as the surface card. Then, a wave equation solution is used to derive the downhole card from the surface card. The downhole card is a plot of load vs. position on the pump's plunger. Also, on the surface, continuous measurements for wellhead pressures, temperatures, and power measurements of the motor are reported. In addition, normal frequent test data for any production system is provided, such as fluid level depth using an acoustic transducer and production of multi-phases using a separator test or production test. Fig. 2.3 shows the sub-modules of the sucker rod pumped well with relevant tests, a dynamometer, and fluid level tests.

The applications of virtual sensing on sucker rod pumps are limited, possibly due to their limited ability to produce high fluid rates. However, some attempts have been made in this regard with various objectives. The first objective is to predict multi-phase flow rates or the dynamic fluid level in the annulus using dynamometer cards, wellhead pressure, and temperature as inputs. The second objective is to infer the dynamometer cards using electrical power data.

Virtual flow meter on rod pumping systems

The problem of predicting oil, gas, and water flow dynamically using the aforementioned pump sensor data is to establish a function that describes the multi-phase flow rates. Data-driven algorithms are used to find a relationship between the pump operational parameters and the produced oil, gas, and water. (Yi et al. 2019) used deep autoencoder-derived features from dynamometer cards to improve real-time production prediction models. The production prediction model, which combines more informative abstract

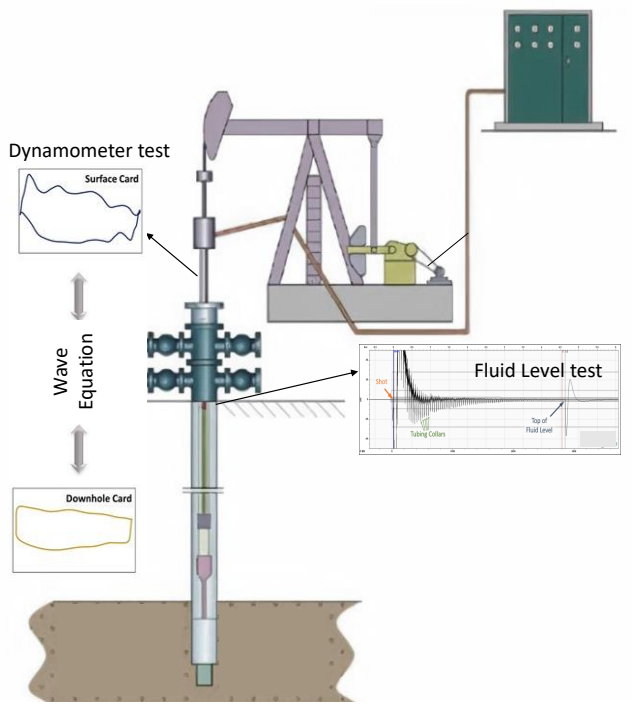


Figure 2.3: A schematic graph of SRP well

features with pump and production data, generates good agreement with the historical data.

Regarding soft sensing replacing the traditional detection method to model the dynamic liquid level of the sucker-rod pumping system, (Yang et al. 2014) proposes a method to calculate the dynamic fluid level. It uses the submerged pressure as a common solution node to analyze both the plunger load variation, which is contributed by the pump dynamometer card, and the pressure distribution in the annulus. (X. Li et al. 2013) presented a simulated annealing-based Gaussian process regression model.

Virtual sensing of the dynamometer card

Over the years, many researchers have studied the relation between electrical parameters and surface cards (S. Zhang and Tang 2008). Some theoretical formulas can also be built to calculate the card from the electrical parameters. However, some parameters in the formulas cannot be quantified or measured, and some assumptions about the values made the card calculation inaccurate and unstable. Thus, a machine learning model for dynamometer card calculation in the rod pumping lift process is used to formulate the complicated process. In these examples, deep neural networks are used to find a good weight combination, allowing the model to come up with rules

from the input data (electrical parameters) to the target data (dynamometer cards).

This study includes extracting power features and constructing an eigenvector in chronological sequence for one period. Afterwards, dynamometer card data and shape curve images are extracted according to coordinates and load data. Then, power features and dynamometer diagram features are normalized by row and mapped between 0 and 1. Finally, the sequence-to-sequence algorithm is used to infer dynacards from the power curve features (Y. Peng et al. 2019; Dandan Zhu et al. 2021).

2.3.3 Surrogate sensing for gas lifted wells

Gas lift (GL) is a method of artificial lift that uses an external source of high-pressure gas to supplement formation gas to lift the well fluids. The principle of gas lift is that gas injected into the tubing reduces the density of the fluids in the tubing, and the bubbles have a "scrubbing" action on the liquids. Both factors act to lower the flowing bottomhole pressure (BHP) at the bottom of the tubing.

One use case for the VFM modeling is to compute uncorrelated estimates of gas-lift rate by using the gas lift flow control valve performance model. Figure 2.4 shows the control volumes of a gas-lifted well. In this system, the inputs are manifold pressure, valve and casing pressure, while the output is gas lift rates.

(Al Selaiti et al. 2020) developed a data-driven approach to find the optimal operating envelope for gas-lift wells. The process involves building multilayer perceptron neural network models for generating instantaneous predictions of multiphase flow rates and other quantities of interest, such as GOR and WCT, using real-time sensor data at the surface, historical performance and sporadic test data. The models were developed to generate short-term (30-day) forecasts of cumulative oil, water, gas and liquid production, multiphase flow rates, WCT, GOR, and reservoir pressure. Using time-series forecasting models, a sensitivity analysis was performed to generate short-term well response for a selected number of combinations of choke settings and gas injection rates.

(Khan and Louis 2021) used AI techniques to develop a robust correlation to forecast production rates in gas-lift-assisted wells. The AI techniques used in this research included artificial neuro-fuzzy inference systems, ANN, functional networks and support vector machines. They collected test data from several gas lift wells and used ANN to develop an equation to forecast oil flow rate. Initially, they applied wide data analytics and then input data

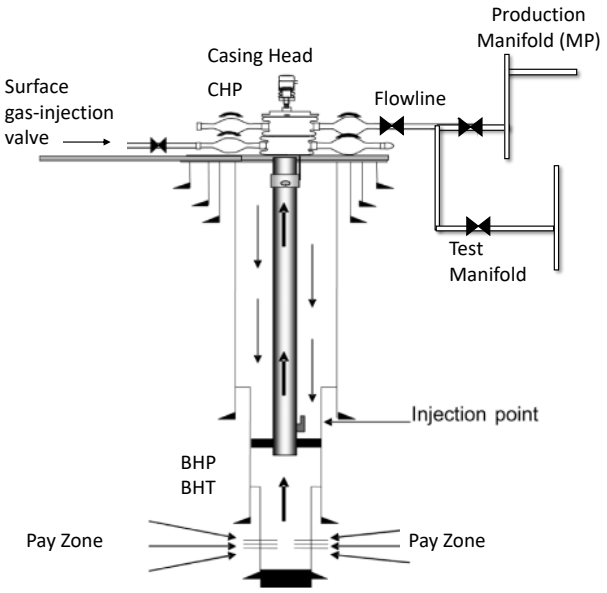


Figure 2.4: A schematic graph of GL well

to the models that were compared to each other and to other empirical models. They could predict oil rates with accuracy exceeding 98%.

2.3.4 Surrogate sensing for gas wells and plunger lifted wells

Conventional plunger lifting is a transient process that consists of cyclic openings and closings of a gas well. Because of this complex behavior, using traditional physics-based models to simulate the coupled behavior of reservoir and wellbore performance is computationally rigorous and challenging. Therefore, machine learning methodology would help in formulating the plunger-lifted well system, including plunger arrival time, tubing pressure, casing pressure and instantaneous gas flow rate.

The plunger is a hollow cylinder that travels up and down inside the tubing of the well. The plunger's upward movement creates a partial vacuum that draws gas from the reservoir into the tubing, while its downward movement pushes the accumulated fluid to the surface. The plunger is operated by a motor valve located at the bottom of the well. The valve alternates between open and closed states, controlled by a series of triggers. The opening and closing of the valve result in the cyclic action of the plunger, enabling the efficient removal of liquids from the wellbore. The plunger's arrival time is carefully calculated to ensure that it reaches the bottom of the well at the end of the liquid unloading cycle. The working mechanism of a conventional plunger lift is a complex process that requires a thorough understanding of

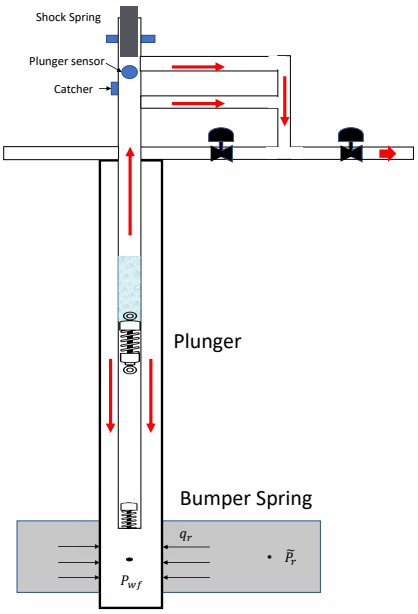


Figure 2.5: A schematic graph of a plunger lifted well

the triggers and their interplay with the reservoir and wellbore performance. Figure 2.5 shows the control volumes of a plunger lifted well.

As aforementioned, virtual flow metering includes physics-based and data-driven methods. When it comes to the plunger lift application, since the process is extremely transient, the application of physics-based methods are extremily complex (Akhiiartdinov et al. 2020). Regarding data-driven models, (Andrianov 2018) demonstrated the application of artificial neural networks (ANN) to simulate the transient behavior of severe slugging and liquid-loaded gas wells using Long short-term memory algorithm (LSTM). on the other hand, (Akhiiartdinov et al. 2020) have attempted to model a VFM on plunger lifted wells. In this study, the objective was to optimize the ”on” and ”off” periods of the control valve, which serve as parameters for building the response surface.

2.3.5 Miscellaneous applications for identifying flow regime

As aforementioned, there are common characteristics of VFMs. One of the main characteristics is the estimation of the fraction of each phase or, in other words, identifying the flow regime. Therefore, various applications arise in this area. Their objective is to construct a classification model to predict various flow regimes based on flow measurements. Their work focuses on the identification of multiple-phase flow regimes by implementing deep learning algorithms in addition to commonly used machine learning

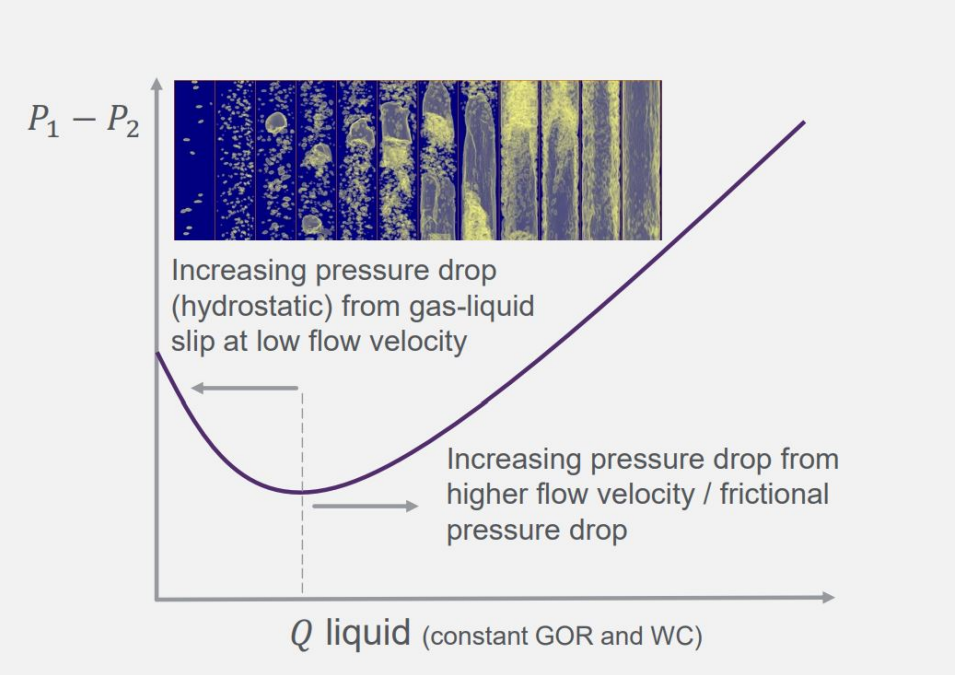


Figure 2.6: Multiple-phase flow regimes

algorithms (Roxas et al. 2022; Arteaga-Arteaga et al. 2021; Alhashem 2020; Manikonda et al. 2021; Mask et al. 2019; F. Popa et al. 2015; Ruiz-Diaz et al. 2022; Rammay and Alnuaim 2015). Figure 2.6 shows the relation between the pressure drop and flow rate with different flow regimes.

2.4 Monitoring and Failure Prediction

Engineers encounter numerous challenges when attempting to identify candidates and rank opportunities for production enhancement in complex systems. These challenges typically stem from the difficulties in comprehending the system's intricacies, accurately diagnosing issues and predicting potential incidents. As a result, machine learning techniques have been employed to diagnose wells and forecast potential failures. By leveraging data-driven predictions, inappropriate repairs can be minimized, downtime can be reduced and overall operational efficiency can be enhanced (Abdalla et al. 2020).

Predictive maintenance, a form of preventative maintenance, is an application of machine learning in the petroleum industry that uses sensor data from condition-monitoring devices and predictive analytics to predict equipment failure. Predictive maintenance has gained popularity as an effective

means of cost-cutting and productivity enhancement in the oil and gas sector. Various analytical techniques and machine learning algorithms have been utilized for this purpose.

In summary, predictive maintenance systems strive to detect early warning signs of equipment failure. In the oil and gas industry, these applications can be classified into artificial lift systems, critical parameter prediction, well integrity applications and other areas (Abdalla et al. 2020).

2.4.1 Artificial lift systems

Detecting Electric Submersible Pump Failures

The key to performing fault detection on the ESP can be better defined as the problem of building an accurate data-driven model that describes the ESP system dynamics. In this area of research, a lot of studies are performed to show that the use of artificial intelligence and machine learning with the concepts of petroleum engineering can predict the imminent or future failure of the electrical submersible pump, extend the life time of the pump and enhance the production.(ESPs). Table 2.5 shows various contributions in the area of predictive maintenance and diagnosis of electrically submersible pumped wells.

Table 2.5: Summary of the most relevant studies related with this work.

Author, Year	Relevant work
(Xi 2008)	A wavelet analysis is used to realize the extraction of excessive shaft thrust and wear fault characteristics.
(X. G. Li 2010)	Neuro-fuzzy networks are used and a dataset is gathered for the ESP in the case of eccentric wear of the impeller, sand plugging of the impeller and eccentric wear of the bearing.
(P. Zhao 2011)	they analyzed the vibration signal of ESP, the feature extraction and the establishment of typical fault vibration mechanical models.
(S. Liu et al. 2011)	Data analysis and application of vibration signals based on wavelet analysis and wavelet transform in the ESP are presented.

Table 2.5: Summary of the most relevant studies related with this work.

Author, Year	Relevant work
(Awaid et al. 2014)	The pattern recognition analysis is used to predict failure for a quick reaction and optimal solution for the input, Flowrate, WHP, Amps, $P_{discharge}$, P_{intake}, Pump dP and Motor temperature. By comparing real-time patterns of surface and downhole data with simple physical correlations, This study was able to reliably anticipate well and reservoir ESP performance.
(D. Guo et al. 2015; van Jansen Rensburg et al. 2019)	They explored surveillance-by-exception on ESP by only training the model with good-quality normal data and predicting anomalies from the data.
(Andrade Marin et al. 2019)	Analysed random forest to obtain a high value of accuracy and recall of ESP failure prediction in 165 cases.
(Bermudez et al. 2021)	used real-time applications of machine learning method to predict imminent and future failures, extend pump run-life and maximize the production of electrical submersible pumps (ESP's). Failure Prediction Index (FPI), Remaining Run Life (RRL) and the VFM are some of the Machine Learning models that have been implemented in this research.
(Barrios and Pydah 2021)	Through analysis and data comparison with the field operations, this study attempts to predict pump and motor performance under the two critical conditions of high viscosity and two-phase flow within the ESP. By doing this, the performance of the system may be anticipated by carefully examining its physical layout, fluid flow paths, ESP power delivery system, pump performance and motor performance and caisson separation characteristics.

Table 2.5: Summary of the most relevant studies related with this work.

Author, Year	Relevant work
(Adesanwo et al. 2016; Gupta et al. 2016; Abdelaziz et al. 2017; Bhardwaj et al. 2019; Sherif et al. 2019; L. Peng et al. 2021)	These studies applied the principal component analysis (PCA) for anomaly detection and failure prediction to identify correlations in the dynamic ESP parameters: intake pressure, intake temperature, discharge pressure, vibrations, motor temperature, motor current, systems current and frequency recorded by the Variable Speed Drive (VSD) at regular intervals.

Identification of sucker rod pump problems

Detecting workflow well failures in sucker rod pumped (SRP) wells leads to tremendous savings in engineers' time as well as minimizing production losses. The richest diagnosis tool for the sucker rod pumping system is the downhole dynamometer card (Abdalla et al. 2020). Therefore, most of the applications for sucker rod pump use downhole cards.

Dynamometer cards are used in both supervised and unsupervised learning scenarios. Supervised learning approaches require a pre-classified dataset. However, the unsupervised algorithms just need an input dataset without classification.

In a study conducted by (X.-y. Peng et al. 2009) self-organizing map (SOM) networks were utilized for unsupervised learning to classify five pumping conditions. The study achieved excellent results by using only the load values. Self-organizing maps is an algorithm that creates a network of points. This network adjusts to the input space features and approximates its density function in an ordered way. Essentially, the algorithm maps input values with similar results close to each other, forming clusters of related data points

For supervised learning, several models have been developed to automatically identify the problems of the sucker rod pumps using dynamometer cards (S. Liu et al. 2011; Gao et al. 2015; Abdalla et al. 2020; C. Wang et al. 2020a; C. Wang et al. 2020b; Xiaoxiao and Hanxiang 2020)

The differences between the previous applications were not only the various algorithms used as classifiers, but also the various feature engineering algorithms used. In other words, how the previous researches tried to represent the dynamometer cards. From all of the previous engineering algorithms,

Fourier transform for dynamometer cards was one of the promising techniques (Abdalla et al. 2020).

2.4.2 Increase well awareness by predicting system critical parameters

Critical velocity prediction for sand production

The oil and gas sector has recently paid close attention to sand transport in multiphase flow. Problematic factors related to sand production include productivity reduction, pipe corrosion, pressure loss and partial pipe blockage. Solid transport models are used to predict the fluid velocity required to transport solid particles in hydraulic and pneumatic systems. It is important that the processes in these applications are designed and operated at a sufficient fluid velocity to avoid solid deposition. Mechanistic models are used to provide a reasonable estimate for the minimum fluid velocity needed to transport the particles. However, those models are limited by the applicability of the empirically based closure relations that are part of such models.

In (Vieira and Shirazi 2022; Ehsan Khamehchi et al. 2014) studies, three machine learning algorithms (support vector machine, random forest, and extreme gradient boosting) were used. These techniques were used to establish the minimum flow rates required to transport grains efficiently in stratified and intermittent gas-liquid flow regimes. The inputs were sand concentration, pipe inclination, pipe dimension, fluid density, fluid viscosity, grain density, and grain dimension , while the objective is to predict the value of critical velocities in the pipe. The random forest model emerged as the most promising approach for accurately forecasting sand deposition. In fact, machine learning methods outperformed traditional correlations and mechanistic models in their ability to make accurate predictions.

In a recent study (Song et al. 2022), machine learning (ML) techniques were applied to predict the sand output of sands containing natural gas hydrates. The study utilized a sand manufacturing experiment to generate data sets for training four ML algorithms, namely K-Nearest Neighbor, Support Vector Regression, Boosting Tree and Multi-Layer Perceptron. The experiment considered eight factors, including well type, permeability, shale content, sand diameter, effective porosity, hydrate saturation, sand-retaining accuracy, uniformity coefficient and sand production as the output value, to train the ML models. The models' performances were evaluated using the mean absolute error and coefficient of determination metrics. The K-Nearest Neighbour model provided the least accurate predictions, while the Boosting

Tree model had the highest precision. However, a combined prediction model utilizing a support vector regressor and multi-layer Perceptron could be used to forecast sand production. This study demonstrates the potential of ML techniques for predicting sand output in natural gas hydrate sands.

Critical rate prediction for liquid loading method of gas pipeline

During the transportation of natural gas through a pipeline, the presence of water vapor in a saturated state is common and varies with temperature and pressure (Lea et al. 2003). As the temperature drops, the water vapor in the gas stream may condense into liquid water, which can collect in certain areas of the pipeline network in mountainous regions (Hong et al. 2022). This can result in an increase in friction, reduced gas flow, decreased gas transmission efficiency and negative impacts on the economics of the gathering pipeline network (Molnar 2022). Liquid loading is therefore a common issue in the transportation of gas through pipelines that can decrease transmission efficiency and cause flow problems. To prevent liquid loading, the gas velocity in the pipeline should be maintained above the liquid loading velocity. Accurately predicting and managing liquid loading are critical challenges in the field of multiphase flow research. New models and measurement techniques are continually being developed to address this issue.

In the natural gas industry, predicting the onset of liquid loading in gas wells is an important area of research and development. Scientists have developed onset prediction models that rely on various mechanisms such as droplet falling back, liquid film adverse flow or energy to determine the critical gas velocity or flow rate at which the flow regime transitions. Despite progress in this field, there is still a lack of a universally validated model that can predict the onset of liquid loading in versatile gas wells such as horizontal, vertical and inclined wells. This highlights the need for further research and development in this area.

There is, however, not much research on the use of ML in liquid loading prediction. (Osman 2002; E. Khamehchi et al. 2014; Ghadami Jadval Ghadam 2015; Hong et al. 2022; Abhulimen et al. 2023) employed ML to develop models to predict the onset of liquid loading and reported higher accuracy than existing models. (Osman 2002) developed a model to predict the critical gas flow rate for continuous removal of liquids from gas wells. The model was based on artificial neural networks and developed using published data of (Turner et al. 1969). Also, (E. Khamehchi et al. 2014) presented an artificial neural network model for predicting the minimum flow rate for continuous removal of liquids from the wellbore. The model was tested against actual

field data that was not used in the training phase. The results show that the developed model provides better predictions and higher accuracy than the published models. Also, (Ghadami Jadval Ghadam 2015) used comparative neural-fuzzy intelligent systems in order to detect the formation or lack of formation of accumulation of liquids.

Recent developments in machine learning (ML) have led to the creation of new applications for predicting natural gas pipeline liquid loading. (Hong et al. 2022) proposed an ML technique to predict pipeline liquid loading based on numerical simulation, with a strong generalization capacity. The authors set various working conditions according to the characteristics of the actual gas pipeline.These conditions are gas velocities, pipe diameters, water contents and outlet pressures. Multiple undulating pipeline topographies with various elevation differences were then established and data needed for machine learning was generated and collected using the OLGA simulator. Another study by (Abhulimen et al. 2023) enumerated the usage of software-based code environments and neural network architectures. These architectures incorporate the usage of a Bayesian neural network, an artificial neural network with a genetic algorithm or particle swarm optimization algorithms for the prediction of liquid loading in gas wells. The effectiveness of these models was evaluated using 106 datasets, as previously adopted by (Turner et al. 1969).

Critical rate hydraulically fractured wells

Hydraulic fracturing is the primary technique employed for the extraction of unconventional fossil fuels such as oil and natural gas from shale and other tight rock formations. A frac hit occurs when hydraulic communication takes place between two horizontally adjacent wells during the fracturing process, with the pumping of hydraulic fluid into a sub-well affecting an existing offset well, also known as the parental well, on the same or adjacent pad. These inter-well interactions can have a significant impact on overall well performance, including productivity and wellbore integrity (Y. Guo et al. 2022).

To improve the identification and understanding of frac hits, machine learning (ML) models have been widely applied. One such approach involves the use of long short-term Memory (LSTM) and multilayer perceptron (MLP) neural networks to creatively identify frac hits by analyzing time-series pressure and production data. This technique can be applied to both intra-pad and inter-pad interactions and has demonstrated promising results in terms of both predictive capabilities and workflow efficiency (Wu et al. 2022).

In addition to identifying frac hits, machine learning has been applied in various other applications related to hydraulic fracturing (Alimkhanov and Samoylova 2014; Anderson et al. 2016; Mohaghegh 2020). For instance, (Alimkhanov and Samoylova 2014) used eleven models to identify criteria for hydraulic fracturing candidates in reservoirs with complex geology. (Anderson et al. 2016) classified hydraulic fractures using a combination of physics-based models and machine learning techniques. Meanwhile, (Nande 2018) employed artificial neural networks (ANN) to develop an innovative methodology for predicting fracturing closure pressure and (Makhotin et al. 2019) introduced an ML model for estimating production gain following hydraulic fracturing operations in a Siberian oilfield.

2.4.3 Well integrity

Well integrity is a crucial aspect of the oil and gas industry, with significant implications for environmental and economic sustainability. It involves the implementation of a series of technical, operational and organizational measures throughout the entire life cycle of a well to ensure the safe containment and prevention of fluid leakage to subterranean formations or the surface. The integrity of a well is maintained by monitoring and assessing the performance of the well barriers, including the cement sheath, casing and blowout preventer (BOP). The effectiveness of these barriers is critical to prevent the release of formation and well fluids, which can pose significant risks to human health, the environment and the surrounding communities.

Well integrity management systems (WIMS) have been developed to ensure the effective management of well integrity. These systems involve the continuous monitoring and assessment of well integrity throughout the entire life cycle of a well, including design, drilling, completion, production and abandonment. The aim is to identify potential integrity threats and take proactive measures to mitigate them before they escalate into incidents or accidents. WIMS use a range of techniques, including risk assessments, inspection and maintenance programs, and advanced technologies such as artificial intelligence (AI) and machine learning (ML), to enhance the accuracy and effectiveness of well integrity management.

Remarkably, ML was used to manage and improve the integrity of wells. (AlAjmi et al. 2015) employed ANN to predict the downhole integrity of casing strings in wells. While (Dethlefs and Chastain 2012) established the wellbore damage analysis model using the fault tree analysis approach for numerous wellbore integrity criteria. These risk indicators are used to develop risk control strategies, which serve as a guide for estimating wellbore integrity and ensuring the safety of oil and gas output.

(Noshi et al. 2019) used nine ML models and determined which approach performed best in order to identify the characteristics influencing casing failure. Real-time data was employed by (Bilogan et al. 2019) to improve the processes of well integrity monitoring. The authors of the study discussed the outcomes of using a pilot approach for a sizable onshore asset. They created an ML model to identify the well events and detect abnormalities after taking into account the benefit of real-time data display.

(Elichev et al. 2019) proposed a method that enables detailed identification of various past or present occurrences and retrospective data analysis. The method was developed for naturally flowing wells but can be easily modified to work with artificial lift wells. (Yakoot et al. 2021) used ML to anticipate the risks associated with well integrity failures by analyzing well integrity data from gas-lift wells. The created model is a novel way to transform well integrity failures into measurable value, allowing precise tracking of any well's overall condition and possible risks.

2.4.4 Miscellaneous

Detection of Faults in Pipeline Systems

Pipelines are the most economical and efficient means of oil and natural gas transportation over long distances in different environments; however, they are subjected to corrosion and degradation. Pipeline accidents result in vast economic losses as well as catastrophic environmental effects such as oil spills. Natural hazards, mechanical failures, operational problems, corrosion and third-party activities are the most probable causes of oil pipeline failure. Therefore, machine learning algorithms are used to locate and detect leaks in liquefied gas pipelines based on the pressure and flow rates at the inlet and outlet of the pipeline.

An example of such studies is (X.-y. Peng et al. 2009; Morteza Zadkarami et al. 2020). The workflow includes employing the OLGA software and extracting the input pressure and output flow rate signals for various leakage scenarios with regard to different leakage locations and severity. It is followed by the statistical feature extraction of pressure and flow signals at different leakage scenarios. After extracting the statistical features the related input matrix includes 14,800 samples along with 16 dimensions. The output classes are dealt with as a multi-classification problem.

Chan Plot Signature Identification

In this study (Garcia et al. 2019), they used data from the water-oil ratio (WOR) and its rate of change to classify the status of WOR into four categories. These categories are: constant WOR, normal displacement, multilayer channeling.

The first category is constant WOR, where water merely follows the oil trend without any change. The second category is normal displacement, where the WOR and water cut (WC) gradually increase over time. This is common in mature wells where the WC can grow to as high as 80% or more. The third category is multilayer channeling, where there is a sudden and clear shift in the slope from a constant WOR or normal displacement situation. This increase can be attributed to a breakthrough from the most conducive layer in the well, resulting in an increased flow of water into the well.

(Garcia et al. 2019) studied how different ML model decision boundaries behave in this dataset. Specifically, naive Bayes, nearest neighbor and radial basis Support vector machine (SVM) displayed nonlinear decision boundaries, whereas linear SVM and multinomial logistic regression displayed linearly separable decision boundaries. The decision tree and random forest displayed a different type of decision boundary than the other models. They concluded that the nearest neighbor model achieved the highest f1-score value of 0.93, whereas nave Bayes, linear SVM, and radial basis SVM achieved the lowest f1-score of 0.90. Decision tree, random forest, and logistic regression achieved an f1-score of 0.90 to 0.91.

2.5 Recommending optimum actions for flow control

Machine learning is a rapidly developing field that is transforming our ability to describe complex systems from observational data rather than first-principles modeling. Until recently, these methods had largely been developed for static data, although there is a growing emphasis on using machine learning to characterize dynamical systems.

The usage of machine learning for dynamic optimization, either to learn control laws (i.e., to determine an effective map from sensor outputs to actuation inputs) or support decision making, is very recent (Brunton and Kutz 2019). Specific machine learning methods for control include adaptive neural networks, genetic programming and reinforcement learning. Many of

these machine learning algorithms are based on biological principles, such as neural networks, reinforcement learning and evolutionary algorithms.

All of these model-free methods have some sort of macroscopic objective function, typically based on sensor measurements (past and present). Such applications are still very few and limited in the engineering process. In the field of subsurface energy decision management and flow control objectives, they can be categorized into two main groups: Control autonomous drilling and production optimization through closed-loop reservoir management.

In the context of autonomous drilling control (H. Liu et al. 2018; ArnØ et al. 2020; Yu et al. 2021), the objective is tracking some set points; therefore, it is often either a penalty function for downhole pressure in the case of managed pressure drilling or for landing position, final inclination, and maximum curvature in the case of controlled directional drilling positioning.

In the context of production optimization, the general configuration of closed-loop reservoir simulation consists of two main parts as follows: (1) model-based data assimilation, which acts as reservoir parameters and states an estimator, and (2) a model-based optimizer. As mentioned above, the task of the optimizer is to maximize the oil recovery factor or other desired economic criterion such as the Net present value (NPV). The required inputs for the optimizer part may be the injection data, production data, the hydrocarbon price, the predicted interest rate and the operating costs.

Based on the mentioned information, the optimizer calculates the optimal values of manipulated variables, which are normally selected as water injection and bottom hole pressure (BHP) trajectories. As updated measurements become available, the optimization process can be repeated. Using data assimilation techniques provide the facilities to estimate the required parameters and states for the reservoir modeling phase. Such algorithms can be classified into two main categories: global search methods and policy or trajectory search methods.

Evolutionary techniques such as genetic algorithms (GA) and particle swarm optimization (PSO) are popular examples of derivative-free optimization methods. They are used for global search methods, which explore the optimization space more comprehensively than local methods. Another main advantage of derivative-free optimization methods is their ability to be implemented in parallel processing configurations, which reduces elapsed computational time and increases the efficiency of the algorithms (Shami et al. 2022).

On the other hand, policy search is a technique used in dynamic optimization problems to find an optimal policy that maps states to actions in order

to maximize a certain objective. It is a crucial aspect of addressing the nonlinear nature of the recovery process, which is often considered to be one of the most challenging aspects. Various ensemble-based optimization methods (Dehdari et al. 2011) and optimization techniques based on quadratic interpolation models (Y. Zhao et al. 2011) have been developed to solve production optimization problems. Additionally, efficient gradient-based approaches that utilize the adjoint technique to compute the necessary objective function derivatives have been developed. However, the need for access to the source codes of reservoir simulators and the computational expense of derivative approximations remain significant challenges in practise (Horowitz et al. 2013; Wen et al. 2014; J. Jansen 2011).

Reduced order models have emerged as an attractive alternative for handling the complexity of reservoir simulation models. These models provide an efficient means of generating approximate solutions with reduced computational resources while preserving the main features of the original model (He and Durlofsky 2011). Another approach for optimizing production of oil and gas reservoirs involves equalizing water breakthroughs of producing wells based on time in streamline simulators (Datta-Gupta et al. 2010). This technique aims to identify and control the water injection rate into the reservoir to prevent early water breakthroughs and ensure optimal recovery.

Moreover, utilizing a reservoir simulator as a black box for solving the optimization problem based on data-driven techniques is gaining popularity in the field of reservoir management. This approach involves training a machine learning algorithm on simulation data to predict the reservoir response to various production strategies. Generalized pattern search (GPS) (Audet and Dennis Jr 2004) and mesh adaptive direct search (MADS) (Audet and Dennis Jr 2006) are two local search derivative-free optimization techniques that are widely used for optimizing reservoir production under uncertainty.

Reinforcement learning (RL) is a relatively new approach in derivative-free optimization methods that have been used in various studies for production optimization. For instance, previous studies (Ma et al. 2019; Hourfar et al. 2019; Miftakhov et al. 2020) have applied RL for waterflooding optimization projects, while (A. Sun 2020) used RL for CO2 storage optimization, and (Guevara et al. 2021) applied the state action reward state action (SARSA) algorithm for steam-assisted gravity drainage.

Several other studies have also applied RL in reservoir and production optimization. For example, (De Paola et al. 2020) applied RL for optimal well placement using the proximal policy optimization (PPO) algorithm. In this study, the PPO algorithm learned the optimal well locations in a

reservoir simulation environment by maximizing the production of hydrocarbons. In contrast, (Dawar 2021) used a different RL algorithm, called deep deterministic policy gradient (DDPG), to learn optimal well placement for maximizing the cumulative oil production over a specific period of time. The authors in this study used a reservoir simulator to model the reservoir, and the DDPG algorithm learned the optimal well placement for achieving the maximum cumulative oil production.

RL has emerged as a powerful technique for addressing optimization problems in the field of reservoir and production engineering due to its ability to learn optimal solutions from experience. The application of RL in production optimization has the potential to enhance production efficiency, reduce operational costs, and minimize environmental impacts.

2.6 Concluding remarks

In this literature study, the contributions of machine learning and deep learning in production systems have been reviewed. The previous work is categorized into three categories: virtual or surrogate sensing, monitoring and failure reduction and data-driven recommendations for optimal actions for production optimization.

Based on the literature review, it appears that data-driven models offer considerable potential for virtual flow metering of artificially lifted wells, particularly those equipped with electric submersible pumps (ESPs). However, the creation of robust and accurate models capable of generalizing to diverse operating conditions is still in its infancy. Some of the primary obstacles include obtaining high-quality training data, selecting appropriate algorithms and ensuring model interpretability. Furthermore, it is necessary to assess the models' predictability using an independent testing dataset to guarantee their effectiveness in practical applications.

To address these challenges, a methodology that includes exploratory data analysis, symbolic regression, XGBoosting and deep learning algorithms is proposed. This methodology aims to develop stable and accurate data-driven models for virtual flow metering of ESP wells, while also ensuring model interpretability and generalization to new operating conditions. Furthermore, the predictability of models is evaluated by using an independent testing dataset to ensure robust performance in real-world applications.

Furthermore, while there has been significant progress in applying machine learning techniques to solve production engineering problems, there are still some gaps and challenges that need to be addressed. For instance, the

majority of the previous applications in the monitoring and failure reduction category have focused on real time diagnosis with limited contributions to the area of predictive maintenance. To address this gap, there is a need to develop more predictive maintenance solutions that leverage machine learning algorithms and production data to detect early signs of equipment failure and schedule maintenance before equipment failure occurs.

Another gap identified is the need for data-driven recommendations for optimal actions for production optimization. Closed-loop reservoir management is an area where machine learning and deep learning can make a significant impact. Closed-loop reservoir management involves using real-time data and advanced analytics to optimize reservoir performance. While some progress has been made in this area, the full potential of machine learning and deep learning has yet to be realized. This gap provides an opportunity for researchers to develop new solutions that leverage machine learning and deep learning algorithms to optimize production, leading to improved reservoir performance and increased production.

To address these gaps, two contributions are proposed. The first contribution applies reinforcement learning (RL) to optimize the steam injection rate for a single agent system. By utilizing previous experiences or interactions with the environment, our approach aims to find an optimal policy of injection rate to maximize the net present value without human intervention.

The second contribution involves implementing multi-agent cooperative-competitive reinforcement learning in a waterflooding model. This approach allows for collaboration between multiple agents to achieve a common goal while simultaneously competing against each other. By leveraging the benefits of multi-agent learning, the waterflooding process can be optimized and cumulative production performance can be improved.

In conclusion, machine learning and deep learning have made significant contributions to performance prediction and optimization in production systems. However, there are still gaps that need to be addressed, such as the limited contribution of previous applications to the area of predictive maintenance and the need for data-driven recommendations for optimal actions for production optimization. Developing closed-loop reservoir management solutions that leverage machine learning algorithms and production data can also help operators make better decisions in real time. Our proposed contribution using RL for production optimization is one step towards addressing these gaps and improving production efficiency.

3 The objectives & the data analysis process

3.1 Introduction

The data analytics strategy encompasses three primary objectives: descriptive, predictive, and prescriptive analytics. Descriptive analytics are intended to provide insight into what has happened. Second, predictive analytics help to model and forecast what might happen. Third, prescriptive analytics are used to determine the best solution or outcome among various choices, given the known parameters.

First, descriptive modelling is a mathematical process that describes real-world events and the relationships between factors responsible for them. Afterward, Alarms and models are deployed that enable subject matter experts (SMEs) to unleash various behaviours downhole. Inferring real-time measurements using ground-truth values from previous sensor measurements or production tests also lie in this area of research. For instance, the technology of VFM is considered one of the main applications in this area.

Following that, predictive modelling is presented. This technique uses historical data to forecast future events, with a focus on events that will occur in the future. Predictive maintenance is one of the most significant applications of this approach. It heavily relies on machine learning and deep learning algorithms to detect early warning signs in sensor data. This enables the identification of potential issues before they become critical, helping to minimize downtime and optimize asset performance.

Finally, prescriptive modelling is referred as the ”final frontier of analytic capabilities.”Applying mathematical and computational sciences to make recommendations for decisions is known as prescriptive analytics. For example, the technological. For instance, the technology can predict the optimum operating parameters, such as pump frequency and injection rate, in order to maximize the NPV.

3.2 Machine Leaning techniques/paradigms

Machine learning involves the use of automated model building to analyze data.Machine learning is the process of teaching computers to do tasks like classification, clustering, and anomaly detection for data analysis.There are three main categories of machine learning algorithms: supervised learning, unsupervised learning, and reinforcement learning.

In supervised learning, the machine learning algorithm maps the input features to a known desired output. This category can be further divided into regression and classification problems, which are based on the type of output variable. Regression is used when the output variable is continuous, while classification is used when the output variable contains multiple classes or labels.

In contrast, unsupervised learning does not have an explicit output variable, and relationships are generated based on the provided data. Unsupervised learning algorithms can reveal hidden structures and relationships between input features.Clustering, dimensionality reduction techniques, and associative rule learning are examples of unsupervised learning. .

Reinforcement learning is The third paradigm of machine learning algorithms. It is designed to associate a reward or penalty with a sequence of decisions made by the algorithm. This reward or penalty helps the algorithm learn the set of decisions it should make to achieve a defined objective. These algorithms are modelled using the Markov decision process (MDP). Reinforcement learning is often considered ”semi-supervised” learning, but in an uncertain and potentially complex environment, the algorithm employs a trial-and-error approach to find solutions by being penalized or rewarded for the actions it performs. An example of a reinforcement learning application is robotics for industrial automation. For instance, (Z. Zhou et al. 2021) presented a novel application of off-policy, model-free deep reinforcement learning for high-precision robotic assembly tasks in unstructured environments. The proposed approach utilized reinforcement learning to enable position-controlled robots to perform assembly tasks compliantly through training without the need for manual tuning.

In conclusion, machine learning has revolutionized the approach to scientific and industrial problems. The three primary paradigms of machine learning are supervised learning, unsupervised learning, and reinforcement learning. Supervised learning predicts outcomes based on labelled data, while unsupervised learning discovers hidden patterns and relationships in unlabelled data. Reinforcement learning enables an agent to learn an optimal policy that

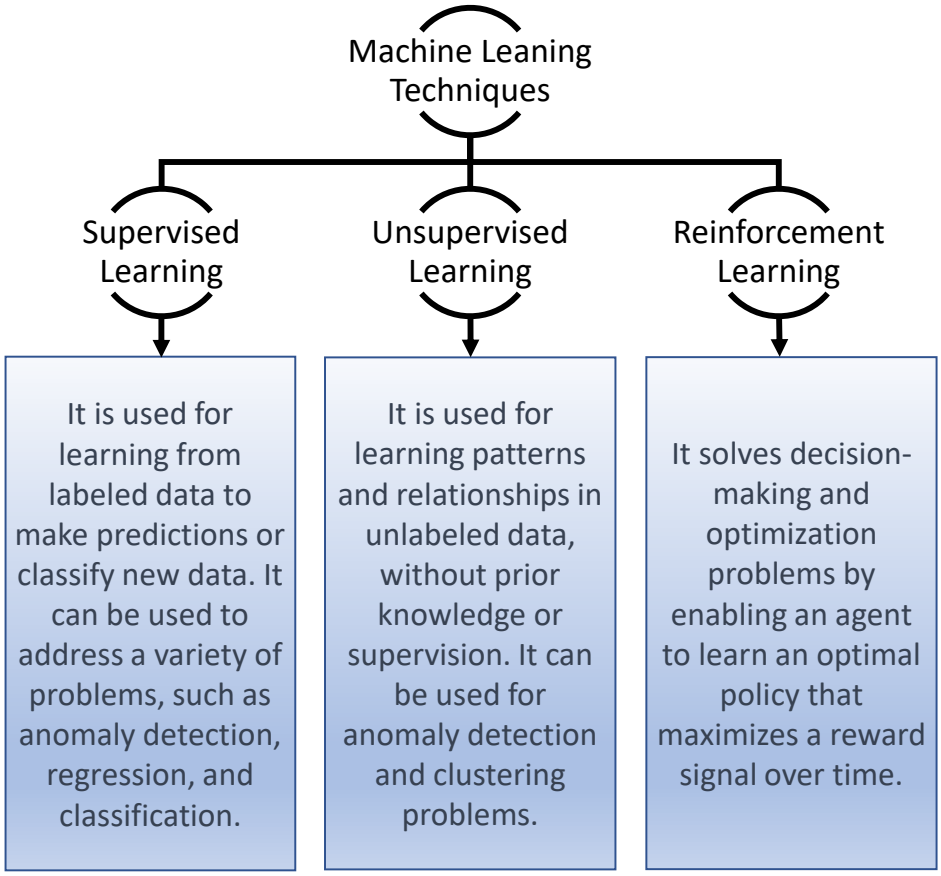


Figure 3.1: Overview of machine learning techniques and their problem domains

3.3 The analysis process pipeline

All of the applications mentioned above follow the process of data analysis. Basically, all of them involve understanding the data and then applying statistics to get insights for an engineering objective. Each study is divided into four modules. Each module has a big impact on the overall performance of the problem. These four modules are:

- Data gathering and exploratory data analysis
- Feature extraction and transformation
- Modeling
- Testing and Evaluation

3.3.1 Data gathering and exploratory data analysis

Once the objective is established, a strategy needs to be created for gathering and aggregating the appropriate data. These data may be quantitative (numeric) data, such as pump frequency for electrical submersible pumps (ESP) or stroke per minute for sucker rod pumps (Sucker rod pump (SRP)), or qualitative (descriptive) data, such as workover failure reasons, etc. In general, the training dataset must include a number of cases from the aforementioned data. Each case contains values for a range of input and output variables.

The next step is data cleaning, which is a crucial step that enhances the quality of data. Many tasks are included in data cleaning, such as removing major errors and duplicates, detecting outliers, and extracting irrelevant observations that have no bearing on the intended analysis.

Likewise, it is important to carry out an exploratory data analysis. This helps identify initial trends and characteristics and can even refine the hypothesis. It will give an idea of which inputs are likely to be influential. Exploratory data analysis helps to reveal hidden patterns, correlations, and anomalies using various tools such as visualizations (scatter and box plots, histograms, etc.), dimensionality reduction techniques, and sometimes unsupervised algorithms for data grouping.

3.3.2 Feature Extraction and transformation

Feature extraction plays a crucial role in the success of pattern recognition applications. A feature refers to a distinctive attribute or property of an object that distinguishes it from others. Machine learning applications often struggle with large volumes of input data in the form of waveforms or images represented as pixels. In such cases, the input data can be transformed into a reduced set of features, also known as a feature vector, through a process called feature extraction. The selected features should contain relevant information from the input data, enabling the desired task to be performed using this reduced representation instead of the complete initial data.

In essence, feature extraction involves taking an initial set of measured data and creating informative and non-redundant derived values (features). This process simplifies subsequent learning and generalization steps.

3.3.3 Modeling

Modeling is the process of developing mathematical or statistical representations of real-world systems or phenomena. These models can be used to simulate or predict behavior, identify relationships between variables, or gain insights into complex systems.

There are different types of modeling approaches depending on the objectives of the analysis. Descriptive modeling is used to describe patterns and relationships within the data. It is typically used to understand the current state of a system or process.

Descriptive modeling involves summarizing and visualizing the data to gain insights into its characteristics and structure. Predictive modeling is used to make predictions about future outcomes based on past data. Prescriptive modeling is used to recommend a course of action to achieve a particular outcome. It is often used in decision-making applications where there are multiple options and the goal is to choose the best one based on the available data.

Modeling involves selecting the appropriate algorithm or technique based on the objectives of the analysis and the nature of the data being analyzed. These algorithms exhibit a trade-off between learning performance and explainability, with more complex algorithms, such as deep neural networks, being highly accurate but more difficult to interpret. Figure 3.2 maps different learning algorithms of AI systems according to their learning performance (i.e., accuracy) vs. explainability.

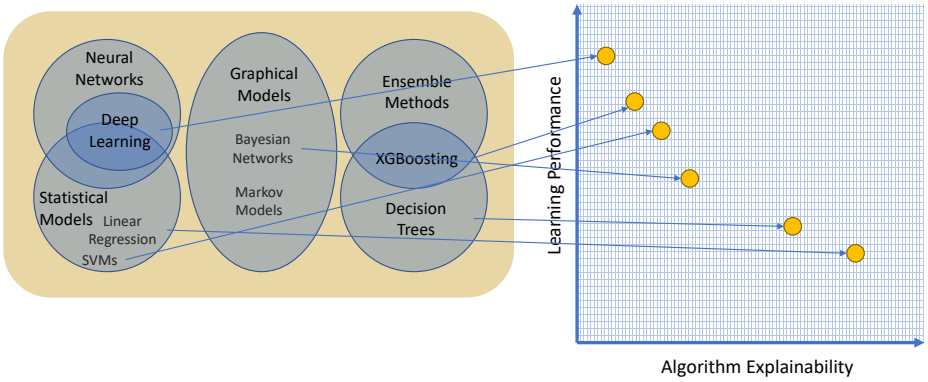


Figure 3.2: Comparison of Machine Learning Algorithms (Learning Performance vs. Explainability)

To conclude, modeling is a critical aspect of data analysis that plays a vital role in achieving business objectives. The three main categories of

business objectives in modeling are descriptive, predictive, and prescriptive. Whether the goal is to understand the current state of a system, make predictions about its future behavior, or optimize its performance, modeling can provide valuable insights and guidance. Ultimately, effective modeling requires careful consideration of data preparation, algorithm selection, and model evaluation to ensure both accuracy and explainability. Figure 3.3 shows the whole data analytics modeling objectives. A brief overview of each modeling business objective is provided.

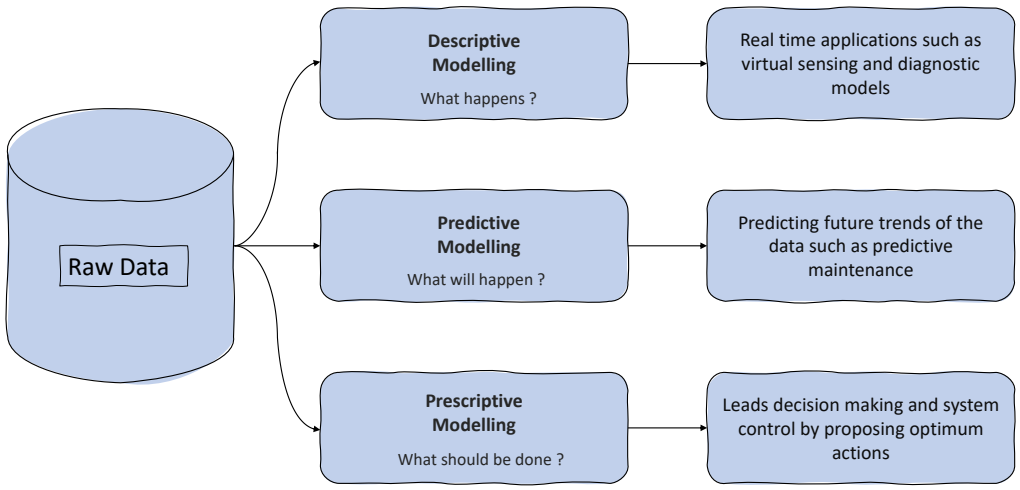


Figure 3.3: Data analytics modeling domain Objectives

Descriptive modeling for Real time estimation and diagnostic modeling

Descriptive analysis is an important aspect of data analysis that involves summarizing past raw data into models. These models are valuable because they can learn from past behaviors and provide insights into how they may impact future outcomes. The past raw data can be from any point in time that an event has occurred, whether it was one minute ago or one year ago. For instance, in the oil and gas industry, the sensor data deployed in wells can be used to create a model to infer well production. By analyzing and summarizing this data, a better understanding of the underlying patterns and trends that drive fluid flow through the well can be gained.

Diagnostic modeling is a valuable tool for understanding the root causes of problems or anomalies in complex systems. In the oil and gas industry, diagnostic modeling can be used to analyze data from various sources, such

as well logs, drilling parameters, and production rates, to identify the reasons behind unexpected events, such as a decrease in production or an increase in water cut. By understanding the underlying causes of these issues, engineers can develop effective solutions to improve well performance and optimize production.

One example of diagnostic modeling in the oil and gas industry is the use of dynamometer cards to analyze the performance of sucker rod pumps. Sucker rod pumps are commonly used in artificial lift systems to extract oil and gas from wells, but their performance can be affected by various factors, such as wear and tear, fluid properties, and well conditions. By analyzing the dynamometer cards, which record the load and position of the sucker rod during operation, engineers can identify the specific issues affecting the pump, such as a broken rod, a worn or damaged plunger, or inefficient inflow. This information can then be used to adjust the pump settings, replace faulty components, or modify the lift system to improve performance. Further information about these applications is discussed in section 2.4.

Predictive modeling for predictive maintenance

Predictive analysis is used to identify future trends and accurately forecast occurrences based on historical data. It has the ability to "predict" what might happen. It has grown increasingly important in recent years due to the evolution of machine learning and deep learning. It is of high importance to the oil and gas industry because it helps increase production efficiencies through the proactive identification of failure events and the avoidance of deferment losses.

Predictive maintenance is a type of predictive analytics that uses condition-monitoring sensor data to predict when equipment will fail. By predicting equipment failures, companies can schedule repairs during routine downtime, avoiding unplanned downtime and production losses. However, in many literature applications, the term "predictive maintenance" is used interchangeably to refer to diagnostic modeling or operating data grouping. A key application in oil and geothermal energy systems is developing an early failure prediction model for artificial lift systems.

Prescriptive modeling for optimising manipulating parameters

Prescriptive analytics is a relatively new field that goes beyond descriptive and predictive analytics by recommending one or more possible courses of

action. It is used to ”prescribe” a number of different possible actions and guide them towards an optimum solution. It serves as an advisory system for the future and is the final step in the analytics process. It is also the most complex, as it incorporates aspects of all the other aforementioned analyses.

Prescriptive analytics is quite different in terms of its application level. It uses a combination of techniques and tools such as simulation studies, computational modeling procedures, and machine learning (ML).

A prominent example of prescriptive analytics is the algorithms that guide Google's self-driving cars. Every second, these algorithms make countless decisions based on past and present data, ensuring a smooth and safe ride. Prescriptive analytics also helps in decision-making for long-term changes in manipulating parameters.

If implemented correctly, predictive analytics can have a large impact on decision-making and the company's bottom line. Technically, such analytics would contribute to the area of production optimization by making decisions on manipulating parameters such as choke opening, pump frequency, injection rate, and pump stroke per minute, among others.

In a nutshell, these analytics are all about providing advice. Prescriptive analytics attempts to quantify the effect of future decisions to advise on possible outcomes before the decisions are actually made.

3.3.4 Testing and evaluation metrics

Model evaluation is very important in data science as it helps understand the performance of models and how well they generalize to an unknown dataset. In this section, some evaluation metrics are discussed, whose objective is to quantify the prediction accuracy of a model, based on which the best-performing algorithm is selected. The metrics are also dependent on the type of problem, whether it is a regression or classification problem.

For quantitative outcomes in a regression model, some of the commonly used metrics include the coefficient of determination (R2), mean absolute error (MAE), and mean squared error (MSE). MAE is the average difference between the original values and the predicted values, which gives a measure of how far the predictions were from the actual output. MSE is quite similar to MAE, the only difference is that MSE takes the average of the square of the difference between the original values and the predicted values.

For classification outcomes, the model accuracy is defined as the ratio of the number of correct predictions to the total number of predictions. For

a classification problem with N different classes, a confusion matrix, or an error matrix is also used, which is an N*N matrix containing N correct classifications on the major diagonal and the rest of the possible errors on off-diagonal entries. This matrix provides a visual approach to evaluate the performance of a machine learning model for classification problems. An illustration of a confusion matrix for binary classification (two classes) is shown in Figure 3.4.

		True diagnosis		
		Positive	Negative	Total
Screening test	Positive	a	b	$a+b$
	Negative	c	d	$c+d$
	Total	$a+c$	$b+d$	N

Figure 3.4: Confusion Matrix

For data-driven optimization outcomes and reinforcement learning, the ”learning curve” refers to the plot that shows how the agent's behaviour changes over time as it receives feedback or reward signals from the environment. Specifically, the learning curve shows how the magnitude or frequency of the agent's conditioned response changes as the number of reinforcement episodes increases. To test the effectiveness of the reinforcement learning model, researchers can compare the trajectories generated by a base policy (the agent's initial policy) and an optimum policy predicted by the agent for validation. By comparing these trajectories, one can assess whether the agent has learned an optimal policy that maximizes its reward or if it needs further training.

3.4 Thesis Objectives

In technological terms, this study aims to contribute to the area of data analytics in subsurface energy systems through the development of intelligent computing applications. These developed applications target the three objectives of data analytics namely descriptive, predictive, and prescriptive objectives. They are implemented using machine learning and deep learning algorithms.

In scientific terms, the primary objective of this study is to develop four applications that leverage intelligent computing techniques to advance data analytics in subsurface energy systems. The first model predicts oil production and water cut in wells with electrical submersible pumps. In other words, it estimates multiphase flow rates in real time. The second

model analyzes the performance and patterns of pump sensor measurements during pre-event days to identify relationships and patterns. The third model predicts the optimal injection rate policies for steam injection projects. Table 3.1 provides a summary of the studies and their objectives.

Table 3.1: Summary of Research Projects and Objectives

Study	Modeling Objective Domain	Study Objective
Virtual flow meter for electrical submersible pump (ESP) wells	Descriptive modeling	Develop a real-time estimation method for oil rate and water cut using deployed pump sensors measurements
Predictive maintenance of ESPs	Predictive modeling	Develop a classification model for abnormal signals based on ESP sensor data with a lead time of 7 to 10 days before failure
Steam injection rate optimization	Prescriptive modeling as a single-agent application	Develop an actor-critic reinforcement learning algorithm to optimize the injection rate policy over time and maximize the net present value (NPV) in steam injection projects
Waterflooding rate optimization	Prescriptive modeling as a multi-agent application	Develop a data driven framework to optimize the injection rate policy over time in waterflooding projects, with the objective of maximizing the NPV while accounting for the interaction between multiple wells

3.5 Summary

Data analytics involves the use of statistical and computational methods to analyze and interpret data, with the aim of extracting useful insights and knowledge. In particular, data analytics can be used to achieve three domain objectives: descriptive, predictive, and prescriptive modeling. Descriptive modeling involves the use of data to describe what happens in real time. Predictive modeling, on the other hand, aims to forecast what may happen in the future based on historical data patterns. Lastly, prescriptive modeling involves the use of data to make recommendations on what actions to take in order to achieve a specific objective.

The achievement of these modeling objectives often requires the use of various machine learning algorithms, which enable the discovery of patterns, trends, and relationships in the data. The use of machine learning algorithms typically involves following a methodology pipeline, which includes several stages such as data gathering and analysis, feature extraction and transformation, modeling, and evaluation.

4 Descriptive modeling for virtual flow metering systems

4.1 Introduction

Accurately estimating multiphase flow rates is crucial for monitoring production processes and predicting field performance throughout the lifecycle of wells. Production testing is a routine test that is commonly conducted to monitor liquid rates and is considered the most popular method of monitoring production and reservoirs. However, it has limitations such as inadequate resolution to identify trends in liquid rates over short periods of time and insufficient testing time to obtain representative samples of reservoir fluids.

Multiphase flow meters (MPFMs) have been developed as a solution to the limitations of traditional production testing methods. They calculate flow rates without separating the phases and are cost-effective and easy to install at the wellhead. However, MPFMs can be expensive and require intervention in case of failure. Hence, they increase operational costs. They also have an operating range beyond which flow rate predictions may decline significantly and can deteriorate from sand erosion or partial obstruction, affecting measurement accuracy (G. Falcone et al. 2001; Gryzlov et al. 2020).

Virtual flow metering (VFM) technology has emerged as a promising solution in the oil and gas industry, considering the challenges and expenses associated with the traditional production testing and MPFMs. This technology employs analytical or data-driven models for real-time calculations of phase production. Its objective is to utilize readily available field data to estimate flow rates through modeling (Bikmukhametov and JÃ¤schke 2020; Rao and David 2015; Haouche, Adrien Tessier, et al. 2012a; Vinogradov and Vorobev 2020).

One of the main advantages of VFM is that they do not require the installation of additional hardware, which reduces both capital and operational costs during field development. Furthermore, VFM systems have the ability to estimate flow rates in real-time, which is an improvement over well testing

approaches that assume constant well flow rates between tests (Rao and David 2015). Additionally, VFM can be used as a standalone solution or in combination with an MPFM as a backup system, which utilizes information from an MPFM to further enhance flow rate estimates.

The classification of VFM is based on modeling paradigms, which are divided into mechanistic and data-driven modeling. Mechanistic modeling relies on physics-based models, whereas data-driven modeling utilizes statistical models. Both paradigms have their own strengths and limitations, and their concepts are briefly described. Additionally, a methodology in the area of VFM for electrical submersible pumped wells is presented, which has shown promising results in improving flow rate estimates.

4.2 Mechanistic and data-driven modeling

The mechanistic modeling approach of VFM systems relies on simulating the near-well region, the wells, pipelines, and production chokes (Ishak et al. 2020). These models are utilized in conjunction with measurements such as pressure and temperature to generate accurate flow rate estimates. An optimization algorithm is then applied to adjust the flow rates and other tuning parameters, aiming to minimize the discrepancies between the model predictions and actual measurements (Ishak et al. 2020). The production system can either be modeled as a whole from the reservoir to the processing facility, or separated into sub-models based on the available measurement data. Currently, most commercial Virtual Flow Metering systems employ first principles models.

The mechanistic VFM methods typically employ conservation equations. However, the problem formulation can take different forms, such as steady-state or quasi-steady-state. To state that more simply, the optimization solver finds a solution for a single point or uses the solution from the previous step as an initial guess for the current time-step prediction. In some cases, the conservation equations can take a steady-state form or disregard time altogether, such as in a choke model (Gryzlov et al. 2020). The dynamic formulation approach for VFM is infrequently utilized in the literature on mechanistic VFM, likely due to its high computational cost (Gioia Falcone 2009).

On the other hand, the data-driven VFM approach is based on collecting field data and fitting a mathematical model to it without an exact description of the physical parameters of the production system, such as wellbore and choke geometry and flowline wall thickness. This approach has become very popular in the past several years, not only for oil and gas applications but for

many other applications as well. If the model is well-trained and the exposed conditions are within the range used for the training, the data-driven model can perform fast and accurate real-time metering. In this approach, deep domain knowledge of production engineering is not as important as in the first principles models, and the model can be constructed at a lower cost.

For the large majority of data-driven algorithms used for VFM, the model formulation considers pressure and temperature measurements at one point in time to predict the flowrates at the same time step. In the next sections, Real data from a field are analyzed and handled through multiple machine learning algorithms in order to predict flow rate and water cut. Typically, the measurement data consist of pump intake pressure and discharge pressure (P_{intake} and $P_{discharge}$), wellhead pressure and temperature (WHP and WHT), choke opening (C_{op}), pump vibration in X and Y directions (V_x and V_y), variable speed drive current (I_{VSD}). The machine learning algorithms applied are symbolic regression, support vector regressor and deep learning algorithm (long short-term memory algorithm with convolutional neural network). They are arranged according to their complexity and ability to be explained.

4.3 Exploratory data analysis (EDA)

Exploratory data analysis involves utilizing various tools such as visualizations (scatter and box plots, histograms, etc.), dimensionality reduction techniques, and sometimes unsupervised learning algorithms to uncover hidden patterns, correlations, and anomalies. According to (James et al. 2013), the process begins with a univariate study that examines the dependent variables, which are flow rate and water cut. Next, a multivariate study is conducted to explore the relationship between the dependent and independent variables. Additionally, clustering and dimensionality reduction are employed to analyze categorical variables. Finally, basic cleaning is performed to address missing data and outliers.

The reported signals for ESP-pumped wells are categorized into four different groups. The first group is the wellhead data that include choke opening (%), wellhead pressure in (psig), flow line pressure in (psig), casing pressure in (psig), and wellhead temperature (°F). The second group is the electrical data that include frequency in Hz, input voltage of the three phases, and variable speed drive (VSD) output current of the three phases in (Amps). The third group is the operational downhole data that include pump intake pressure in (psia), pump discharge pressure in (psia), intake temperature T_i (°C), motor temperature T_m (°C), vibration in 2D (V_x, V_y), and differential

pressure in (psia). The fourth is the MPFM data, which are water cut in percent and produced fluid rate in (BPD).

The initial step in EDA usually involves a univariate analysis, which entails examining the individual variables in the dataset. In this case, the focus is on the dependent variables, which are fluid rate and water cut. The visual analysis employs density plots, which provide a smooth estimate of the probability density function for a continuous variable. Density plots were created to gain a better understanding of the distribution of the target variables, as depicted in figures 4.1 and 4.2.

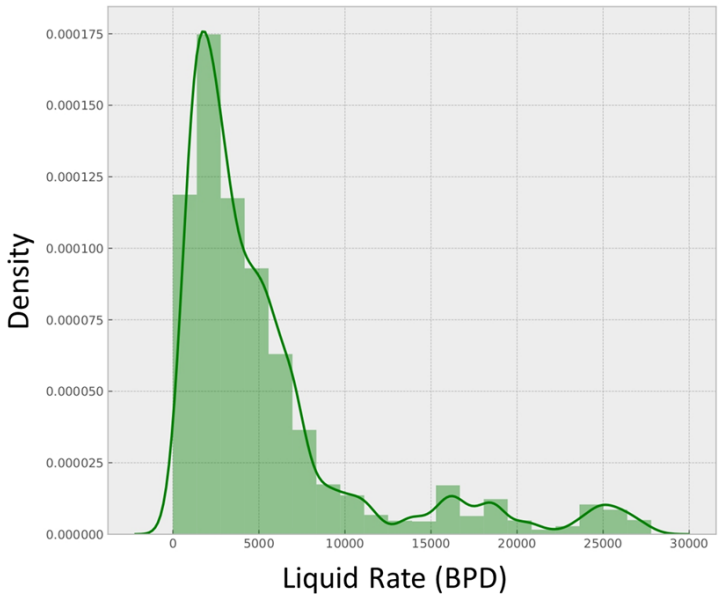


Figure 4.1: Density Plot for fluid rate

The rates appear to be skewed left, with some outliers above 15,000 BBLs. The same trend is observed for water cuts exceeding 60 percent. Calculation of skewness and kurtosis reveals a skewness of 2.09 and kurtosis of 3.98 for flow rates, and a skewness of 1.22 and a kurtosis of 0.90 for water cuts. Additionally, the dataset contains many zero-production points, indicating the presence of data related to non-producing periods.

Regarding the skewness of the flow rate and water cut, there are two solutions: either to remove the outliers using a specific technique or to use log transformation for those target variables, resulting in a normal distribution of the flow rate and water cut. This will help in enhancing modeling results. Regarding the zero-reported production points, dropping them is necessary as they are just noise data from sensors during the shut-in time. This will be done in the later step through pre-processing.

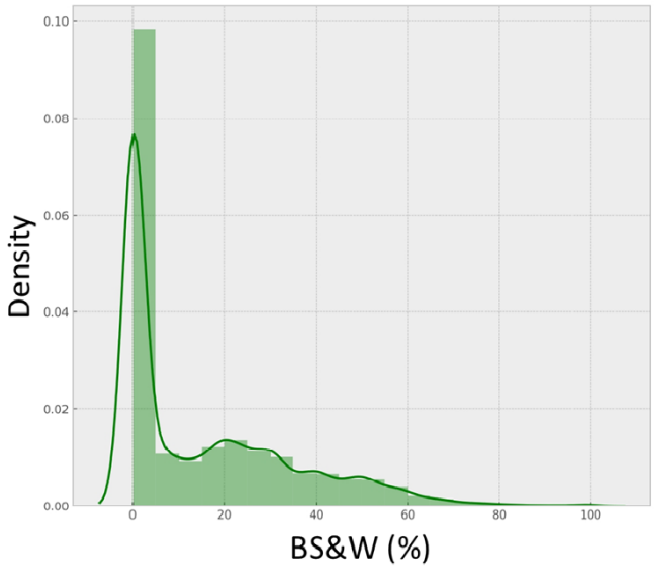


Figure 4.2: Density plot for water cut

The second step in EDA involves conducting a multivariate study to explore the relationship between the dependent and independent variables. Figures 4.3 and 4.4 show the histogram for each signal as a univariate figure and the correlation matrix for them as a multivariate analysis. In Figure 4.3, a pair plot visualizes the distributions of multiple signals, including flow rate, current from the variable speed drive, vibrations in the x and y directions, and choke opening. This type of graphical representation facilitates the simultaneous comparison of these variables. Upon detailed examination, it becomes evident that the distribution of the flow rate closely mirrors those of the current from the variable speed drive, vibrations in both x and y directions, and the choke opening. This similarity suggests shared patterns in the data distribution among these signals.

Pair plots are invaluable tools in data analysis, allowing for the observation of individual variable distributions and potential relationships between pairs of variables. Specifically, in this pair plot (Figure 4.3), the focus is on comparing the flow rate with other signals. The comparable distributions imply possible underlying connections or dependencies. For instance, fluctuations in flow rate might influence variations in the current from the variable speed drive, vibrations in the x and y directions, and the choke opening. Conversely, these signals might also impact the flow rate.

Additionally, figure 4.4 shows that flow rate has a high correlation with the current from the variable speed drive and choke opening, while basic sediments and water (BS&W) signal is more related to wellhead pressure

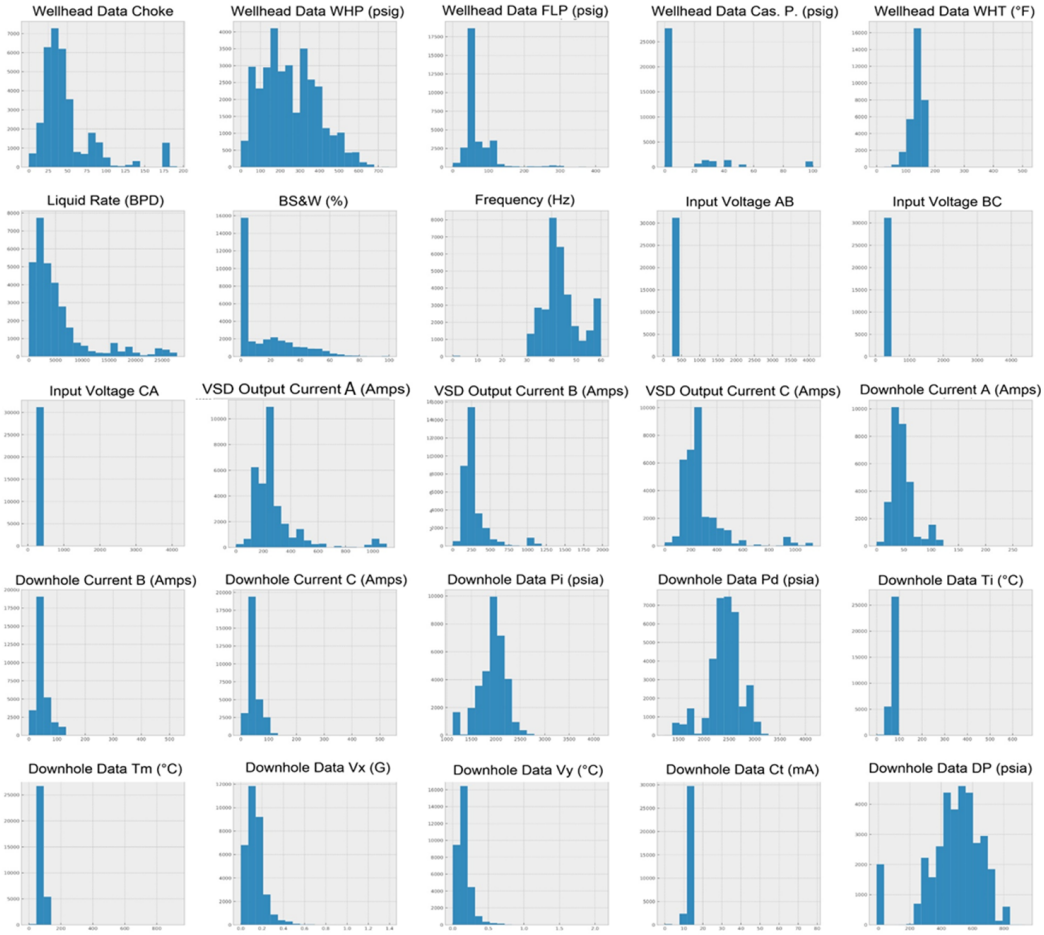


Figure 4.3: Pair plot histogram for signals

and has a slight inverse proportionality with current and flow rate.

The third step involves examining the relationship between flow rate and basic sediments and water with the categorical feature, which is the well name. The spread and skewness of the target variables data per well are demonstrated through Figures 4.5 and 4.6.

To ensure data accuracy, missing data points were removed from four signals, namely the liquid rate signal, variable speed drive output current, pump intake pressure, frequency, and discharge pressure. Zero production-reported sensor measurements do not make sense for these signals, and therefore, they were excluded from the analysis. The missing data for all signals are presented in table 4.1. To eliminate signals with a significant amount of missing data, a threshold is applied. Accordingly, the three-phase input voltage signals (AB, CA, and BC) are discarded since they display a low correlation coefficient with flow rate and basic sediment and water

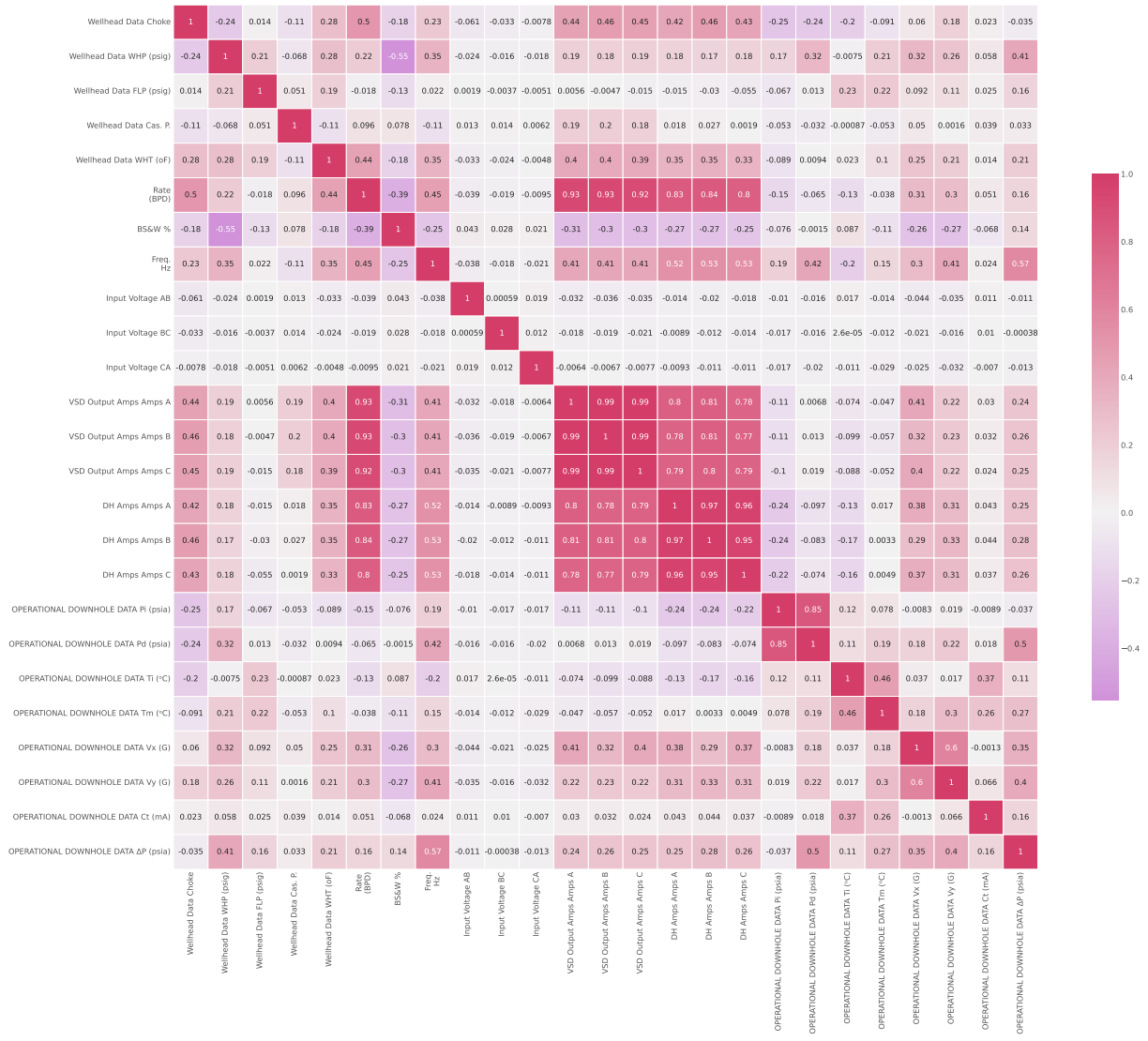


Figure 4.4: Correlation matrix for signals

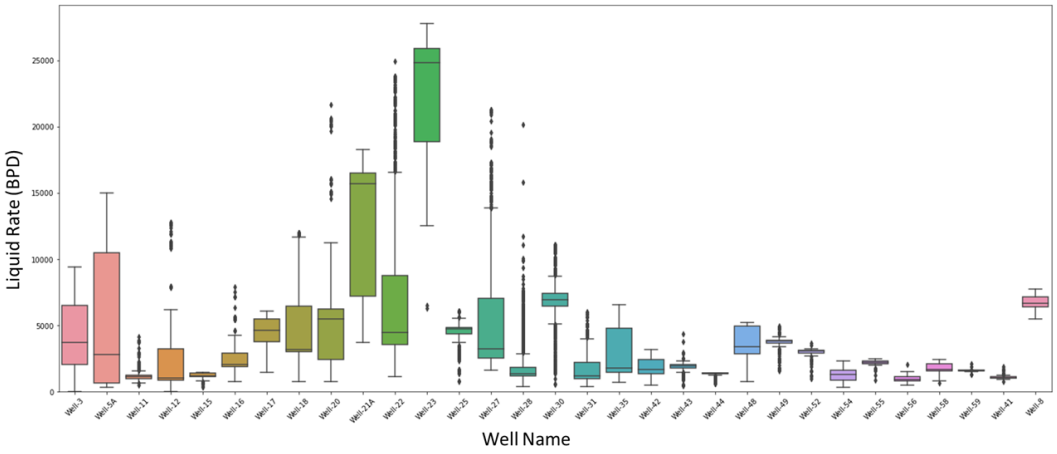


Figure 4.5: Boxplot of flow rate for each well

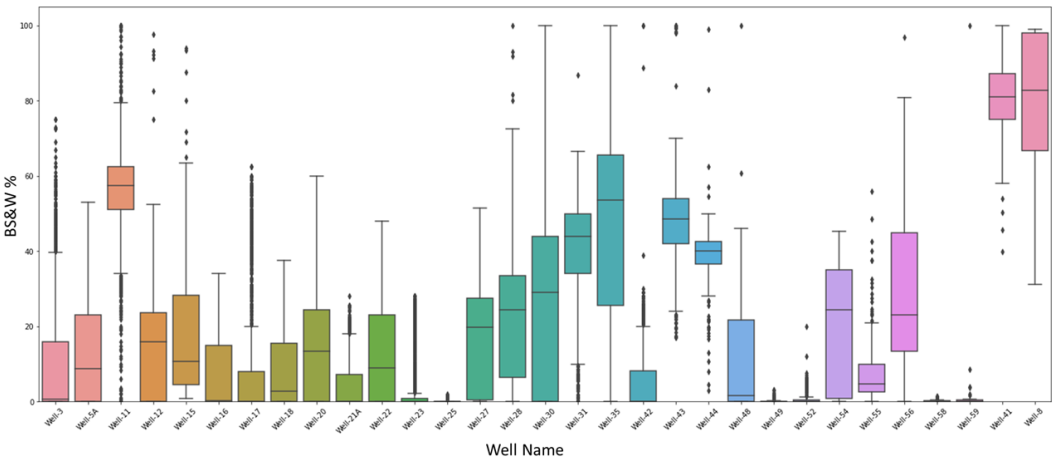
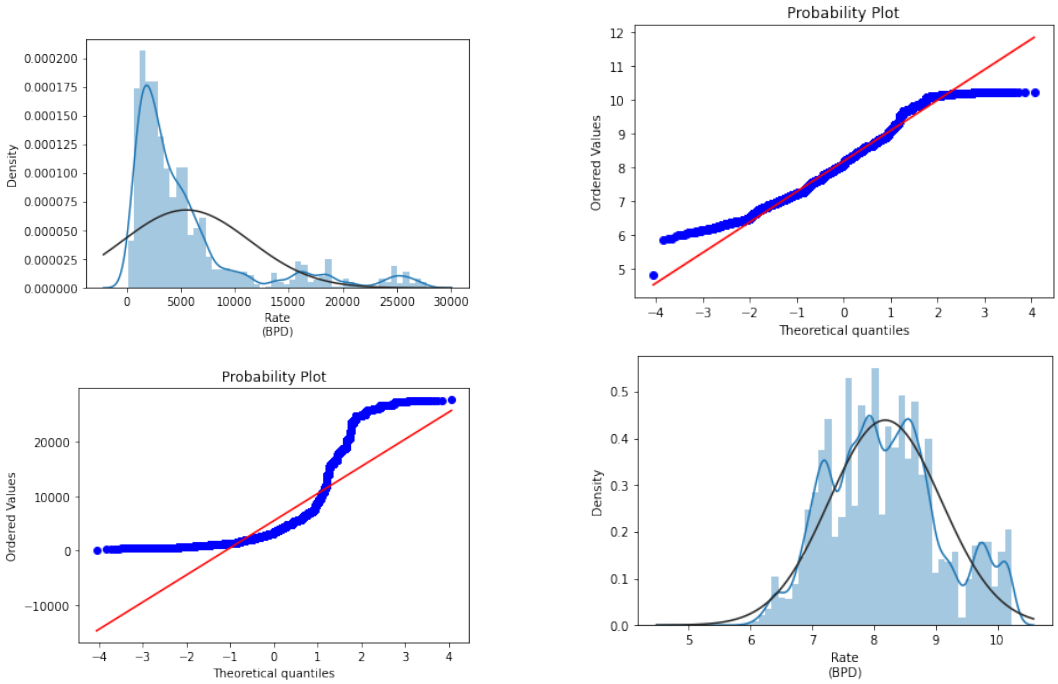


Figure 4.6: box plot of basic sediments and water for each well



a) Before Log transformation

b) After Log transformation

Figure 4.7: Probability plot and the distribution for flow rate

signals, as depicted in figure 4.4, and possess the highest number of missing values as demonstrated in table 4.1. The motor current signal is also excluded from the analysis since it is highly correlated with the recorded current of the variable speed drive and is one of the signals with the highest proportion of missing data.

As mentioned previously in the univariate study, the probability density distribution of the target flow rate is skewed from the normal distribution, as demonstrated in figure 4.7. To address this issue, a log transformation is applied to the flow rate sensor measurements. Log transformation is a commonly used method for addressing skewed data and is aimed at making the data more normal and improving the validity of the associated statistical modeling. Figure 4.7 illustrates the probability plot and distribution for flow rate before and after log transformation.

One important aspect of data exploration is the application of unsupervised learning techniques to uncover any underlying structure or grouping in the input data. In this regard, two widely used methods, namely principal component analysis (PCA) and K-means clustering, are investigated.

K-means is an unsupervised machine-learning algorithm that groups data points into distinct clusters based on their similarity. This is achieved by

Reported Signal	Total	Percent
Input Voltage AB	2098	0,067
Input Voltage CA	2098	0,067
Input Voltage BC	2098	0,067
DH Amps A	1469	0,047
DH Amps C	1468	0,047
DH Amps B	1468	0,047
T_m (°C)	1446	0,046
C_t (mA)	1444	0,046
V_x (G)	1443	0,046
T_i (°C)	1440	0,046
V_y (G)	1440	0,046
BS&W (%)	590	0,019
Wellhead Data FLP (psig)	341	0,011
Wellhead Data WHT (°F)	280	0
Wellhead Data Cas. P.	12	0
P_d (psia)	0	0
Wellhead Data Choke opening	0	0
VSD Output Amps Amps A	0	0
P_i (psia)	0	0
VSD Output Amps C	0	0
VSD Output Amps B	0	0
Wellhead Data WHP (psig)	0	0
Frequency (Hz)	0	0
Rate (BPD)	0	0
ΔP (psia)	0	0

Table 4.1: Missing Data Per Signal

partitioning the data into k groups, where k is the pre-specified number of clusters. The initial centroids are randomly selected, and each data point is assigned to the nearest centroid based on Euclidean distance. The centroids are then recalculated as the mean of all the data points assigned to that cluster, and this process is iterated until convergence is achieved. K-means is commonly used for data pre-processing, image segmentation, and exploratory data analysis.

PCA, on the other hand, is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional representation while retaining most of the variance in the original data. It does this by finding the principal components, which are linear combinations of the original variables that explain the maximum variance in the data. The first principal component explains the most variance, followed by the second, and so on. The principal components are uncorrelated with each other. PCA is commonly used for data visualization, feature extraction, noise reduction, and identifying patterns and relationships between variables. In summary, both K-means and PCA are useful tools for exploring and analyzing data in a more efficient and meaningful way.

Figures 4.8 and 4.9 show the projection of the input signals on the first two principal components. There is a slight clustering per well and a slight grading of the input signals with pump frequency, which means a slight enhancement in modeling can be made either by using well name or frequency as a categorical parameter.

The K-means algorithm has been also applied to the input signals to study the effect of grouping the data before modeling. Figure 4.10 depicts the results of the silhouette analysis of the k-means clustering algorithm. This analysis is used to measure the separation distance between resulting clusters and provides a visual assessment of parameters such as the number of clusters, with scores ranging from -1 to 1. A score of +1 indicates that a sample is far from neighboring clusters, while a score of 0 indicates that it is very close to the decision boundary between two clusters. Negative scores suggest that samples may have been assigned to the wrong cluster. The silhouette plot in figure 4.10 shows that the best average silhouette score, 0.28, was obtained for five clusters. Therefore, clustering is not recommended, and if attempted, it should be limited to five clusters only.

4.4 Feature permutation

Feature permutation importance measures the predictive value of a feature by evaluating the average prediction error for all models in which a specific

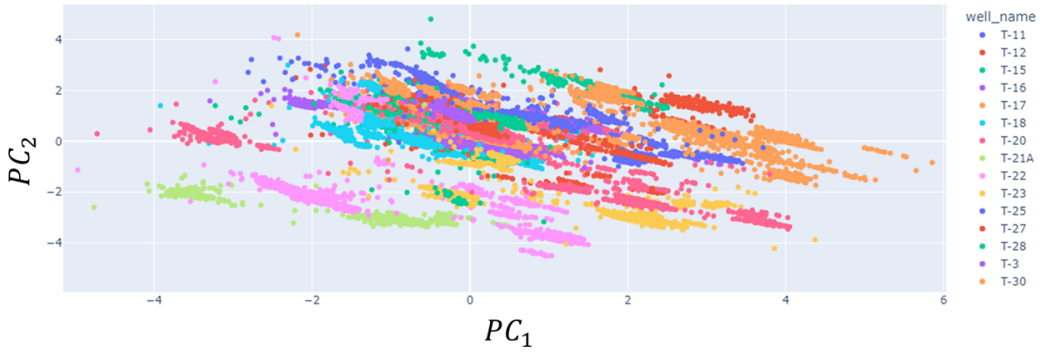


Figure 4.8: PCA of the input signals per well

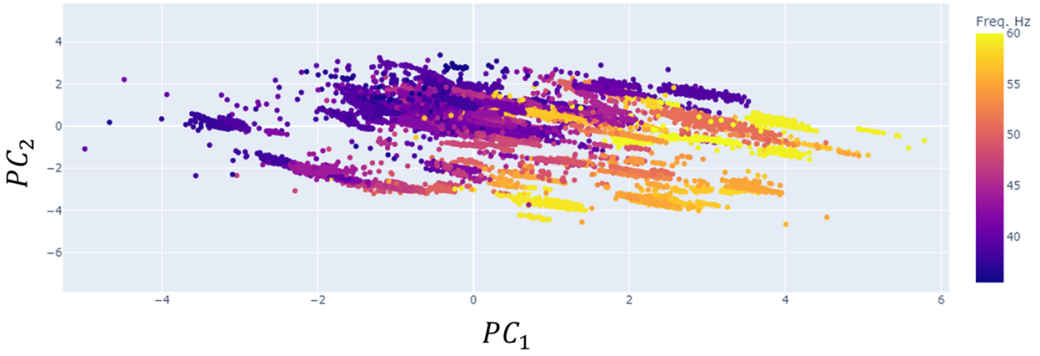
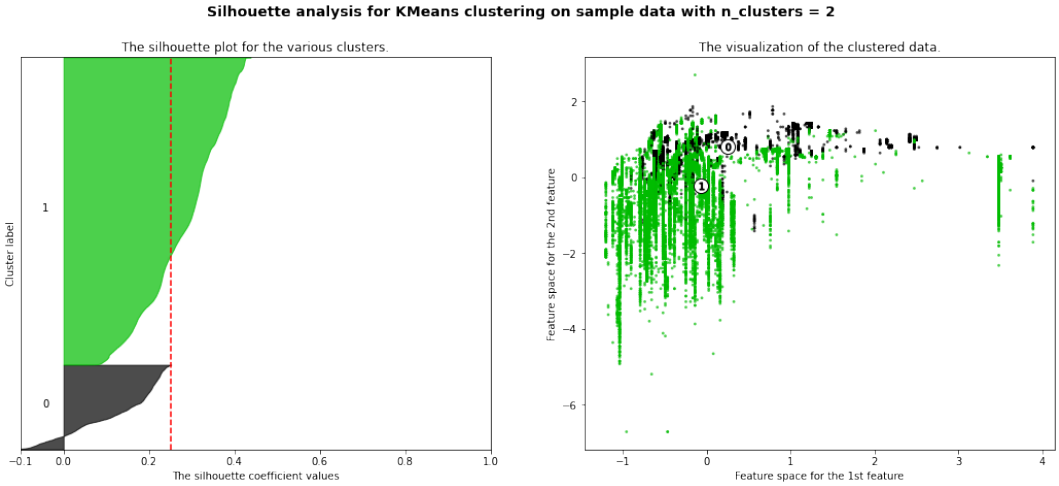
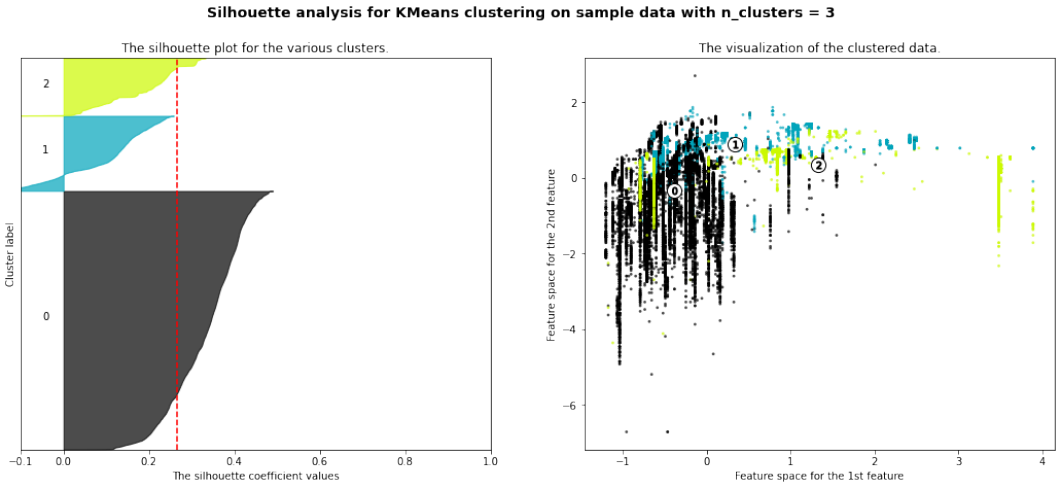


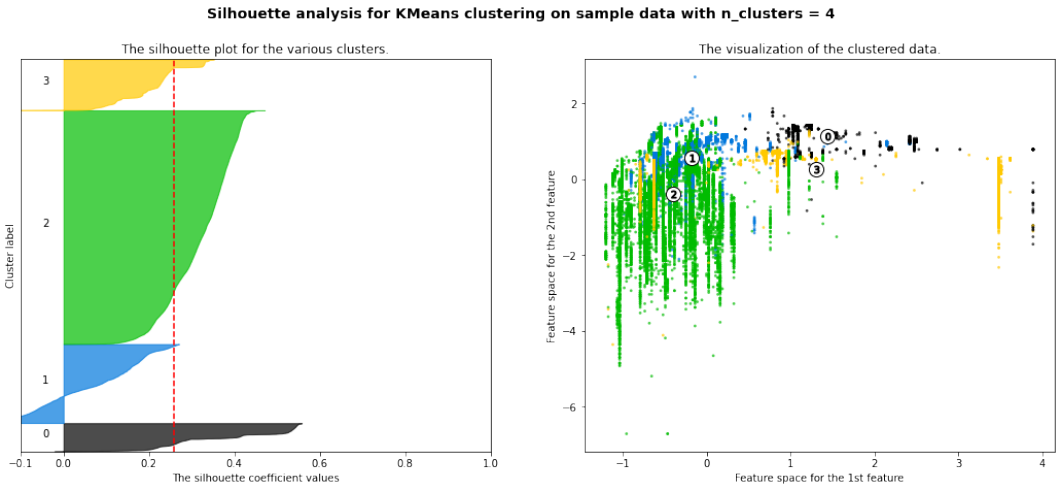
Figure 4.9: PCA of the input signals With Frequency



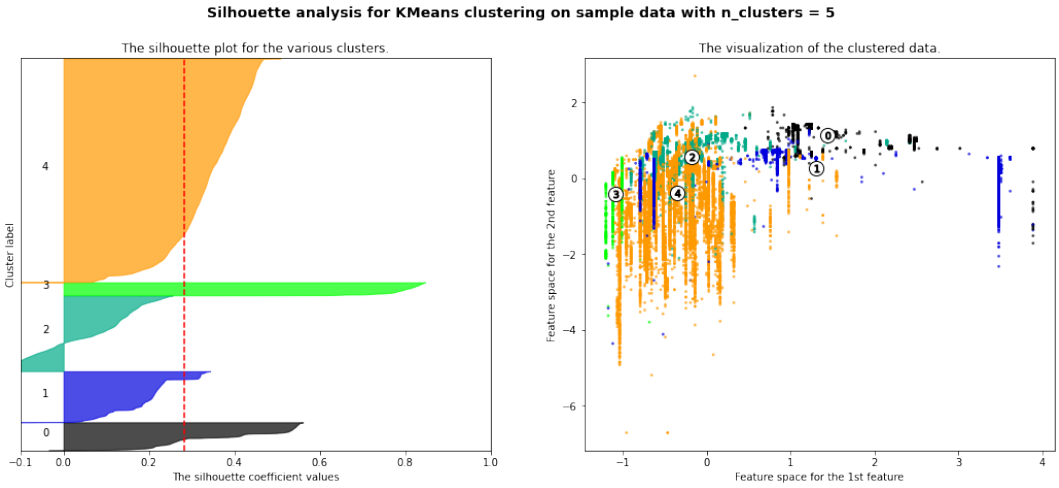
a) Silhouette Analysis for 2 clusters



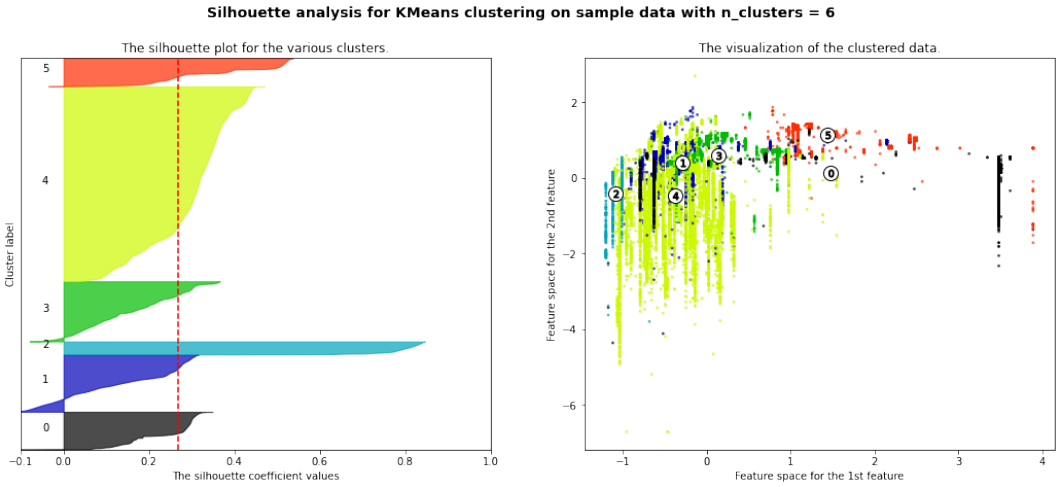
b) Silhouette Analysis for 3 clusters



c) Silhouette Analysis for 4 clusters



d) Silhouette Analysis for 5 clusters



e) Silhouette Analysis for 6 clusters

Figure 4.10: Silhouette Analysis from 2 to 6 clusters

input is incorporated. This approach allows for the ranking of features based on their relative importance to the model's predictions.

To assess the predictive value of each feature and its impact on the model's predictions, a custom type of feature permutation importance model was utilized. This model-agnostic method is useful in providing insights into the behavior of a machine-learning model. The steps to compute the custom feature permutation importance are as follows:

1. Split the dataset into training and testing sets.
2. Obtain all permutations of the input features.
3. Shuffle the feature column in the dataset randomly.
4. Train each model on the same training set.
5. Calculate the prediction error on the test dataset.
6. Record the input features and the training and testing results.
7. Repeat steps 3 to 6 multiple times and save the results in an experiment data frame.
8. For each input feature, calculate the average error reported for all the models it has contributed to, to mitigate the effects of random shuffling.
9. For each input feature, the average error reported for all models it has contributed to. This averaging is to mitigate the effects of random shuffling.
10. Rank the features based on the average impact each feature has on the model's score, with the features that have contributed to the smallest error models being assigned higher importance.

Tables 4.2 and 4.3 show the signals recommended as input features for flow rate and water cut, respectively, based on the feature rotation random search. The presented information pertains to a comparative analysis of signals, their respective names, and the average Mean absolute error (MAE) values corresponding to the models that utilize this signal. In scientific terms, the analysis provides insight into the performance of different models in predicting outcomes based on various input signals.

4.5 Algorithms

Typically, machine learning models perform differently depending on the data they are given. Therefore, there is no universally accepted standard algorithm that can be applied to all datasets. The best model can only be chosen after analyzing the data and comparing different machine learning algorithms. In this section, the algorithms used in this research are explained. These algorithms include symbolic regression, support vector regressor

Table 4.2: Feature Rotation for flowrate

Signal	Average MAE
VSD Output Amps (A)	0,039
OPERATIONAL DOWNHOLE DATA Pd (psia)	0,040
OPERATIONAL DOWNHOLE DATA Pi (psia)	0,041
Frequency (Hz)	0,041
Wellhead Data Choke	0,041
Wellhead Data WHP (psig)	0,041
Wellhead Data FLP (psig)	0,041
OPERATIONAL DOWNHOLE DATA Tm ($^{\circ}C$)	0,041
Wellhead Data Cas. P.	0,041
OPERATIONAL DOWNHOLE DATA ΔP (psia)	0,041
Wellhead Data WHT ($^{\circ}F$)	0,041
OPERATIONAL DOWNHOLE DATA Ti ($^{\circ}C$)	0,041
OPERATIONAL DOWNHOLE DATA Vy (G)	0,042

Table 4.3: Feature Rotation for BS&W

Signal	Average MAE
VSD Output Amps A	2,16
OPERATIONAL DOWNHOLE DATA ΔP (psia)	2,20
Wellhead Data WHP (psig)	2,21
OPERATIONAL DOWNHOLE DATA P_d (psia)	2,21
Wellhead Data Choke	2,22
OPERATIONAL DOWNHOLE DATA P_i (psia)	2,24
OPERATIONAL DOWNHOLE DATA T_m ($^{\circ}C$)	2,31
OPERATIONAL DOWNHOLE DATA V_y (G)	2,31
OPERATIONAL DOWNHOLE DATA T_i ($^{\circ}C$)	2,35
Wellhead Data Cas. P.	2,38
Wellhead Data FLP (psig)	3,40
Freq. Hz	3,41
Wellhead Data WHT ($^{\circ}F$)	4,41

(SVR), and 1D-CNN with LSTM, which are deep learning algorithms used to predict flow rates and BS&W.

4.5.1 Symbolic regression

Symbolic regression can be considered a special type of regression analysis. This algorithm searches for mathematical expressions that best fit a given dataset, which consists of sensor data as input and produced rates as output. It is a form of genetic programming. The algorithm is given a grammar that defines basic functional building blocks, such as addition, subtraction, multiplication, logarithms, trigonometric functions, etc. The algorithm then tries different combinations in an evolutionary process that retains the better terms and recombines them to create even more fitting terms. In the end, a good formula is produced that captures the dynamics of the system without overfitting the noise.

Unlike traditional regression methods, symbolic regression can determine both parameters and structures of regression models simultaneously (Awange et al. 2018). In traditional numerical regression, the functional form is predefined to be linear, polynomial, or nonlinear, and the task is to determine the coefficients in the functional form. In symbolic regression, the task is to automatically find a suitable functional form in complex data, either linear or nonlinear and simultaneously determine the coefficients of the functions.

Symbolic regression is a technique based on Darwin's theory of evolution, where the competitive mechanism ensures that individuals with superior performance have a greater chance of survival while poor-performing individuals are gradually removed. The process involves three genetic operators: selection, crossover, and mutation, as illustrated in Figure 4.11. When a superior gene appears in some individuals, it is selected, duplicated, and spread across the population. The gene's contribution to the fitness of the model determines whether it remains in an individual during the evolutionary process. This mechanism ensures that only important genes are selected to form the models gradually, similar to the survival mechanism in Darwin's theory of evolution. The emergence of a superior gene helps identify the significant factors contributing to the functions found by symbolic regression. In other words, each factor's occurrence reflects its ability to describe the data, with higher frequency indicating greater importance.

Symbolic regression encodes candidate solutions as trees, with terminal nodes representing constants and variables, and intermediate nodes encoding mathematical functions such as addition, subtraction, multiplication,

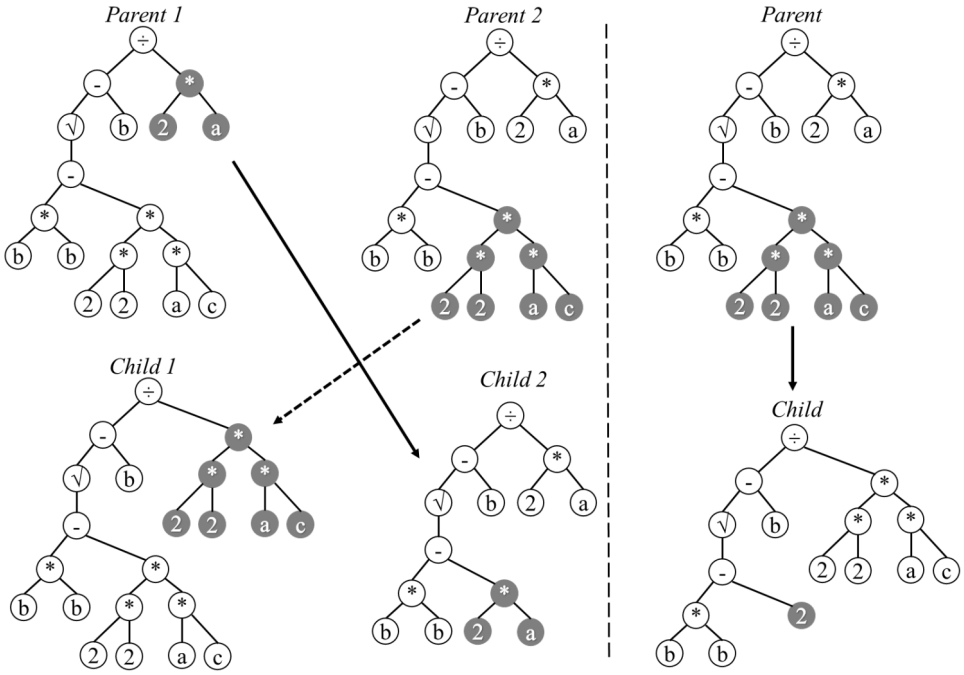


Figure 4.11: (a) Crossover, (b) mutation genetic operations in the symbolic regression (revised from Cui et al. 2020)

and division. All nodes are collectively referred to as "building blocks". The fitness function is typically proportional to the absolute or squared error between experimental data and the candidate solution's predicted values, with parsimony corrections favoring more concise equations.

One of the major advantages of symbolic regression is that it can automatically discover models, but this comes at the cost of a vast search space and significant computing resources. However, SR does offer the benefit of delivering human-readable models, unlike models produced by neural networks and support vector machines. Equations can always be analyzed, even if they are highly complex, by a human expert. The results of the automatically generated models can be used to infer properties of the target phenomenon and serve as a foundation for building improved models.

4.5.2 Deep Learning (LSTM and CNN)

For massive data and high computational performance, deep learning models have shown recently great advantages in automatically extracting and learning multivariate data features. In particular, the Recurrent neural network (RNN) and its various improved algorithms have been applied to feature extraction of time series data. In this study, a general framework

for multivariate prediction problems is proposed, including Convolutional neural network (CNN) and Long short term memory algorithm (LSTM).

There are two primary components in this model. As seen in Fig.4.12, CNN is used to extract the lateral characteristics of multidimensional data, while LSTM is used to recover the temporal features. To clarify, CNN is used for extracting horizontal relationships of multidimensional variables, and LSTM is used to learn timing relationships and make predictions based on the features obtained by CNN. Such a network architecture brings together the advantage of time series processing by using LSTM and signal processing for the data that come from pump sensors by using a 1D convolutional network.

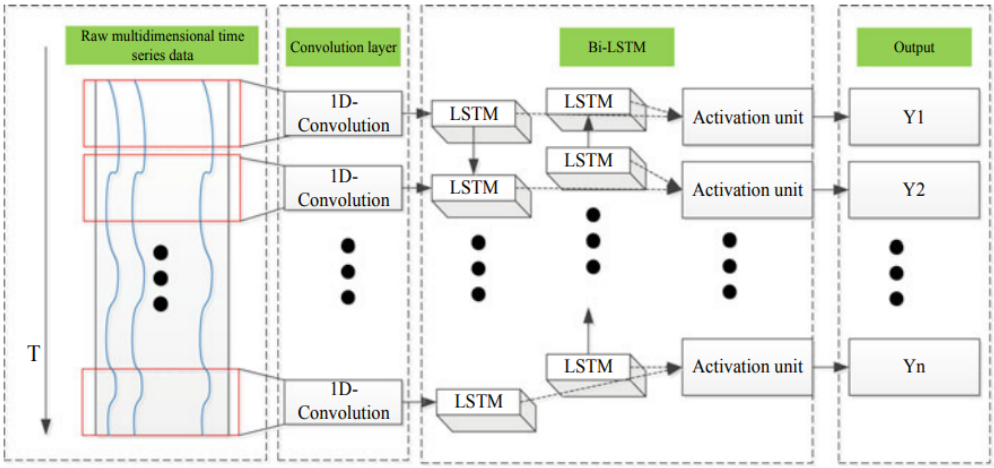


Figure 4.12: Model framework

Convolutional Neural Networks (CNN)

In general, conventional neural networks are designed to process 2D data such as images and videos. Convolutional Neural Networks (CNNs) are specifically designed to operate on images, and each neuron in a CNN contains 2-D planes for weights (known as the kernel) and input/output (known as the feature map). CNN architecture typically includes three main types of layers: convolutional layers, pooling layers, and fully connected layers.

A convolutional layer is the fundamental building block of a CNN, consisting of a set of filters (or kernels) whose parameters are learned during training. The filter size is usually smaller than the actual image, and each filter convolves with the image to create an activation map. During convolution, the filter slides across the height and width of the image, and the dot product

between every element of the filter and the input is calculated at each spatial position.

The second type of layer is a pooling layer, which serves as a downsampling or dimensionality reduction layer. It is used to reduce the computational complexity of the network by decreasing the number of eigenvectors of the convolutional layer output, while also extracting the key features and performing feature compression (Theodoridis 2020).

Finally, the fully connected layer performs the task of classification based on the features extracted by the previous layers and their respective filters. In the fully connected layer, each node in the output layer is directly connected to a node in the previous layer, enabling the network to model the input signals based on the extracted features (S. Kiranyaz, Avci, et al. 2021a; A. Popa et al. 2022; Howard et al. 2017; Kim 2014; Defferrard et al. 2016).

Convolutional neural networks (CNN) have a unique architecture that allows them to combine feature extraction and modeling into a single learning process. This makes them very useful for processing sensor signals. One of the advantages of CNN is their ability to handle small transformations in input data, such as translation, scaling, skewing, and distortion. As a result, many recent applications of time series data analysis and signal processing have utilized one-dimensional (1D) convolutional neural networks.

1D CNNs can be considered a modified version of 2D CNNs. They are more advantageous over 2D CNNs in dealing with signals and sensor data due to a main reason which is the lower computational complexity of 1D CNN. A prevailing trend observed in recent studies is that the majority of 1D CNN applications have employed concise architectures, typically with one or two hidden CNN layers (S. Kiranyaz, Ince, Hamila, et al. 2015; S. Kiranyaz, Ince, and Gabbouj 2016; S. Kiranyaz, Ince, and Gabbouj 2017; Avci et al. 2018; Avc1, Abdeljaber, S. Kiranyaz, et al. 2017; Abdeljaber, Avc1, S. Kiranyaz, et al. 2017; Avc1, Abdeljaber, M. S. Kiranyaz, et al. 2018; Abdeljaber, Avc1, M. S. Kiranyaz, et al. 2018; Ince et al. 2016; S. Kiranyaz, Gastli, et al. 2019; Abdeljaber, Sassi, et al. 2019; Eren et al. 2019; Eren 2017). Conversely, nearly all 2D CNN applications have utilized more complex and extensive architectures. In summary, shallow networks are simpler to train and use, while compact 1D CNNs are ideal for low-cost and real-time applications due to their low computational demands. The one-dimensional CNN model is illustrated in Figure 4.13.

In each CNN layer, 1D forward propagation (1D-FP) is expressed as follows:

$$X_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv1D(W_{ik}^{l-1}.S_i^{l-1}) \tag{4.1}$$

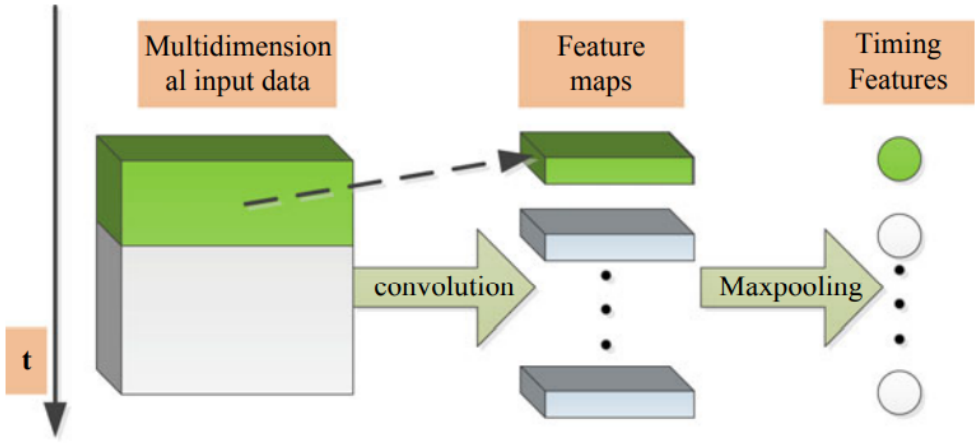


Figure 4.13: One-dimensional convolution

where:

X_k^l is the output of the K^{th} neuron at layer l.
b_k^l is defined as the bias of the K^{th} neuron at layer l.
S_{ik}^{l-1} is the input to the K^{th} neuron at layer l from the i^{th} neuron at layer l-1.
W_{ik}^{l-1} represents the weight connecting the output of the i^{th} neuron in layer l-1 to the K^{th} neuron in layer l.
$conv1D(...)$ is used to perform 1D convolution without zero-padding.

The output, y_k^l. can be expressed by passing the input X_k^l through the activation function, $f(.)$ as given in 4.2

$$y_k^l = f(X_k^l) \tag{4.2}$$

The backpropagation algorithm is a widely used method in neural network training, which starts by propagating errors from the output layer ($l = L$) and moves backward through the network to adjust the weights and biases of the neurons. In the output layer, the mean squared error (MSE) is a common measure of the error and can be calculated using equation 4.3:

$$E_p = \frac{1}{2}(y^L - t^p)^2 \tag{4.3}$$

Here, y^L represents the output of the output neuron, and t^p is the target output for the input vector p. In regression problems, the goal of the back-

propagation algorithm is to minimize the Mean squared error (MSE) by adjusting the weights and biases of the neurons through gradient descent.

Long short memory algorithm (LSTM)

LSTM is a subfield in deep learning that aims to overcome the problem of sequential data, as well as the issues of gradient disappearance and explosion that arise in the RNN network. To achieve this, the LSTM architecture uses cells as the information storage module, which helps to maintain the long-term memory of the sequence data.

At the core of the LSTM architecture is the memory cell, which has internal states used by the architecture to ensure consistent error flow. To update the cell state, the LSTM uses three gating units that have distinct functions and calculation methods. These gating units include an input gate, an output gate, and a forget gate, as shown in Figure 4.14. Together, they enable the LSTM to maintain long-term memory and overcome the limitations of the RNN network.

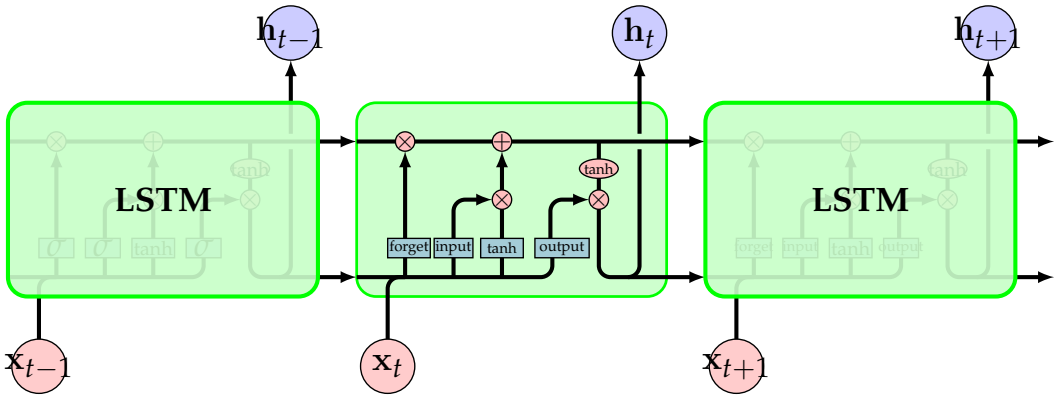


Figure 4.14: Structure of LSTM cells

The input gate in LSTM has two components. Firstly, it selects the new data to be stored in the memory cell. Then, a tanh layer creates a vector of new candidate values that will be added to the state.

The forget gate in LSTM uses the sigmoid function to generate a value between 0 and 1, which indicates how much knowledge of the prior hidden state and the present input should be retained.

The output gate in LSTM applies the sigmoid function to the prior hidden state and current input, multiplies it by the tanh function applied to the new memory cell, and provides values between -1 and 1 for the memory cell's

output. Equations (4.4), (4.5) and (4.6) present the equations for the input, forget, and output gates in LSTM, respectively.

$$i_t = \sigma\left(w_i[h^{t-1}, x^t] + b_i\right) \tag{4.4}$$

$$f_t = \sigma(w_f[h^{t-1}, x^t] + b_f) \tag{4.5}$$

$$o_t = \sigma(w_o[h^{t-1}, x^t] + b_o) \tag{4.6}$$

Where:
i_t = represents the input gate.
f_t = represents the forget gate.
o_t = represents the output gate.
σ = represents sigmoid function.
W_x= weight for the respective gate (x) neurons.
H^{t-1} = output of the previous lstm block (at timestamp t-1)
X^t =input at current timestamp.

The equations for the cell state, candidate cell state and the final output:

$$\tilde{C}^t = tanh(wc[ht-1, xt] + bc) \tag{4.7}$$

$$C^t = F_t \times C^{t-1} + I_t \times \tilde{C}^t \tag{4.8}$$

$$H_t = o_t \times tanh(C^t) \tag{4.9}$$

Where:
Ct = cell state (memory) at timestamp (t).
$\tilde{C}^t$ = represents a candidate for cell state at timestamp (t).

The gate units of the LSTM are crucial becuse of their ability to overcome the vanishing gradient problem that is often found in traditional RNNs. It enables the storage of long-term dependencies of sequential data.

To conclude, LSTM is a combination of an input gate, a forget gate and an output gate. Its unique memory cell stores and retrieves information from sequential data. LSTM is well-suited for tasks that require the preservation of long-term information.

4.6 Experiments

In this section, the hyperparameters for evaluation, including learning rate, batch size, and number of hidden layers, are specified along with their corresponding ranges. These hyperparameters are used to train and evaluate the machine learning models on the electrical submersible pump dataset, and the resulting performance metrics are analyzed. The ranges for the hyperparameters have been chosen based on previous research in the field of virtual sensing.

The goal of this regression is to find the best mathematical expressions that accurately approximate the production rate and BS&W of an electrical submersible pump. For the present regression, a function symbol set is considered. It includes addition, subtraction, multiplication, division, power, minimum, constant, integer constant, and input variable operations $(+, -, *, /, pow, min, c, respectively)$.

The training and testing data sets are normalized, and the mean absolute error (MAE) is used as the fitness function to evaluate the performance of each candidate solution. Hyperparameters that need to be tuned for this task include the maximum tree depth, population size, mutation rate, crossover rate, and number of generations. A preliminary tuning using a random search was conducted, and the selected hyperparameters are shown in table 4.4.

To evaluate the performance of the symbolic regression models, the best solutions obtained from the genetic programming search are applied to a separate testing data set. The performance metrics used to evaluate the models include the MAE, root mean square error (RMSE), and coefficient of determination (R^2).

Table 4.4: Symbolic Regression Hyperparameter

Hyperparameter	Value
Binary Operators	$(+, -, *, /, pow)$
Unary Operators	$(exp, sin, cos, sqrt)$
Loss	MAE
Population Size	5000
Number of Generations	50
Mutation Rate	0.1
Crossover Rate	0.9
Maximum Tree Depth	6

For deep learning applications, there are a lot of hyperparameters. Some of the variables that determine the network structure (e.g., the number of

hidden units, the size, and type of the network layers). On the other hand, some variables determine how the network is trained (e.g., learning rate, batch size).

In this study, recommendations on the structure made by (S. Kiranyaz, Avci, et al. 2021b) have been followed. The configuration of the 1D CNN-LSTM used in all experiments has two hidden convolutional layers, followed by an average pooling layer that down-samples the input representation by taking the average value over a window of five. The 1D CNNs have 32 and 16 neurons on the first and second hidden convolutional layers, while the two LSTM layers that followed that average pooling had 32 neurons followed by a dropout of 0.25. Then, two dense layers are added, with 10 neurons on the hidden dense layer. Finally, the output layer size is 2, which is the BS&W and liquid rate.

For all experiments, the maximum number of backpropagation iterations is set to 150, and another stopping criterion is the minimum train Huber error level that is set to 1% to prevent over-fitting. Therefore, the training will terminate if either of the criteria is met. Initially, the learning factor, e, is set at 10^{-3} and global learning rate adaptation is performed during each BP iteration, as follows: if the train MSE decreases in the current iteration, e is slightly increased by 5%; otherwise, it is reduced by 30%, for the next epoch. In addition, all models are trained using the Adam algorithm, which optimizes a predetermined loss function to obtain a final model.

Regarding the loss function used in this study, both the mean squared error and the Huber function were considered. However, the reported model was trained using only the Huber function as it was found to result in more stable models than the mean squared error. This result was expected as the Huber function is commonly used in robust regressions due to its lower sensitivity to outliers in the data compared to the squared error loss. The Huber function used in this study is shown in Equation 4.10.

$$L_\delta(a, y) = \begin{cases} \frac{1}{2}(y-a)^2, & |y-a| \leq \delta, \\ \delta\,|y-a| - \frac{1}{2}\delta^2 & \text{Otherwise} \end{cases} \tag{4.10}$$

4.7 Summary

Accurate estimation of multiphase flow rates is essential for monitoring and improving production processes in the oil and gas industry. Traditionally, well flow rates are estimated by directing the stream into a test separator,

which splits it into oil, gas, and water. However, this method has limitations in identifying production trends. An alternative solution is the use of multiphase flow meters (MPFMs), which estimate production rates by measuring a single physical property of the stream and relating it to different fluid rates. While MPFMs provide more accurate measurements, they are expensive to install and maintain, making them a less viable option for some operations.

Virtual flow meters (VFM) are a promising alternative to MPFMs and production testing. This technology relies on machine learning and physics-based transient modeling to estimate flow rates without the need for physical separators or flow meters. VFMs use either analytical or data-driven models to make real-time calculations of production phases. This approach has gained attention in the oil and gas industry as it offers a cost-effective and accurate way to estimate multiphase flow rates.

This chapter introduces a new data-driven model to calculate multiphase flow rates using sensor measurements from electrical submersible pumps. Detailed exploratory data analysis is conducted, and features are prioritized through experiments to identify the most dominant parameters that affect rate prediction. Finally, models are compared using mean absolute error, mean squared error, and R squared

In summary, this study uses real-time data from electrical submersible pumps and employs two different machine-learning approaches, symbolic regression and deep learning. Specifically, symbolic regression and a convolutional neural network (CNN) pipeline with a long short-term memory (LSTM) algorithm are implemented to predict the liquid rate and water cut based on sensor data. These methods provide a cost-effective and accurate way to estimate flow rates and identify trends, which can improve production processes and reduce maintenance costs.

5 Predictive maintenance for electrical submersible pumps

5.1 Introduction

As with any pumping artificial lift method, Electrical Submersible Pumps (ESPs) are prone to failures. Maintaining ESPs requires significant resources and manpower, and typically involves reactive monitoring of multivariate sensor data to detect any potential issues. In order to facilitate predictive maintenance and avoid operation downtime, machine learning techniques can be utilized to understand the equipment status.

ESPs are installed in many producing wells that are subject to harsh environments and need to pump complex fluid mixtures, which undergo changes in composition, pressure, and temperature over time. Timely interventions are required to ensure reliable fluid delivery in the face of these challenges. To this end, the field of "digital oil field" has emerged, focusing on deploying machine learning and data-driven models for predictive pump maintenance of ESPs.

This chapter presents a contribution to the area of increasing efficiency in the maintenance of ESPs. Specifically, it aims to reduce the time required for dismantling the system, inspecting it, and performing failure analysis. To achieve this objective, the proposed approach involves applying Principal Component Analysis (PCA) as a dimensionality reduction technique, followed by pipelining its output with a model trained using the XGBoosting algorithm for future prediction of pre-failure events in the system.

In addition to the aforementioned approach, this chapter explores the application of deep learning algorithms such as 1D convolutional neural networks and LSTM, as well as LSTM with attention, in addressing this problem. The results obtained from these approaches are compared to those obtained from the proposed approach, in order to assess their respective efficacy.

The proposed model is capable of identifying deeper functional relationships and longer-term trends inferred from historical data. The novel workflow,

along with the predictive model, can provide alarms up to seven days before an actual failure event occurs.

5.2 Data gathering and Pre-Processing

Time series data is collected from sensors on ESP wells. The reported measurements are Pump frequency (FRQ), Pump discharge pressure (PDP), Pump intake pressure (PIP), WHP, WHT, Motor temperature (MT), Casing head pressure (CHP), and Variable speed drive output current (Current), which have different frequencies. Additionally, well status sheets for the same wells at the same time periods are gathered on a daily basis. These data come from a field with a polymer flooding project. Based on the status sheets, pumps exhibit two main categories of problems: Motor downhole failures (MDHF) and Electrical downhole failures (EDHF), both of which are related to electrical pump failures.

Electric failures of the downhole facilities, including the electric cable, motor electrical components such as the stator, and downhole sensor, are referred to as MDHF and EDHF. Failures associated with electric cables are mainly caused by cable insulation due to corrosion, material failure, or cable failure due to overload. Meanwhile, electrical failures associated with the motor are usually the result of stator failure. The stator has been reported to fail due to overheating. As the motor is the hottest point in the well, this appears to worsen polymer deposition on the motor body, reducing heat dissipation and leading to increasing motor winding temperature. This in turn makes the deposition worse and eventually leads to a ramp down of ESP frequency when the maximum motor temperatures are reached. Likewise, the high temperatures around the motor aid in the precipitation of solid polymer in fluids flowing past the motor and are the source of polymer plugging in the pump inlet.

The workflow for predictive modeling starts with the data-cleaning process. It is important to eliminate nonphysical values (e.g., negative or enormous pressure values), remove further outliers, and align units. Additionally, it is a critical step for handling noisy data while maintaining the realistic anomalies that may identify downhole problems of the pumps.

After visual inspection of the data, a pipeline of a pre-processing strategy is created. Firstly, it starts by resampling the data using a moving median in one-hour steps. Fig 5.1 shows the boxplot of the data after resampling and before outlier removal. It is obvious that some measurements include unreasonable values. For example, well head temperature reaches 18500 $^{\circ}$F, which is an obvious error measurement. Therefore, the second step is

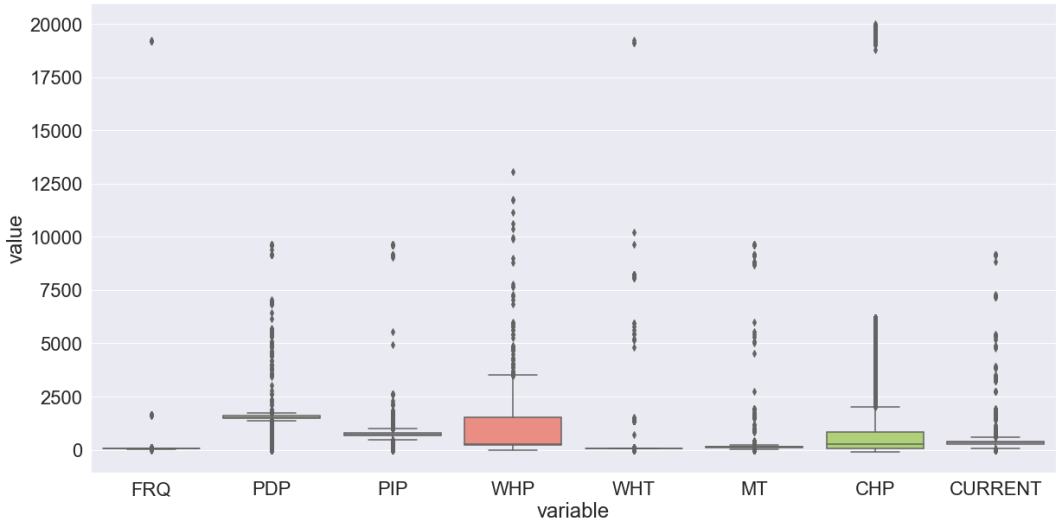


Figure 5.1: Box-plot before outlier removal

removing outliers, which includes first removing measurements where oil production is zero and then removing outliers by limits.

Outlier removal by limits depends mainly on quartiles; therefore, boxplots were used to summarize sample data using the 25th, 50th, and 75th quartiles. The mid-spread or middle 50th, also known as the H-spread or interquartile range (IQR), is a measure of statistical dispersion, equal to the difference between the 75th and 25th percentiles. It is similar to the Z-score in terms of finding the distribution of data and keeping a threshold to identify outliers. To define the outlier base value, the upper and lower bounds were defined above and below the dataset's normal range. The upper and lower bounds are calculated using equations 5.1 and 5.2.

$$upper = Q3 + 1.5 * IQR \tag{5.1}$$

$$lower = Q1–1.5 * IQR \tag{5.2}$$

Afterwards, a standard scaler is employed (subtracting mean from each point and dividing by the variance) transforming the mean value to zero and scaling the data to unit variance. Finally, the moving difference is applied on all sensors measurements. Fig. 5.2 shows the box plot after outliers removal. Fig. 5.3 shows the box plot after normalization.

Table 5.1 describes the main signals after outliers and zero production points are removed without standardization. Table 5.2 shows the number of available data points after mapping the sensor data. These data points are classified, based on workover sheets, into "Normal data", "Pre-workover", and "Workover". Pre-workover data are data points that are reported seven

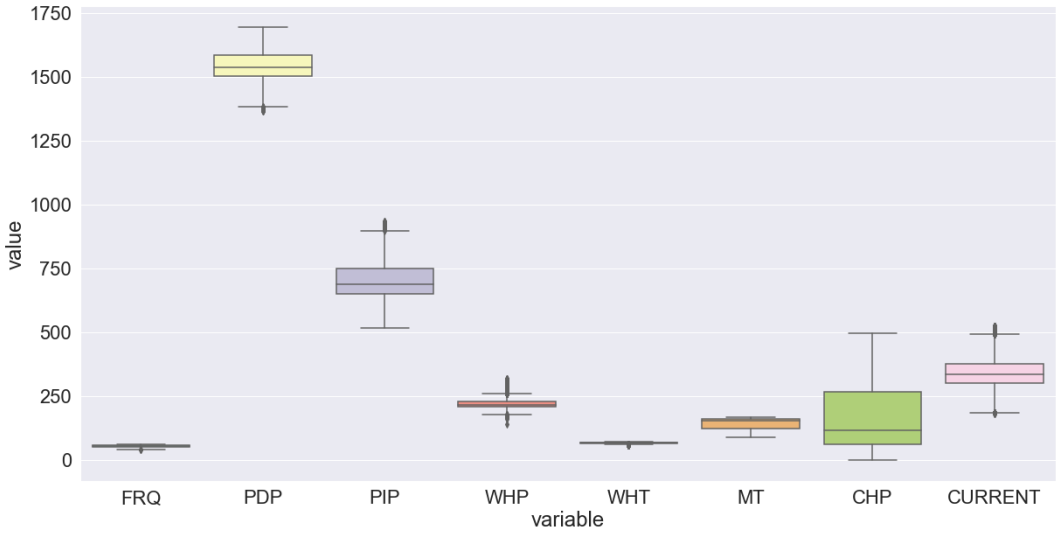


Figure 5.2: Box-plot after outlier removal

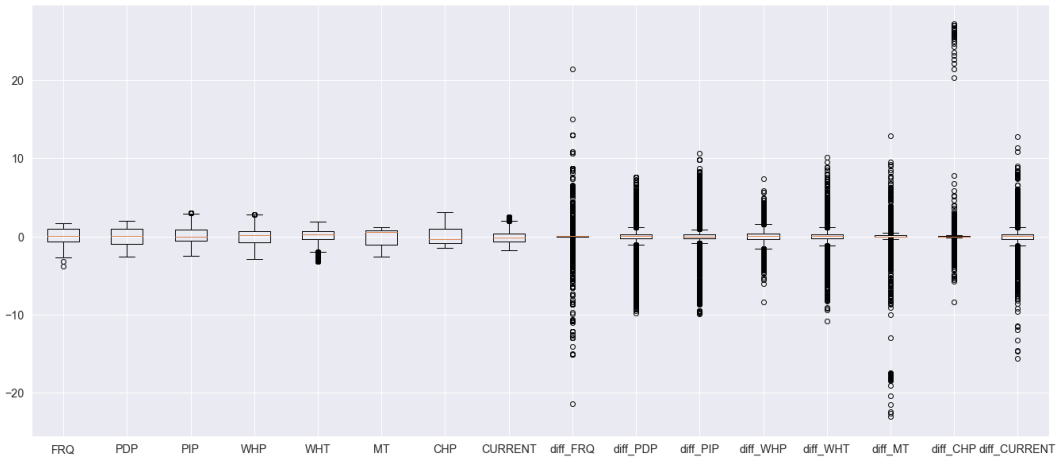


Figure 5.3: Box-plot after outlier normalization

Table 5.1: Data Exploration

	FRQ[Hz]	PDP[Psi]	PIP[Psi]	WHP[Psi]	WHT[°F]	MT[°F]	CHP[Psi]	Current[A]
mean	52.68	1522.97	855.89	642.21	62.38	137.12	922.13	349.65
std	4.29	187.46	211.08	608.61	11.49	32.40	1883.03	86.91
min	35.00	1086.52	610.46	194.06	60.55	120.63	0.00	101.60
25%	49.70	1506.13	660.90	213.39	63.23	122.20	91.54	317.01
50%	52.77	1543.80	721.80	243.58	66.00	154.19	261.38	354.01
75%	56.58	1595.05	778.90	547.34	67.66	160.60	405.29	394.74
max	64.96	1893.24	1578.29	845.86	122.84	169.30	986.74	598.21

Table 5.2: Data Points Classification

Condition	Reported Data Points
Normal	339,089
Pre-Workover	1728
Workover	288

days before the workover day. Workover events are the data points made available on workover day.

5.3 Principle Component Analysis

PCA is defined as an unsupervised dimensionality reduction technique. It reduces large dimensionality data sets into lower dimensions called principle components. This happens while preserving as much information as possible. It makes use of the interdependence of original data to build a PCA model. This results in reducing the dimension of production parameters by making the most of the linear combinations and by generating a new Principal Component space (PCs)

5.3.1 PCA Calculations

The process of obtaining a PCA model from a raw dataset is divided into four steps as follows: Firstly, the covariance matrix $\sum$ of the whole dataset is computed. It is important to see whether there is a relationship between contributing features. Equation 5.3 is used to find the covariance between each pair of dataset columns.

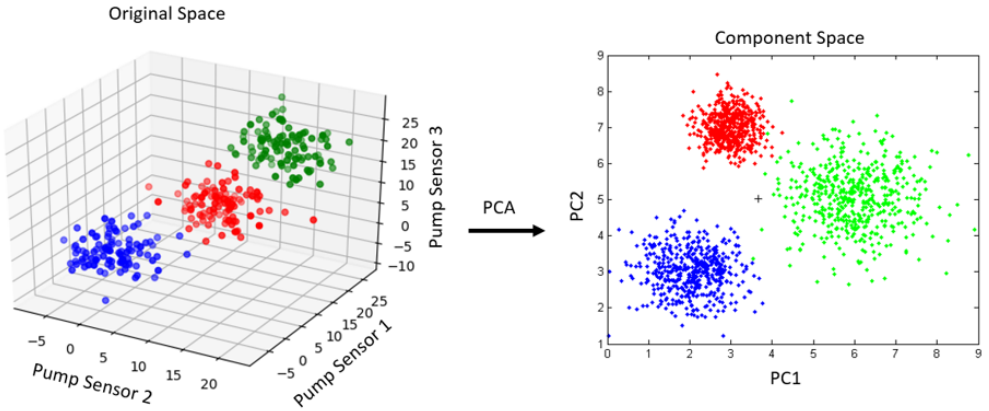


Figure 5.4: Geometric meaning of PCA

$$\sum = \frac{1}{n-1} \sum_{i=1}^{n} (X_{ij} - \widetilde{x_j})(X_{ik} - \widetilde{x_k}) \tag{5.3}$$

The second step is to calculate eigenvectors and corresponding eigenvalues. Let A be the covariance matrix that has been computed in the first step, ν a vector and λ a scalar that satisfies $A\nu = \lambda\nu$, then λ is the eigenvalue corresponding to eigenvector ν of A. This step is considered the calculation of the principal components of the data.

The third step is determining the number of principal components. The eigenvectors only define the directions of the new axis while the eigenvalues represent the variance of the data along the new feature axes. Therefore, the eigenvectors are sorted based on the eigenvalues. Hence, a threshold is chosen on the eigenvalues and cut off is made on the eigenvectors to select the most informative lower dimensional subspace. In other words, Lower variance dimensions are omitted. This is because they possess the least information about the data's distribution.

The Fourth step consists of transforming the samples into the new subspace. In this last step, the lower dimensional subspace W is selected. In the current step, the dataset samples are transformed into this new subspace via the equation $Y = W'.X$ where W' is the transpose of the matrix W. In the following, two principal components are computed and the data points are reoriented onto the new subspace. Fig. 5.4 shows the simple geometric meaning of PCA.

5.3.2 Application of PCA on Electrical Submersible Pumps

In ESP systems, sensor data are generally highly correlated, e.g., wellhead pressure is directly proportional to discharge and intake pressures. However, when a downhole problem occurs or is about to occur, anomalous data can be identified because it breaks certain rules in the input signals and their relative changes, i.e., if there is a tubing leak, the annulus discharge pressure decreases while intake pressure and annulus pressure increases, etc.

Principal Component Analysis then suits the engineers' purpose to create an anomaly detection system. This is mainly because it makes use of the interdependence of original data to build a model. The primary goal of this step is to create clusters out of the data. As discussed earlier, the selection of the principal components is made based on the maximum variance criterion. The highest variance is captured in the first principal component while the next highest variance is captured in the second principal component, where information from the first principal component has already been removed. In a similar manner, consecutive principal components (3rd, 4th, ..., Kth) can be constructed to evaluate the original system.

The PCA model finds the kth principal component to construct the PCs, where most of the information belonging to the initial system is contained. The kth principal component is represented in equation 5.4 below where PC1 is given as an example.

$$PC_k = a_{1k} * P_{intake\ pressure} + a_{2k} * P_{discarge\ pressure} + a_{3k} P_{Motor\ temperature} + \ldots + a_{pk} P_{intake\ pressure\ moving\ difference} + .. \tag{5.4}$$

Figure 5.5 shows the projection of ESP well sensor data on the principal components. The developed model is used also to evaluate near failure conditions. The problematic days and seven days before workover, sensor data clearly show specific failure patterns in line with the reported MDHF and EDHF.

The goal of the PCA is to come up with optimal weights from each sensor measurement. That means capturing as much information as possible from the input signals, based on the correlations among those variables. the loadings are the correlations between the variables and the component. The weights in the weighted average were computed from these loadings. To compute the Loading matrix, namely the correlations between the original variable and the principal components, the cross-covariance matrix is needed to be computed from equation 5.5.

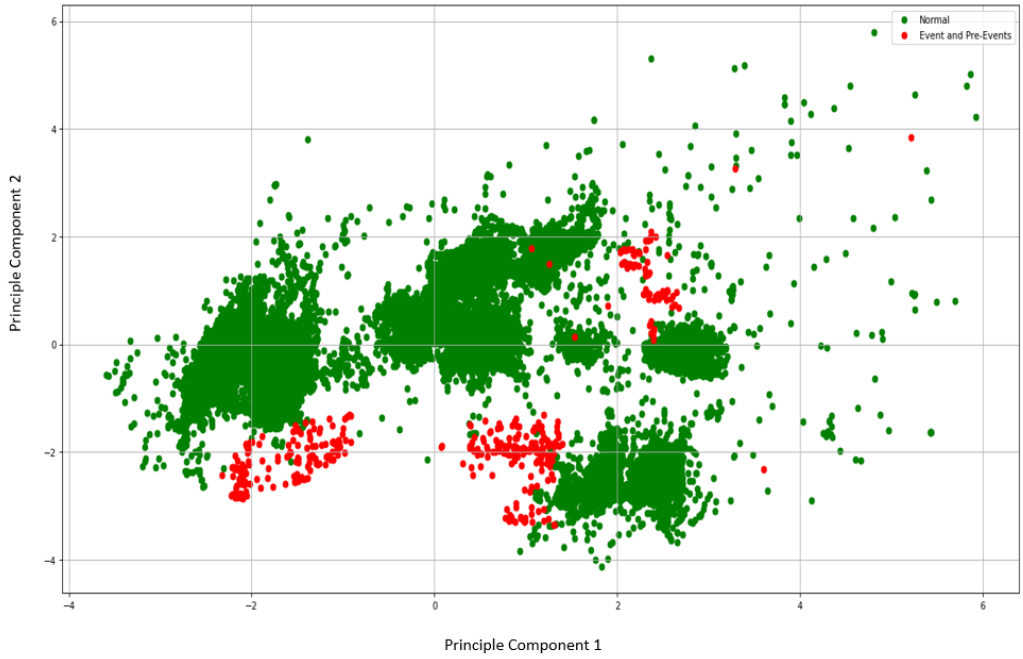


Figure 5.5: "Principle component analysis of ESP wells"

$$Cov\,(X,\ Y) = V\sqrt{E} \tag{5.5}$$

Where:

X refers to the original variables or features of the data.
Y represents the principal components obtained by linearly transforming the original variables.
V denotes the principal axes or eigenvectors of the covariance matrix of X, which are used to compute the principal components.
E are the eigenvalues of the covariance matrix of X, which represent the amount of variance explained by each principal component.

Table 5.3 represents the loading factors for each input parameter on the relevant principal components up to the 8th component. However, for the purpose of this analysis, the loading factors of parameters on the first and second principal components are of primary interest, as these components explain approximately 0.6 of the variance in the data, as depicted in Figure 5.6. Large loadings, whether positive or negative, indicate a strong relationship between a particular variable and a particular principal component. The sign of a loading indicates whether a variable and a principal component are positively or negatively correlated. Therefore, the parameters most strongly correlated with the first principal component are pump frequency, casing head pressure, current, motor temperature, and well head temperature.

Table 5.3: Loading for input parameters

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
FRQ	-0.90	-0.05	0.06	0.05	-0.05	-0.26	0.05	0.01
PDP	-0.54	-0.14	-0.70	0.32	-0.07	-0.08	0.04	0.05
PIP	0.03	-0.17	-0.47	0.14	-0.05	0.81	-0.18	-0.04
WHP	0.10	-0.02	-0.79	0.37	-0.01	-0.15	0.23	-0.28
WHT	-0.63	0.00	0.35	-0.30	0.18	0.22	0.12	-0.32
MT	-0.79	-0.12	-0.07	0.04	-0.12	0.25	-0.16	0.22
CHP	-0.85	-0.15	-0.07	0.15	-0.03	-0.27	0.07	0.12
CURRENT	-0.84	-0.10	0.21	-0.17	0.07	0.21	-0.06	-0.19
diff_FRQ	-0.14	0.78	0.03	0.27	0.09	0.02	-0.22	-0.17
diff_PDP	-0.05	0.09	-0.45	-0.54	0.21	-0.19	-0.40	0.28
diff_PIP	0.06	-0.35	-0.34	-0.67	-0.21	0.06	0.18	0.11
diff_WHP	-0.03	0.11	-0.42	-0.50	0.36	-0.15	-0.05	-0.42
diff_WHT	-0.10	0.43	-0.08	-0.11	0.37	0.23	0.69	0.26
diff_MT	-0.15	0.84	-0.10	-0.10	-0.34	0.03	-0.03	-0.03
diff_CHP	0.03	-0.29	0.03	0.30	0.85	0.01	-0.14	0.11
diff_CURRENT	-0.13	0.80	-0.14	-0.04	0.16	0.05	-0.09	0.20

5.4 XGBoosting Application

XGBoosting is a tree-based ensemble model. Ensemble learning is a systematic solution that combines the predictive abilities of multiple models, eventually resulting in a single model. This single model provides the aggregated output of several models that, on their turn, only perform slightly better than random guessing. Therefore, Extreme Gradient Boosting (XGBoost) is an ensemble set of predictors, with a unified objective of predicting the same target variable. A final prediction is performed through the combination of these set of predictors.

5.4.1 XGBoosting Calculations

Building an XGBoost model has the following sequence. It starts with a single root (contains all the training samples). Then, an iteration is performed over all features and values per feature and subsequently, each possible split loss reduction is evaluated. Equations 5.6 and 5.7 represent the objective function (loss function and regularization, respectively) at each iteration that is needed to be minimized.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, p_i + O_{value}\right) + \frac{1}{2}\lambda O_{value}^2 \tag{5.6}$$

$$l\left(y_i, p_i\right) = -\left[y_i \log\left(p_i\right) + \left(1 - y_i\right) \log\left(1 - p_i\right)\right] \tag{5.7}$$

Where:
y_i is the true value required to be predicted of the i-th instance
p_i is the prediction of the i-th instance
(y_i, p_i) The loss function for typical classification problem
O_{value} The output of the new tree
$\frac{1}{2}\lambda O_{value}^2$ Regularization Term

Tianqui stated "XGBoost objective function cannot be optimized using traditional optimization methods in Euclidean space"44. Therefore, in order to be able to transform this objective function to the Euclidean domain, the second-order Taylor approximation is used enabling traditional optimization techniques to be employed. Equation 5.8 and 5.9 represent the Taylor approximation of the loss function.

$$\mathcal{L}^{(t)} \cong \left[\sum_{i=1}^{n} l\left(yi, Pi\right) + gi\ O_{value} + \frac{1}{2} hi\ O_{Value}^2\right] + \frac{1}{2}\ \lambda O_{Value}^2 \tag{5.8}$$

Where :
G_i = is the gradient and calculated by $g_i = \frac{\partial}{\partial p_i} 1(y_i, p_i)$.
h_i = is the hessian, calculated by $H_i = \frac{\partial^2}{p_i^2} 1(y_i, p_i)$.

Finally, removing the constant parts, the simplified objective to minimize at step t, results in:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} g_i O_{value} + \frac{1}{2} h_i O_{value}^2 + \frac{1}{2}\lambda O_{Value}^2 \tag{5.9}$$

Equations 5.10 and 5.11 show how to minimize that function

$$\frac{d}{dO_{value}} \sum_{i=1}^{n} g_i O_{value} + \frac{1}{2} h_i O_{value}^2 + \frac{1}{2}\lambda O_{value}^2 = 0 \tag{5.10}$$

$$O_{value} = -\frac{\sum_{i=1}^{n} g_i}{\sum_{i}^{n} h_i + \lambda} \tag{5.11}$$

By combining equation 10 with the first and the second derivative of the classification loss function g_i and h_i, the similarity equation is derived. The similarity score is calculated as follows in equation 5.12:

$$Similarity = \frac{\sum Residual_i}{\sum Previous\ Probability_i * (1 - Previous\ Probability_i) + \lambda} \tag{5.12}$$

The similarity score is calculated for a "leaf" of the "tree". Various thresholds are used to split the tree into more leaves. The similarity score is calculated for each new leaf followed by calculating the so-called gain as presented in equation 5.13 below:

$$Gain = Left_{similarity} + Right_{similarity} - Root_{similarity} \tag{5.13}$$

Then thresholds continue to be set until higher gain thresholds are reached and the tree keeps growing. There is a minimum number of residuals in each leaf where the tree stops growing. This number is determined by calculating a parameter called cover. It is defined as the denominator of the similarity score minus lambda. During boosting, the operation is performed such that trees are sequentially constructed. Each tree reduces the error of its predecessor and learns from it while simultaneously updating the residual errors. As a result, each tree growing in the sequence will learn from a version of the residuals that's already been updated.

Further, in boosting, the base learners are weak due to their high bias and their predictive power has only a slight improvement over random guessing. Nevertheless, some vital information for prediction is supplied by each of these weak learners. By means of boosting, a strong learning effect is produced through combining these weak learners into a single strong learner that reduces both the bias and the variance.

5.4.2 XGBoosting experiments

In our proposed model, Principal Component Analysis (PCA) for sensor measurements and moving difference is pipelined with XGBoost and K-Folds Cross-validation to identify near failure regions. The dataset is divided into two groups: A training dataset containing 70% of the data and a black box testing set with the remaining 30% of the data.

The importance of principal components is evident because it shows to which extent this component is able to explain the variance in the dataset.

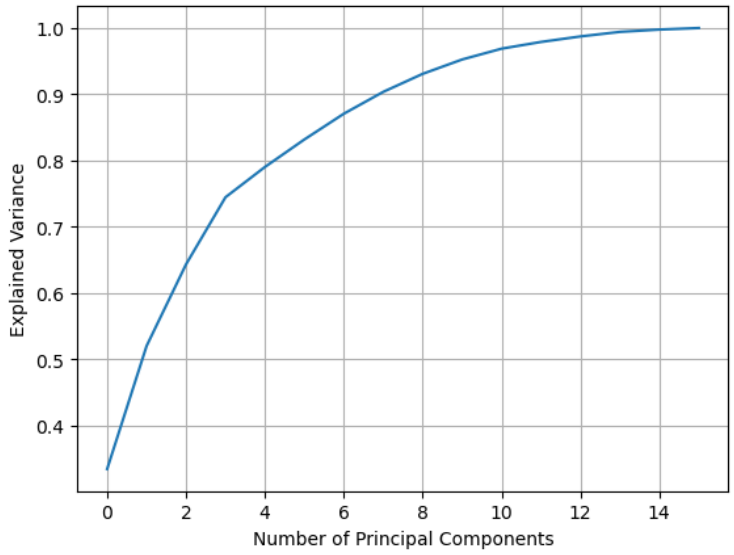


Figure 5.6: Explained Variance of the proposed model

Therefore, Figure 5.6 shows the cumulative explained variance with each principal component. It is shown that 8 principle components will include more than 90% of the explained variance in the dataset of ESP sensors and their derived features.

In the cross-validation algorithm, the data set is divided into 3 components as follows: a training set constituting 70% of the data, a validation set constituting 15% of the data and a testing set constituting the remaining 15% of the data. Each model is then trained on the training subset only, in order to infer some hypothesis. Finally, the hypothesis with the smallest error on the cross-validation set is selected.

A better estimation of each hypothesis is achieved through testing a set of examples (validation set) that the models were not trained on. A true generalization error is also obtained. As a consequence, a single model possessing the smallest estimated generalization error can be then proposed. Upon validation set error minimization, this can be further expanded such that the proposed model is retrained on the entire training set, including the validation set. It is worth noting that some risk exists in selecting validation points, which may contain a disproportionate amount of difficult and obscure examples. Therefore, the k-fold cross validation maybe applied to avoid such occurrences.

A K-fold cross validation algorithm aims at selecting validation sets. initially, the dataset is randomly divided into (k) disjoint subsets. In each subset, the number of readings is equal to the total number of data points (m) over (k).

These subsets are indicated by m1 to mk. Then, subset is evaluated for each model as follows:

All these subsets are used to train The XGBoost model, with the exception of the subset mj. The intention behind excluding this subset is to infer a hypothesis which is eventually tested on (mj). As such, the error of testing the hypothesis on the subset (mj) is calculated and the estimated generalization error of the model is calculated by averaging over (mj). Afterwards, the selection of the model with the lowest estimated generalization error is performed, and lastly the selected model is retained on the entire training set (m). The hypothesis resulting from such operation would be the final answer. When Performing cross validation, It is typically a standard that the chosen number of folds is equal to 10 (k=10).

Hyperparameter tuning is considered one of the important steps while creating any data driven model to get the best results from the deployed algorithm. Regarding the XGBoost Algorithm, hyper-parameters are divided into three categories. These categories are known as general parameters, booster parameters, and learning task parameters.

General hyper-parameters define the type of algorithm to be either linear or tree based, the verbosity to print results and the number of threads to run on. Booster parameters include the main tuned parameters for the algorithms such as learning rate, the minimum sum of weights of all observations required in an internal node in the tree, and the learning parameters to specify the minimum loss reduction required to make a split. These parameters are used to define the optimization objective and the metric to be calculated at each step. Table 5.4 shows ranges that are used for hyper-parameters tuning.

5.5 Deep Learning for Predictive maintenance

In the context of electrical submersible pumps, deep learning algorithms such as 1D Convolutional Neural Networks (1D CNNs) and Long Short-Term Memory (LSTM) networks with attention mechanisms can be used for predictive maintenance.

As aforementioned in the previous chapter, 1D CNNs are well suited for processing time series data, such as the telemetry data generated by electrical submersible pumps. They can learn to identify patterns in the data that are indicative of impending failures, and can be trained on historical data to make predictions about future failures.

Table 5.4: Hyper-parameters tuning

Parameter	Reference to	Sampling type	range
max_depth	control of over-fitting, higher depth facilitates such that the model learns relations that are specific to a particular sample.	Suggest integer value	2,10
min_child_weight	a minimum sum of weights is defined for all observations required in a child.	Log uniform	1e-10, 1e10
colsample_bytree	The subsample ratio of columns when constructing each tree.	Uniform	0, 1
learning_rate	Overfitting prevention through step size shrinkage in updates	Uniform	0, 0.1
Gamma	Specification of the minimum loss reduction required to make a split.	Suggest integer value	0, 5

LSTM networks with attention mechanisms are a type of recurrent neural network that can learn dependencies between time steps in a time series. The attention mechanism allows the network to focus on specific parts of the time series that are most relevant for making predictions.

By using the strengths of these two deep learning algorithms, it is possible to build a predictive maintenance model that can accurately predict the likelihood of failure of electrical submersible pumps. Models are trained using Focal Loss function.

Focal loss has emerged as a powerful tool in deep learning for classification tasks, especially in cases where the dataset is highly imbalanced. The loss function is designed to downweight the contribution of well-classified data points, while emphasizing the importance of hard, mis-classified points. This is achieved by introducing a tunable parameter known as the focusing parameter, which controls the degree of downweighting for easy examples.

As described in Lin et al. 2017, the focal loss function is defined as 5.14:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{5.14}$$

In this equation, p_t represents the predicted probability for a given sample, α_t is a scaling factor that can be used to balance the importance of different classes, and γ is the focusing parameter that controls the degree of downweighting of well-classified examples. By incorporating the focal loss into

the training process, the model is able to focus more on difficult-to-classify points and reduce the impact of easy-to-classify examples. This can lead to improved overall performance and better generalization to new data.

In the context of predictive maintenance, where the dataset is often imbalanced due to the rarity of failures, focal loss can be used to improve the performance of deep learning models by emphasizing the importance of hard, misclassified data-points. This can lead to more accurate and reliable predictions, ultimately improving the efficiency and effectiveness of maintenance operations.

In summary, focal loss is a powerful tool in deep learning for classification tasks, particularly in cases where the dataset is highly imbalanced. By emphasizing the importance of hard, misclassified examples, focal loss can improve the performance of deep learning models and lead to more accurate and reliable predictions. In the context of predictive maintenance, where the dataset is often imbalanced, focal loss can be a valuable tool for improving the efficiency and effectiveness of maintenance operations.

5.5.1 1D CNN

The details of 1D CNN layers and main components are presentd in chapter 4 under section 4.5.2. Hence, the architecture used for this application is presented. It consists of two layers, specifically designed for convolutional operations and max pooling. This design was adopted from a previous study (Yuan et al. 2020). The inclusion of batch normalization is also a key factor in improving the training process of the model. Batch normalization helps to stabilize the distribution of inputs to each layer, reducing the risk of overfitting and accelerating convergence during training. This improves the overall performance of the 1D CNN model and helps to ensure its stability and reliability.

5.5.2 LSTM with attention

Long Short-Term Memory (LSTM) with attention is a type of recurrent neural network (RNN) architecture that is designed to handle sequential data. LSTMs have been widely used for tasks such as natural language processing, speech recognition, time series analysis and predictive maintenance. LSTM layers and main components are presented in chapter 4 under section 4.5.2 therefore the difference between traditional LSTM and LSTM with attention is elaborated.

In traditional LSTMs, the output at each time step is a function of the input at that time step and the hidden state from the previous time step. However, this approach can sometimes lead to information loss or degradation over time, particularly when dealing with long sequences of data.

To address this issue, LSTMs with attention have been introduced. In these architectures, the attention mechanism allows the model to focus on specific parts of the input sequence, rather than processing the entire sequence equally. This helps the model to better capture the most relevant information from the sequence and improves its ability to make accurate predictions.

The attention mechanism in LSTM with attention enables the network to selectively focus on relevant segments of the input sequence during prediction. This is achieved by assigning different weights to different time steps of the input sequence, thereby allowing the network to attend to the most salient parts of the sequence. In the context of hydraulic systems predictive maintenance, this mechanism can be leveraged to identify critical trends or anomalies in the production data that are relevant to the prediction.

As illustrated in fig. 5.7, the LSTM model with attention involves T time steps (i.e., T days) in total. At each time step, the model takes as input the weighted average of news vectors for that day. The initial weights are set at the beginning and subsequently updated using gradient descent. The attention model is used to assign higher weights to pump data that are more closely related to the pump system's maintenance needs, thereby enabling the model to better predict maintenance requirements (Tseng and Tran 2023).

LSTM with attention is implemented by adding an attention layer after the LSTM layer. The attention layer takes the hidden states of the LSTM layer as input and computes attention scores for each time step. The attention scores are then used to weight the hidden states and obtain a context vector that summarizes the important information from the input sequence. The context vector is then used as input to the prediction layer to make the final prediction.

One of the key advantages of LSTMs with attention over traditional LSTMs is their ability to handle complex and variable-length sequences more effectively. This is because the attention mechanism allows the model to dynamically weigh the importance of different parts of the input sequence, based on the task at hand. As a result, LSTMs with attention are particularly useful for tasks such as sentiment analysis, machine translation, and text classification, where the input data is often highly variable in length and content.

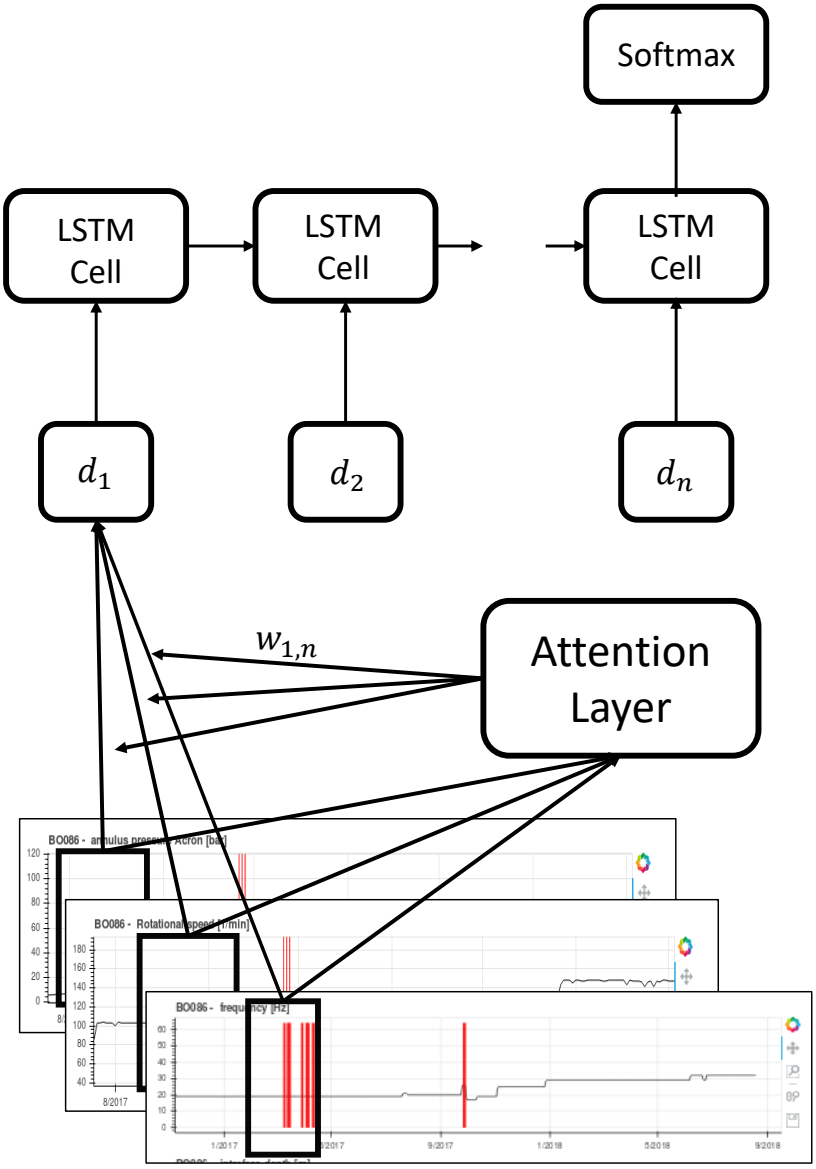


Figure 5.7: Attention weights assigned by the LSTM with attention architecture during prediction

Another advantage of LSTMs with attention is that they are typically less prone to overfitting, compared to traditional LSTMs. This is because the attention mechanism helps the model to avoid memorizing irrelevant information from the training data, and instead focuses on the most important aspects of the input sequence. Additionally, the attention mechanism also helps to reduce the number of parameters in the model, which can lead to faster training and better generalization.

5.6 Validation and Testing

It is worth noting that the success of predictive maintenance models for electrical submersible pumps will depend on the quality and quantity of data

available, as well as the specific details of the pumps and the environment in which they operate. Additionally, the choice of deep learning algorithm may need to be adjusted based on the specific characteristics of the data and the problem. In the following, the model validation and model testing for evaluating the resultant models are presented.

Model validation and model testing are crucial steps in the development of a predictive maintenance model for electrical submersible pumps using deep learning. Model validation involves evaluating the model's performance on a set of data that has not been used for training, to ensure that the model is generalizing well to unseen data. Model testing involves evaluating the model's performance on a completely independent dataset, separate from the training and validation data, to provide a final estimate of its performance and give confidence in its ability to make accurate predictions. Both model validation and model testing are performed by evaluating the model's predictions of pump failures on real-world data and comparing them to the actual failure times, using metrics such as accuracy, precision, recall, and F1 score.

5.6.1 Model Validation

Overfitting is a phenomenon that occurs when a machine learning model fits the training data too closely, resulting in a decrease in input dataset error and an increase in testing dataset error. This occurs when the learning algorithm tracks noise in the input dataset rather than the underlying concept. Overfitting can be prevented through the use of regularization and validation techniques.

Regularization can be considered as applying brakes on fitting the noise. Both hard and soft constraints can be used. Validation determines the maximum number of iterations to adjust the classifier and serves as a stopping point for the back-propagation algorithm.

In the cross-validation algorithm, the dataset is first randomly split randomly into training set (70% of the data), validation set (15% of the data), and testing set (15% of the data). Each model is then trained on the training set only to obtain a hypothesis. The hypothesis with the smallest error on the cross-validation set is then selected.

Optionally, after selecting the model based on minimizing error in the validation set, the proposed model can be retrained on the entire training set, including the validation set.

There is also a risk in the selection of validation points, as the validation set may contain a disproportionate number of difficult or obscure examples. To combat this, k-fold cross-validation can be performed.

A k-fold cross-validation algorithm is used to select validation sets. First, the dataset is split into (k) disjoint subsets. The number of examples in each subset is equal to the total number of data examples (m) over (k). These subsets are referred to as m_1 to m_k. Then for each model, it is evaluated as follows:

The model is trained for all subsets except for one of the subsets (m_j) to get some hypothesis. The hypothesis is tested on (m_j). Then the error of testing over (m_j) is calculated. The estimated generalization error of the model is then calculated as the average over (m_j).

The model is picked with the lowest estimated generalization error. Finally, the model is retrained on the entire training set (m). The resulting hypothesis is then output as a final answer. A typical choice for the number of folds to use would be k = 10. Fig. 5.8 shows 10-fold cross-validation.

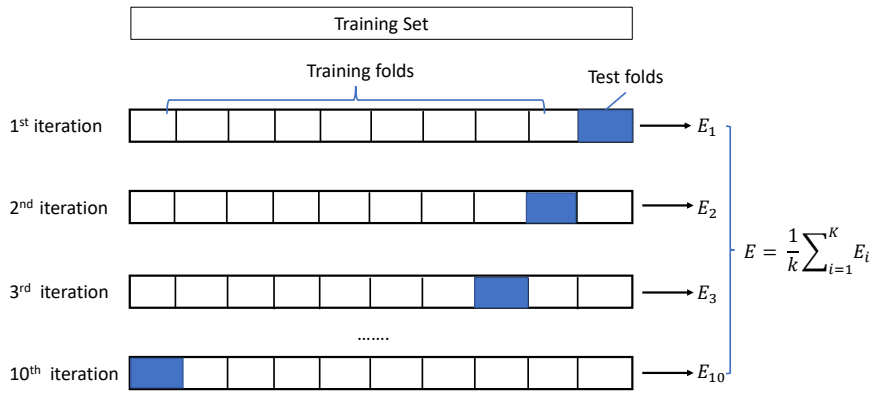


Figure 5.8: 10-fold cross validation

5.6.2 Model Testing

After choosing the best model, the classifier output quality is evaluated. The accuracy is simply the proportion of correctly classified instances. It is usually the first metric when evaluating a classifier. However, what if the test data is unbalanced? In other words, most of the instances belong to one of the classes. The accuracy does not really capture the effectiveness of a classifier in that case.

In the technical level classification scenario, it is assumed that the testing set includes 99% of the sensor data as normal pumping condition. It is possible to achieve 99% accuracy by predicting the class "Normal Condition" for all instances. The classifier in this case appears to be doing a good job overall. However, it fails to detect any ESP problems.

For that reason, it is helpful to compute additional metrics that capture more specific aspects of the evaluation. To recap on the confusion matrix that is explained in chapter 3 in section 3.3.4. The class labels in the training set can take only two possible values. These two values are either positive or negative. The positive and negative instances that a classifier predicts correctly are called true positives (TP) and true negatives (TN), respectively. Similarly, the incorrectly classified instances are called false positives (FP) and false negatives (FN).

Understanding the performance of the classifier requires answering some important questions. A very natural question is: 'Out of the sensor data classified as specific condition, how many were classified correctly?' This question can be answered by looking at the Precision of the model. The Precision is the proportion of the positives that are classified correctly (Abdalla et al. 2020). Equation 5.15 is the mathematical form of the Precision:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ positive} \tag{5.15}$$

Another common question is "Out of all cards having specific condition (TP+FN), how many did the classifier classify correctly (TP)?" This is actually the Recall, or the true positive rate. Equation 5.16 is the mathematical form of the Recall:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{5.16}$$

To conclude, the Precision-Recall metric is an important tool to evaluate the classifier output quality. Precision is a measure of how good predictions are with regard to false positives. Recall is also known as sensitivity or true positive rate. It measures how good the predictions are with regard to false negatives.

5.7 Summary

In this study, sensor measurements with moving differences were utilized to predict pumping conditions. A dimensionality reduction technique, namely PCA, was applied to reduce the dimensionality of the data, and the resulting lower-dimensional data points were then fed into a supervised learning algorithm, specifically XGBoosting, for further processing. The training dataset comprised input-output pairs, where the inputs were PCA projected features and the outputs were representative of the pumping conditions, specifically, seven days before the reported failures in the case of ESP failure prediction. Each input-output pair was considered as a "data point" for training, validation, and testing the proposed model.

In addition, deep learning techniques such as 1D CNN, LSTM, and LSTM with attention were explored in the application of predicting pumping conditions. The 1D CNN model was employed to extract relevant features from the sensor measurements by applying a one-dimensional kernel over the time-series data. The LSTM model was used to capture the sequential and long-term dependencies in the sensor measurements. Furthermore, the LSTM with attention model enhanced the LSTM model by allowing it to selectively attend to important parts of the sensor measurements during prediction.

6 Prescriptive Analysis for steam injection optimization

6.1 Introduction

Closed-loop reservoir management involves the use of dynamic optimization routines to maximize the net present value (NPV) of production or hydrocarbon recovery over the reservoir lifecycle. This is achieved by optimizing well placement, drilling schedules, and injection strategies among other parameters. Optimization routines use data from production and injection wells, geological data, and reservoir simulation models to determine the optimal control actions. These routines are critical in ensuring that the well operations are cost-effective and environmentally friendly while maximizing hydrocarbon recovery.

In recent years, advancements in upstream field technology have popularized closed-loop reservoir management approaches, where the goal is production optimization (Hou et al. 2015). The objective of closed-loop reservoir management is to use mature reservoir models and optimal well control parameters to maximize the net present value (NPV) of production or hydrocarbon recovery during the reservoir lifecycle (Foss et al. 2015).

Optimization algorithms for closed-loop reservoir management can be classified into steady-state and dynamic methods. Steady-state methods solve for a single point and include evolutionary algorithms such as genetic algorithms and particle swarm optimization algorithms. Dynamic methods, on the other hand, provide an optimal control strategy over time, including model predictive control algorithms, and reinforcement learning. Optimization algorithms also can be classified into model-free data-driven algorithms, such as reinforcement learning, and model-dependent algorithms such as gradient descent and adjoint methods. The choice of optimization algorithm depends on the complexity of the problem and the availability of data and a reliable reservoir model.

For instance, (Ameli and Mohammadi 2018) particle swarm optimization, genetic algorithm, and imperialist competitive algorithm are applied to

optimize steam to oil ratio. Their results showed that the genetic algorithm worked 6% better compared to other optimization techniques and was also faster than the continuous optimization algorithm. However, the biggest drawback of this approach is that it only provides a single value for steam injection or steam ratio, which is not sufficient for such a complex problem.

On the other hand, the second group of optimization solutions deals with steam injection as an optimal control problem, which includes a model predictive control algorithm and an adjoint-based method. Model Predictive Control (MPC) is the most widely used advanced control method in refining, chemical, and petrochemical processes (Saputelli et al. 2005; Patel et al. 2014; Purkayastha et al. 2015; Eaton et al. 2017; Vembadi et al. 2018). In the steam injection problem, a proxy model is introduced to formulate the problem, which is a relation between injection rates and oil and water production rates. Finally, the formulated problem is optimized over the prediction horizon to find the steam injection rates that maximize the net present value. Although the MPC approach is preferable, the drawback of combining it with a proxy model to formulate the system makes it unstable and only suitable for the small horizon of production (Najmudeen Sibaweihi et al. 2021).

Gradient-based optimization methods, such as the adjoint method, lie also in the second group. It is performed by some sort of gradient-based optimization method where the derivatives are obtained through the use of an adjoint equation or co-state equation (Haili Dong, Bingsheng Wang, Chang-Tien Lu n.d.). It depends on the barrier function to formulate the augmented objective function. Hence, the augmented objective function includes the calculation of the net present value and the constraints that formulate the reservoir dynamics. The disadvantage of this approach is that the equations representing the gradient of the augmented objective function with respect to the steam injection rate must be hard-coded which is not easy in the case of thermal oil recovery and compositional modeling (Guevara et al. 2021).

While steady-state optimization methods have been widely used in the past, they are not always suitable for complex reservoir simulations. In contrast, model-dependent optimization methods, such as gradient descent and adjoint methods, require complex interaction with the reservoir model which limits their practical applications. A potential solution to these challenges is data-driven optimization, which relies on machine learning algorithms to learn patterns from the reservoir model data and make predictions for optimal control strategies. Reinforcement learning is one such algorithm that has shown promising results in optimizing the control of injection and production rates in reservoir management. It allows for the optimization of non-linear and non-convex objective functions while handling uncertainties and noise in the data.

This study explores the potential of reinforcement learning for optimizing production in the oil and gas industry. It investigates the use of reinforcement learning in two different scenarios: a single-agent approach to optimize steam injection by finding the optimal injection rate for a single well, and a multi-agent approach using the Egg Model for waterflooding optimization, which involves multiple wells interacting with each other in a complex network. The study aims to showcase the efficacy of reinforcement learning in resolving the challenges of closed-loop reservoir management. This chapter focuses on steam injection optimization, while the upcoming chapter explores waterflooding optimization using a multi-agent approach.

Steam Flooding is a thermal oil recovery method. In this method, steam forms a condensing zone inside the reservoir. The heat of condensation is utilized to heat up the heavy crude oil, facilitating its displacement due to a reduction in viscosity (Ali and Meldau 1979). The injected steam forms a steam chamber around the injection well (J. Zhang and Chen 2018). Figure 6.1 shows the steam chamber around the injection well. This steam chamber is expanded towards the production well. Consequently, the condensed water displaces the reservoir fluid into the production well (Shafiei and Dusseault 2013; J. Zhang and Chen 2018).

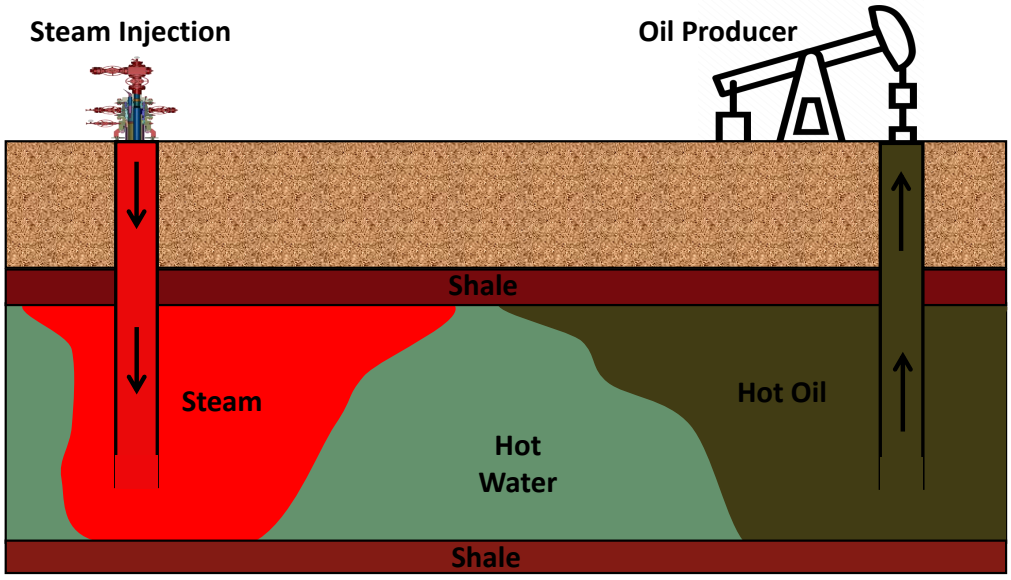


Figure 6.1: Steam Injection process

Reinforcement learning has been proposed as a means to optimize steam-assisted gravity drainage systems. Guevara et al. (Guevara et al. 2021) utilized the state-action-reward-state-action (SARSA) algorithm for this purpose. However, another algorithm, the actor-critic reinforcement learning

algorithm, has also been recommended. This approach combines the benefits of value-based and policy-based methods.

In the actor-critic algorithm, a policy neural network generates actions, which are evaluated based on the corresponding change in state potential. These values are provided by another neural network, the critic network, which approximates the expected cumulative reward from a given state (Thuerey et al. 2022). One advantage of this algorithm is its ability to update the policy and explore until it learns the optimal policy. The actor-critic algorithm is also useful in competitive non-Markovian environments where a stochastic action may be preferred over a deterministic one and in scenarios involving continuous action spaces, such as continuous robotic control (Thuerey et al. 2022). Therefore, the actor-critic algorithm shows potential as a reinforcement learning method for optimizing steam-assisted gravity drainage systems, offering benefits over SARSA.

To conclude, model-free reinforcement learning (RL) is applied in this study for steam injection rate optimization. It was selected because it overcomes the shortcomings of the aforementioned methods in two ways. Firstly, it does not require a full description of the process required to be optimized. Secondly, this approach takes advantage of previous experiences or interactions with the environment to find an optimal policy of injection rate to maximize the net present value without human interference. Also, among reinforcement learning algorithms, we explore the usage of actor-critic (A2C). The advantages of such an algorithm are the following:

1. Sample efficiency: Actor-Critic is more sample efficient than some other reinforcement learning algorithms, such as Q-learning and SARSA. This is because it uses both the value function and the policy to learn, which can help to reduce the number of interactions needed with the environment.
2. Convergence to an optimal policy: Actor-Critic has a strong theoretical basis for convergence to an optimal policy, and it guarantees improvement at each iteration. This means that the performance of the policy will improve over time.
3. Handling of non-stationary environments: Actor-Critic can handle non-stationary environments where the distribution of the data changes over time. This is because the policy is updated using its own experience, and the critic's value function is updated using the temporal difference error.
4. Ability to balance exploration and exploitation: Actor-Critic can balance exploration and exploitation using the policy and value function. The policy is used to explore the environment and discover new solutions, while the value function is used to exploit the current knowledge

of the environment and make decisions based on the expected return of each action.

6.2 Elements of RL

Reinforcement learning is a preferred method for many applications due to its ability to optimize policies and support automated transformation beyond conventional approaches. In reinforcement learning problems, an agent learns to map situations to actions in a closed-loop architecture where its actions influence later inputs and converge towards maximizing the reward signal. Through trial and error, the agent discovers which actions yield the best reward. The agent-environment interaction in reinforcement learning is illustrated in Figure 6.2.

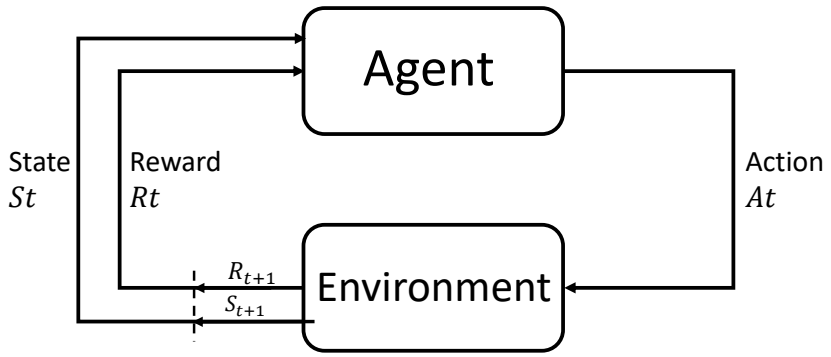


Figure 6.2: The agent-environment interaction in RL

In challenging environments, an agent's actions may not only impact immediate rewards but also affect the next situation and, therefore, all subsequent rewards. There are three key features of reinforcement learning that make it a preferred option for many applications: (1) its closed-loop structure, (2) its lack of direct instructions on actions and the consequences of those actions, and (3) its ability to fluctuate actions over extended time periods (Sutton and Barto 2005). In the following, the relevant elements are explained in further detail.

6.2.1 Environment

In reinforcement learning, the term "environment" refers to a mathematical model that simulates the behavior of the real-world environment. It allows the agent to interact with the environment by receiving observations, taking

actions, and receiving rewards. The environment can be modeled in various forms, such as a grid world or a simulated physics engine.

The environment takes the current state of the agent as an input, and based on the action taken by the agent, it returns the next state and the associated reward. The environment is responsible for updating the agent's state and providing feedback on the agent's actions.

The environment plays a crucial role in reinforcement learning because the agent's performance depends on the quality of the environment model. If the environment model is accurate, the agent can learn an optimal policy that maximizes its cumulative reward. On the other hand, if the environment model is inaccurate or incomplete, the agent may learn a suboptimal policy or fail to learn altogether. Therefore, it is important to choose an appropriate environment model that accurately represents the real-world environment.

6.2.2 Reward function

The reward function plays a crucial role as it acts as the means to guide the agent towards achieving the optimal policy. It is a mathematical function that maps the current state of the environment and the action taken by the agent to a scalar value, which represents the immediate feedback or reinforcement provided to the agent. The reward signal is considered to be the target or the objective function that the agent aims to maximize.

A reward signal is a single number that is transmitted from the environment to the agent at every time step, based on the agent's current action and the current state of the environment. The agent can influence the reward signal through its actions, and hence, maximizing the total reward that the agent receives over the long run is considered to be its primary objective. The reward signal provides feedback to the agent about the goodness or badness of its actions, and it is denoted by R_t at time step (t). The design of the reward function is a critical aspect of reinforcement learning, as it directly affects the agent's behavior and its ability to learn the optimal policy.

6.2.3 Policy function

A policy is a crucial element in reinforcement learning that specifies the learning agent's behavior at a particular time. It serves as a mapping function that connects the input vector from the environment at a given time step (t) to the actions that the agent should take. Specifically, a policy defines how the agent should behave by mapping states to actions. The policy function is

represented as π_t, where $\pi_t(a|s)$ denotes the probability of the agent taking a specific action (a) when it is in a certain state (s). The policy function plays a central role in determining the actions that the agent takes in response to the current state of the environment, and it is the key element for achieving the optimal policy that maximizes the total reward over time.

6.2.4 Value function

A value function is used to evaluate the success on the long run. It is defined as the expected return starting from a certain state, and following a particular policy. In other words, the value function at a state s considers the immediate reward available in s and the expected long-term rewards from the subsequent states. Mathematically, the value function is defined as $V_\pi(s) = \mathbb{E}\pi\left[G_t \mid S_t = s\right] = \mathbb{E}\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]$ where $V_\pi(s)$ is the value function for a given state s under a specific policy π, and G_t is the total discounted reward from time t onwards.

While the reward function determines the immediate return at a specific state, the value function takes into account the potential future rewards that may be obtained by following a particular policy. By estimating the value function for each state, an agent can determine which states are more desirable and use this information to make better decisions about which actions to take.

6.3 Learning dynamics (Agent-Environment interactions)

Over a sequence of discrete time steps $t = 0, 1, 2, 3, \ldots, T$, an agent interacts with an environment E (see fig 6.2). In every iteration, the agent receives a current state from the environment (s_t) and selects an action from some set of possible actions (A) according to its policy pi. In return, the agent receives the next state s_{t+1} and receives a scalar reward r_t. The process continues until either the agent reaches a terminal state or the maximum number of time steps satisfied after that point the process restarts (Bilgin 2020).

The total accumulated return from time step t can be calculated from $R_t = \sum_{k=0}^{\infty} \gamma^k \, r_{t+k}$ were $\gamma \in [0, 1]$ is the discount factor. The state-value function at state (s) under policy pi is defined as the expected return from that given state $V_\pi(s) = \mathbb{E}\left[R_t | s_t = s\right]$. Another important function while learning is

the action-value function. It is the expected yield for a specific action (a) at a state (s) and following specific policy $Q_{pi}(s,a) = \mathbb{E}\left[R_t|s_t\right]$.

Solving a reinforcement learning task means reaching the optimal policy. It is the one that is going to give us the maximum value in any state we are in $\pi^* = argmax_\pi \mathbb{E}_\pi \sum_{k=0}^{\infty} \gamma^k \, r_{t+k}$. we can use dynamic programming algorithms to compute optimal policies, which lead to the highest possible sum of future rewards at each state. Dynamic programming algorithms work on the assumption that we have a perfect model of the environment's Morkov decision process (MDP). So, we're able to use a one-step look-ahead approach and compute rewards for all possible actions. To find an optimal policy for a given MDP, there are two techniques. They are either value iteration technique algorithms or policy iteration technique algorithms.

In the value-iteration model-free reinforcement learning technique, there is no explicit policy stored but the problem of the value function only is solved. The policy is here implicit and can be derived directly from the value function by picking the action with the best value. For instance, DQN is the most popular Q-learning algorithm. In this algorithm, the action value function is represented using a function approximator, such as a neural network $Q(s,\ a;\ \theta)$. The objective is to directly approximate the optimal action-value function: $Q*(s,\ a)\ \approx\ Q(s,\ a;\ \theta)$. The iterative learning is done through continuous updates of the parameters θ of the action value function minimizing a sequence of loss functions.

On the contrary, we explicitly build a parameterized representation of the policy $PI(a_t|s_t;\theta)$ in the policy-based technique. Then, gradient ascent on $\mathbf{E}[R_t]$ is performed to update the parameters θ. An example of such a method is the REINFORCE family of algorithms.

In these algorithms the policy parameters θ are updated in the direction $\nabla\theta\ log\ \pi(a_t|s_t;\ \theta) * R_t$. In order to solve the problem of the high variance of such estimates while keeping it unbiased, A learned estimate of the value function is commonly used as the baseline $bt(st)\ \approx\ V\pi(st)$. This baseline is subtracted from the return. Hence, The resulting gradient is calculated from $\nabla\theta\ log\ \pi(at|st;\ \theta)\ (Rt\ -\ bt(st))$ (Sutton and Barto 2005).

Actor-critic reinforcement learning combines both techniques, the value-based and the policy based, together. It consists of two neural networks and the advantage function. The latter calculates the agent's TD Error or Prediction Error. The actor-network can be considered as a policy gradient algorithm that chooses an action at each time step. On the other hand, the critic network evaluates the Q-value or provides a feedback on how to adjust. While the critic network learns which states are better or worse, the actor

uses the critic results to teach the agent to seek out good states and avoid bad states.

6.4 Steam injection model

A reservoir simulation model is used to represent the environment, in this case, a one-third of an inverted nine-spot pattern is used to displace oil by steam (Aziz et al. 1987). The total area of the pattern is 2.5 acres with a grid system of 23 x 12 x 12. The case is shown in Fig. 6.3. Grid points are uniformly distributed in the horizontal plane as shown. The well radii for all three wells are 0.3 ft.

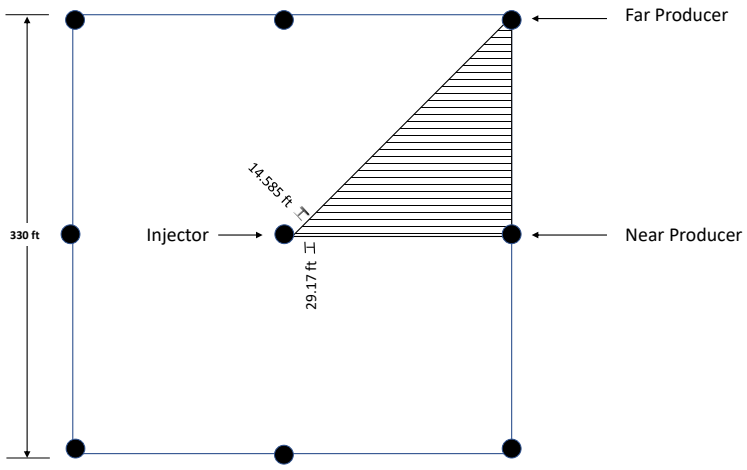


Figure 6.3: Element of symmetry used in the simulation of steam injection in an inverted nine-spot

The vertical permeabilities are 50% of the horizontal values. The porosity of all layers is 0.3 (fraction). The thermal conductivity of the reservoir overburden and underburden is 24 BTU/(ft-D-OF), and the heat capacity of the reservoir overburden and underburden is 35 Btu/(ft^3 of rock-OF). The effective rock compressibility is 5×10^{-4} psi^{-1}.

Properties of pure water are assumed. Oil density at standard conditions is 60.68 Ibm/ft3. compressibility is $5x10^{-6}$ psi^{-1}. The coefficient of thermal expansion is $3.8x10^{-4}$ OR-1, and the molecular weight is 600. Temperature and viscosity data are shown in Table 6.1. Further information about the produced fluid properties and initial oil composition can be found in table 6.2 and 6.2.

Table 6.1: Temperature and viscosity data

Temperature (◦F)	Viscosity (cp)
75	5,780
100	1,380
150	187
200	47
250	17.4
300	8.5
350	5.2
500	2.5

Table 6.2: properties of oil components

	Components		
	1	2	3
Molecular weight	250	450	600
Specific heat, Btu/Ibm-oR	0.53	0.55	0.6
Density at standard conditions Ibm/ft3	52.3	57.64	61.2
Critic pressure, psia	225	140	–
Critical temperature, oF	800	950	–

Table 6.3: Initial oil Composition

Components	Mole Fraction
C1	0.5030
C2	0.1614
C3	0.3356

Rock-fluids interaction: For water/oil system, the residual oil saturation (Sorn) is 0.15. For gas/oil system, the residual oil saturation (Sorg) is 0.10. The critical gas saturation (Sgc) is 0.06.

Regarding permeabilities, the oil relative permeability at interstitial water saturation, kroiw is 0.4. For water/oil system, water relative permeability at residual oil saturation (krwro) is 0.1. For gas/oil system, gas relative permeability at residual oil saturation (krgro) is 0.2. Equations (6.1) – (6.4) define the relative permeability for the water/oil system and for the gas/oil system as

$$k_{rw} = k_{rwro} \left(\frac{S_w - S_{wir}}{1 - S_{orw} - S_{wir}} \right)^{2.5} \tag{6.1}$$

$$k_{row} = k_{roiw} \left(\frac{1 - S_{orw} - S_w}{1 - S_{orw} - S_{iw}} \right)^{2} \tag{6.2}$$

$$k_{rg} = k_{rgro} \left(\frac{S_g - S_{gc}}{1 - S_{iw} - S_{gc}} \right)^{1.5} \tag{6.3}$$

$$k_{rw} = k_{rwro} \left(\frac{1 - S_{iw} - S_{org} - S_g}{1 - S_{iw} - S_{org}} \right)^{2} \tag{6.4}$$

Initial Conditions: The initial oil and water saturation is 55% and 45% respectively. The reservoir temperature is 125°F. The pressure at the center of the top layer is 75 psia.

Operating Conditions: The steam is injected into the bottom layer only while production occurs from all four layers. Steam injection capacity is subject to the following constraints: (I) maximum BHP of 1,600 psia at the center of the bottom layer and (2) maximum injection rate of 300 STBID on a full-well basis. The capacity of the production wells is subject to the following constraints: (I) minimum BHP of 17 psia at the center of top layer, (2) maximum production rate of 1,000 STBID of liquids. The operation shall take 820 days of injection and production.

The next step would be to reformulate the reservoir simulation problem to match with the MDP. After preparing the model, the actor-critic agent starts to interact with the environment for a production period of 820 days. Multiple well interacting with the reservoir simulation environment as a multi-agent actor-critic reinforcement learning can be considered as a next level of study.

6.5 Steam injection problem formulation based on RL

To formulate the steam injection model discussed in section 6.4 using reinforcement learning elements outlined in section 6.2, we can consider the problem as an episodic Markov decision process (MDP), where the goal is to maximize the cumulative discounted reward over a finite horizon. The subsequent discussion will elaborate on the actions, state and reward function definition based on the steam injection model.

6.5.1 State

As aforementioned, a true MDP state should provide something that is Markovian and captures the environment at its fundamental level. So, the state of the reservoir simulation should carry enough information to capture the history of the process or the previously applied injection rates. Equation 6.5 shows how the state is set at every timestep.

$$St = [Cumulative\ oil\ production\ , Cumulative\ steam\ injection, Cumulative\ water\ production] \tag{6.5}$$

6.5.2 Actions

In this study, a discrete action space is used. It consists of three (3) possible actions. Those actions are: increase, decrease and no change in steam injection rate with reference to the previous time step injection rate by a constant value (20 BPD). Such designation of action space prevents the steam injection from dramatically changing. The action space for the agent is designated according to Equation 6.6.

$$a_t = \begin{cases} Q_{inj}(t-1) + 20 & if action == 0 \\ Q_{inj}(t-1) - 20 & if action == 1 \\ Q_{inj}(t-1) & if action == 2 \end{cases} \tag{6.6}$$

6.5.3 Reward function

The reward function is very important as it is responsible for orienting the agent during the learning process. In this study, The equation used for the NPV calculation is as follows in equation 6.7. The objective is to maximize the sum of all NPV calculated at each time step.

$$R_t \;=\; NPV_t \;=\; \sum_{n=1}^{N} \frac{P_o q_o \;-\; C_{steam}\, q_s \;-\; C_{water}\, q_w}{1 + i^{\frac{t - t_{ref}}{365}}} \tag{6.7}$$

Where:

P_{oil}, C_{steam} and C_{water} are the oil price, the cost of steam generation and the cost of produced water handling in [USD/STB].
$q_o,\;\; q_s \; and \;\; q_w$ are the oil production rate, steam injection rate and water production rate in [STB/day].
$t, \; and \; t_{ref}$ are the current time and the reference time to which NPV is discounted.
i is the annual discount factor, The economical parameters are shown in table 6.4.

Table 6.4: Economical factor

Parameter	Value
P_{oil}	100
C_{steam}	12
C_{water}	3
$i_{[fraction]}$	0.2

6.5.4 Environment components summary

The environment in RL refers to the context or situation in which the agent interacts with to take actions and receive rewards. In the case of steam injection, the environment includes the reservoir, the steam injection system. The environment is modeled as a Markov decision process (MDP) where the state of the environment at each time step depends only on the state and action taken at the previous time step. Table 6.5 presents the projection of these RL elements on the steam injection model.

Table 6.5: Elements of the reinforcement learning in the context of steam injection

Element	Steam Injection context
Environment	Compositional models of reservoir simulation
Task	To find optimal injection steam injection rate autonomously along 820 days of production
State	Operating conditions $St = [Cumulative\ oil\ production, Cumulative\ steam\ injection, Cumulative\ water\ production]$
Reward	Net present value
Agent	To send actions to the injector well
Policy	To find optimal policy of steam injection rate to maximize the net present value
Action	Manipulating parameter of the operating conditions which is the injection rate

6.5.5 Implementation the Actor-to-Critic Method

The model consists of three wells (one injector and two producers), and a production horizon of 820 days (one episode) is considered. The state of the environment is defined as cumulative, including oil and water production and steam injection. For each time step, the three possible actions defined previously are considered, and the reward is represented by the NPV.

The training process is carried out through the agent's successive interactions with the environment. The agent acts according to a specific policy each time and then receives a reward. As a result, the policy is improved by the successive actions taken and the observation of new environmental states at each interaction. The agent then learns to gain rewards to act correctly in situations not present in the training set.

In the actor-critic method, the actor proposes a set of possible actions given a determined state, i.e., the actor assumes the function of the policy (where to go?). and the critic, on the other side, evaluates the actions taken by the actor. This evaluation is defined as the ”estimated value function”.

Using data from a reservoir, the environment is built as a simulation model. Fig. 6.4 shows the agent (A2C)-environment (reservoir simulator) interaction in our implementation for the determination of optimal operating conditions.

In this study, the agent can be inferred to represent an injector well. It provides the environment with actions that result in the optimal operating conditions for the simulated system. This agent is trained based on the net present value that is calculated using the feedback from the simulator. The agent used is an actor-critic.

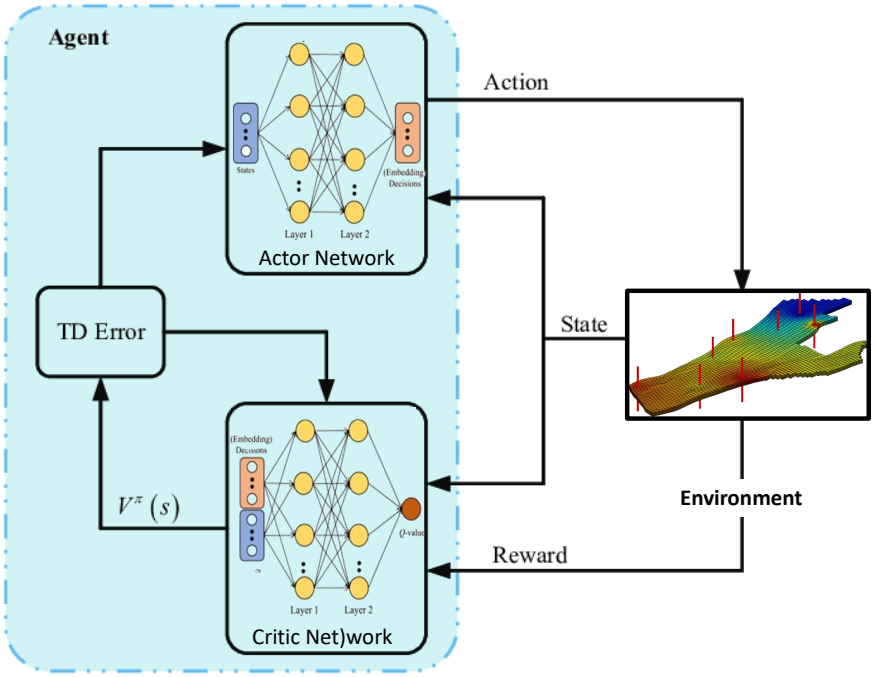


Figure 6.4: Implementation of the actor-critic architecture and its interaction with the environment

The actor network represents a parameterized policy (π_θ). Hence, it is responsible for mapping states (st) to actions (at). The output of the actor's network is a 3-dimensional vector representing the probability of the three actions. Those are increase, decrease and not changing in steam injection rate with reference to the previous injection rate. Then, using the probability distribution presented in Equation 6.8, the action is determined. On the other hand, the critic network evaluates the impact of actions by estimating the Q-value of a state-action pair. Hence, it takes both the state and the action as inputs. This ensures that the actor-to-critic agent consistently makes the best decision.

$$\pi_\theta(s,a) = \arg\max(\mathbf{softmax}p[a \mid s,\theta]) \tag{6.8}$$

The actor agent is trained through the policy gradient method and optimized policy parameters are obtained through iterations. Equations 6.9 and 6.10 show the policy gradient cost function and parameters update.

$$\nabla_\theta J(\theta) = \nabla_\theta log_{\pi_\theta}(a|s) V^{\pi_\theta}(S_t) \tag{6.9}$$

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta) \tag{6.10}$$

Combining equations (6.9 and 6.10) with the value function equation (6.11) results in Equation 6.12

$$V_{\pi}(S) = \sum_{a \in A} \pi\left(a, s\right)\left(R_{s}^{a} + \gamma \sum_{s' \epsilon \ S} T_{ss'}^{a} V^{\pi}(s')\right) \tag{6.11}$$

$$\theta_{t+1} = \theta_{t} + \alpha \left[\nabla_{\theta} log_{\pi_{\theta}}\left(a|s\right) \sum_{a \in A} \pi\left(a, s\right) \left(R_{s}^{a} + \gamma \sum_{s' \epsilon \ S} T_{ss'}^{a} V^{\pi}\left(s'\right) \right) \right] \tag{6.12}$$

During iterations, the trajectory update was observed to have a high variance due to the stochasticity of the environment and the stochasticity of the policy. The solution to mitigate this problem is the usage of an advantage function equation 6.13. The advantage function captures the degree of importance an action has compared to others for a given state, while the value function judges the strength of the decision.

$$A^{\pi_{\theta}}\left(S, a\right) = Q^{\pi_{\theta}}\left(S, \ a\right) - V^{\pi_{\theta}}\left(S\right) = r\left(a_{t}, S_{t}\right) + \gamma \ V^{\pi_{\theta}}\left(S_{t+1}\right) - V^{\pi_{\theta}}\left(S\right) \tag{6.13}$$

Using Equation 6.12 and 6.13 equations of actor and critic weights update are derived 6.14 and 6.15

$$\theta_{t+1} = \theta_{t} + \alpha [\nabla_{\theta} log_{\pi_{\theta}}\left(a|s\right) A^{\pi_{\theta}}(S, \ a)] \tag{6.14}$$

$$W_{t+1} = W_{t} + \beta A [\nabla_{w} V^{\pi_{\theta}}\left(S_{t}\right)] \tag{6.15}$$

Algorithm 1 presents the pseudo-code for the actor-critic algorithm for the steam injection process.

6.6 summary

Steam injection is a common method to enhance the recovery from mature oil fields. Commonly, a constant steam rate is applied over a long time without considering varying physical phenomena and reservoir characteristics, consequently resulting in sub-optimal performance of these thermal heavy oil recovery processes. However, finding the optimal steam injection strategy is a challenge due to the complex dynamics, i.e., nonlinear formulation,

Algorithm 1 Actor Critic algorithm for the steam injection process

Require: MDP formulation of the reservoir model
Require: Actor network: policy improvement, actions
Require: Critic network: policy evaluation, value function
Ensure: Networks parameters are set: learning rate, number of neurons and discount factor

for episode **in** Episodes **do**
    Reset the environment, returns S_t (first state of the episode
    for step **in** Steps **do**
        Execute an action$S_t, a_t,\ R_t,\ a_{t+1}, \pi(a_t \mid S_t)$
        Agent get the environment state $S_t, a_t,\ R_t,\ S_{t+1}$
        Compute advantage $A^{\pi_\theta}(S, a)$
        Update policy parameter θ_{t+1} in actor network
        Update value function parameter W_{t+1} in critic network
        Update action and state
    end for
end for

variations over time, and reservoir heterogeneity. Such a problem can be reformulated using the Markov decision process for the application of reinforcement learning (RL). Subsequently, a decision-maker called an agent generates actions to maximize the production process's yield. The agent does this by interacting continuously with the environment until it reaches the optimum route for injection rates.

In this work, an actor-critic Reinforcement learning (RL) architecture is used to train an agent to find the optimal strategy (i.e., policy) through interacting continuously with the environment. In this study, the environment is represented by a reservoir simulation model. At each time step, the agent executes an action by either increasing, decreasing, or keeping the steam injection rate constant, and a subsequent reward is received by the agent. A reward can be defined as a distinct number from the environment. Then, the agent observes the new state of the environment. Such a state could be cumulative steam injection, pressure distribution, or any other input that could be representative of the environment, which is the reservoir simulation model in our case. During this interaction, a policy function and a value function are trained by actor-critic reinforcement learning. A policy gives a probability distribution of the actions that the agent can take. For a specific policy, a value function determines the expected yield for an agent starting in a given state. This dynamic process is executed for several episodes until convergence is achieved.

7 Cooperative competitive multi-agent reinforcement learning for waterflooding optimization

7.1 Introduction

Waterflooding is a common method to increase oil recovery from hydrocarbon reservoirs. Injecting water pushes oil towards producing wells, and optimizing injection and production rates maximizes economic benefit. However, uncontrolled rates can decrease oil recovery (Xu et al. 2020) and lead to early water breakthrough. Optimizing injection and production rates is a highly effective method to improve the waterflooding process and increase the oil recovery factor in the reservoir.

In this study, a multi-agent reinforcement learning algorithm is used to optimize the waterflooding process. It's the first attempt to use a multi-agent actor-critic approach to solve production optimization problems. The study introduces the deep deterministic policy gradient method that learns complex multi-agent coordination policies, considering other agents' action policies. Memory buffer and target networks are used during training to stabilize the learning process. A memory buffer, also known as a replay buffer or experience buffer, is used to store an agent's past experiences with the environment. These experiences are later sampled and used to train the agent's model. This helps the agent to learn more efficiently by breaking harmful correlations between experiences. A target network is a copy of the agent's main Q-network that estimates Q-values. Its parameters are periodically synchronized with the main network to improve training stability.

To summarize, The aim of this chapter is to explore the effectiveness of reinforcement learning in optimizing waterflooding in the oil and gas industry. A multi-agent approach is investigated on the Egg Model, a complex reservoir optimization benchmark where multiple wells interact with each other.

The research aims to contribute to the understanding of closed-loop reservoir management challenges and demonstrate the potential of reinforcement learning in addressing these challenges.

7.2 Multi agent deep deterministic policy gradient

This section outlines an algorithm designed for multi-agent environments. The algorithm has two main assumptions. Firstly, policies are only able to access local information, specifically, the parameters of a well's grids that are obtained via reservoir simulation. These parameters include volumetric parameters like pressure and water saturation. Secondly, the communication method between agents is not specified. By satisfying these two assumptions, a versatile multi-agent learning algorithm is created that can be used for both cooperative agents (e.g., injectors) and competitive agents (e.g., injectors and producers) that interact with the environment physically.

In single-agent problems, the Deep deterministic policy gradient (DDPG) has been shown to be an effective method for continuous control tasks, where the agent learns a deterministic policy and can explore the action space with noise added to the action output (Lowe et al. 2017). In the DDPG algorithm, the agent learns a policy to do a task in the environment. This policy is a neural network (NN) trained through a back-propagation algorithm. Policy learning is guided by Q-value, which also is a learned (NN) by Q-value. In such problems, learning is not affected by the environment dynamics as it is stationary from the agent's perspective. Hence, the agent can learn the environment model through its own single interactions. However, in a multi-agent setting, the environment becomes non-stationary from the perspective of any individual agent. This non-stationarity happens because the agent's reward cannot be explained by changes in its own policy.

In multi-agent problems, the goal of learning multi-agents is achieved through adopting the DDPG algorithm and a framework for centralized training with decentralized execution (Lowe et al. 2017). In this framework, all agents are able to observe the states and actions of all other agents during training. However, during execution, each agent only has access to its own state observations and must predict its own actions based on this information. This helps in easing the training as the environment becomes stationary for each agent. In other words, all agents' states and actions are concatenated in the training process which serves as an input to the critic neural network (Q-value). This critic (Q-value) is used as a baseline to train the actor, which gets only particular agent state observations as an input and its output is the action for that agent.

In summary, policies are allowed to use extra information to ease the training process. However, this information is not used during the testing process. It is not possible to do this with Q-learning, as the Q-function generally cannot contain different information at training and test times. Therefore, Multi-Agent Deep Deterministic Policy Gradient (MADDPG) is applied. It is considered an extension of the DDPG algorithm. It uses an actor-critic architecture to learn policies for multiple agents in a cooperative or competitive environment. In MADDPG, each agent has its own actor-network that selects actions based on its local observations, while a centralized critic network estimates the value function based on the observations and actions of all agents. In the upcoming section, the DDPG algorithm is clearly explained and how it is combined with the centralized critic and decentralized actor. The concepts of replay buffer, target networks, and exploration are presented under DDPG.

7.2.1 Deep deterministic policy gradient (DDPG)

DDPG is an actor-critic algorithm that can be used to solve continuous control problems in a reinforcement learning setting. It is an extension of the Deep Q-Network (DQN) algorithm that is used for discrete action spaces.

The algorithm is based on the idea of using a deterministic policy, which is a function that maps states to actions directly, instead of a stochastic policy, which outputs a probability distribution over actions. This is achieved by using an actor-network to represent the policy, which takes in the state as input and outputs the action.

The critic network is used to evaluate the policy by estimating the state-action value function, which is the expected sum of rewards starting from the current state and taking the action specified by the policy. This is used to train the actor network by backpropagating the gradient of the estimated value with respect to the actor network parameters.

DDPG employs four neural networks, namely a Q-network, a deterministic policy network, a target Q-network, and a target policy network. The parameters of the deterministic policy neural network and its corresponding target neural network are represented by the symbols ϕ and ϕ_{targ}, while the Q-network and the target Q-network learning parameters are denoted by θ and θ_{targ}, respectively.

The algorithm's pseudo-code is presented in algorithm 2. The algorithm starts by initializing the actor network, critic network, target actor-network, and target critic network with random weights, as well as initializing the replay buffer. For each episode, the environment is initialized and the initial

state is observed. Then, for each time step within the episode, an action is selected using the actor-network and exploration noise, and executed in the environment. The next state and reward are observed and the transition is stored in the replay buffer. A random batch of transitions is then sampled from the replay buffer and used to update the critic network by minimizing the mean squared error between its predicted value and the target value computed using the target critic network and the Bellman equation. The actor-network is updated using the sampled policy gradient. The target networks' parameters are updated slowly towards their respective main networks' parameters. This process is repeated until the end of the episode. In the following, the four primary components that form this algorithm are explained. These components are exploration, replay buffer, actor-critic networks update and target networks update.

Exploration

In reinforcement learning with discrete action spaces, exploration is typically achieved through the probabilistic selection of a random action using methods such as epsilon-greedy or Boltzmann exploration. However, in Deep Deterministic Policy Gradient (DDPG) training, which is conducted in an off-policy manner, this approach may not allow agents to discover a wide enough variety of actions to generate useful learning signals. To address this, (Lillicrap et al. 2015) introduced an Ornstein-Uhlenbeck (OU) process to add noise to the action output, which encourages exploration. The OU process is a stochastic process that generates temporally correlated noise.

However, recent studies suggest that uncorrelated, zero-mean Gaussian noise works as well (Airaldi et al. 2022). It is also preferred since it is simpler than OU. Therefore, the fixed scale zero-mean Gaussian noise is kept during training to ensure better exploration, but noise is not applied during testing to evaluate the policy's ability to exploit what it has learned.

Replay buffers

DDPG uses a replay buffer to sample experience to update neural network parameters. It is a set $\mathcal{D}$ of previous experiences. During each episode roll-out, all the experience tuples (state, action, reward, next state and done) are saved in a finite-sized cache. Then, random mini-batches of experience are sampled from the replay buffer when the value and policy networks are updated. In order for the algorithm to have stable behavior, the replay buffer should be large enough to contain a wide range of experiences.

Algorithm 2 Deep Deterministic Policy Gradient

Input: initial policy parameters θ, Q-function parameters ϕ, empty replay buffer $\mathcal{D}$

Set target parameters equal to main parameters: $\theta_{\text{targ}} \leftarrow \theta$, $\phi_{\text{targ}} \leftarrow \phi$

repeat

Observe state s and select action $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{\text{low}}, a_{\text{high}})$, where $\epsilon \sim \mathcal{N}(0, \sigma)$

Execute a in the environment

Observe next state s', reward r, and done signal d indicating whether s' is terminal

Store (s, a, r, s', d) in replay buffer $\mathcal{D}$

if s' is terminal **then**

Reset environment state

end if

for time step t in $1, 2, ..., T$ **do**

Randomly sample a batch of transitions, $k = (s, a, r, s', d)$ from $\mathcal{D}$

Compute targets:

$$y_t = r_t + \gamma(1 - d_t) Q_{\phi_{\text{targ}}}(s_{t+1}, \mu_{\theta_{\text{targ}}}(s_{t+1}))$$

Update Q-function by one step of gradient descent using the mean squared error loss:

$$\nabla_\phi \mathcal{J}(\phi) \approx \nabla_\phi \frac{1}{k} \sum_{(s,a,r,s',d) \in B} \left(Q_\phi(s, a) - y \right)^2$$

$$\phi \leftarrow \phi - \alpha_\phi \nabla_\phi \mathcal{L}(\phi)$$

Update policy by one step of gradient ascent using the policy gradient:

$$\nabla_\theta \mathcal{J}(\theta) \approx \frac{1}{K} \sum_{s \in B} \nabla_a Q_\phi(s, a) | a = \mu\theta(s) \nabla_\theta \mu_\theta(s)$$

$$\theta \leftarrow \theta + \alpha_\theta \nabla_\theta \mathcal{J}(\theta)$$

Update target networks with a soft update:

$$\phi_{\text{targ}} \leftarrow \rho \phi_{\text{targ}} + (1 - \rho)\phi$$
$$\theta_{\text{targ}} \leftarrow \rho \theta_{\text{targ}} + (1 - \rho)\theta$$

end for

until convergence

Actor and critic networks update

The critic network estimates the expected cumulative reward for an action in a given state. It is updated by minimizing the mean squared error between the predicted value and the target value. The target value is calculated using the Bellman equation and the target critic network as depicted in equation 7.1.

$$y(r, s', d) = r + \gamma(1 - d)Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s')) \tag{7.1}$$

Where:
r is the reward obtained by taking action a in state s, s' is the next state, d is the done flag indicating whether the episode is over, γ is the discount factor, $Q_{\phi_{\text{targ}}}(s', \mu_{\theta_{\text{targ}}}(s'))$ is the target Q-value predicted by the target critic network, and $\mu_{\theta_{\text{targ}}}(s')$ is the target action selected by the target actor network in the next state s'. Then, the gradient of the loss with respect to the critic network parameters is used to update the parameters in a direction that reduces the loss. Equation 7.2 shows the critic loss function.

$$\nabla_\phi \mathcal{J}_{\text{critic}} \approx \nabla_\phi \frac{1}{K} \sum_{(s,a,r,s',d) \in B} \left(Q_\phi(s, a) - y(r, s', d)\right)^2 \tag{7.2}$$

Where:
K represents a mini-batch of experience. The gradient of the loss with respect to the critic network parameters is then computed and used to update the parameters in a direction that reduces the loss. This is done using gradient descent with the learning rate α_{critic}. The critic update is shown in equation 7.3

$$\phi \leftarrow \phi - \alpha_{\text{critic}} \nabla_\phi J_{\text{critic}} \tag{7.3}$$

In DDPG, the actor network is updated using the policy gradient. It is computed by taking the gradient of the expected cumulative reward with respect to the actor network parameters. The policy gradient is defined as 7.4:

$$\nabla_\theta \mathcal{J}_{\text{actor}} \approx \frac{1}{K} \sum_{i=1}^{K} \nabla_a Q_\phi(s_i, a) | a = \mu\theta(s_i) \nabla_\theta \mu_\theta(s_i) \tag{7.4}$$

Where:
$\mathcal{J}_{\text{actor}}$ is the expected cumulative reward, θ are the actor network parameters,

K is the batch size, s_i is the state at time i, a is the action, $Q_\phi(s_i, a)$ is the critic network's estimate of the action-value function, $\mu_\theta(s_i)$ is the actor network's output action at state s_i, and ∇_θ and ∇_a denote the gradients with respect to the actor network parameters and actions, respectively. The actor network parameters are then updated in a direction that maximizes the expected cumulative reward as defined in 7.5.

$$\theta' \leftarrow \theta + \alpha_{\text{actor}} \nabla_\theta J \tag{7.5}$$

Here, θ' are the updated actor network parameters and α_{actor} is the learning rate for the actor network.

The target networks

Target networks are used to stabilize the learning process. There are two target networks: a target actor network and a target critic network. These networks are copies of the main actor and critic networks, respectively, with their parameters updated slowly via "soft updates" towards their respective main networks' parameters.

The target networks are used to compute the target values for training the main networks. For example, the target critic network is used to compute the target value for training the main critic network using the Bellman equation. In other words, the solution (the target networks method) is to use a set of parameters for actor and critic networks that comes close to the main networks but with a time delay. The usage of these target networks helps to reduce the correlation between the target values and the current estimates which can improve the stability of training.

The parameters of the target networks are denoted ϕ_{targ} and θ_{targ}. The target network is updated once per the main network update by Polyak averaging 7.6 and 7.7:

$$\phi_{\text{targ}} \leftarrow \rho \phi_{\text{targ}} + (1 - \rho)\phi \tag{7.6}$$

$$\theta_{\text{targ}} \leftarrow \rho \theta_{\text{targ}} + (1 - \rho)\theta \tag{7.7}$$

where ρ is a hyperparameter between 0 and 1 (usually close to 1). This hyperparameter is called `polyak`.

7.2.2 Multi-agent decentralized actor, centralized critic approach

As aforementioned, a centralized critic is used to estimate a Q-function and a decentralized actor to approximate each agent's policy function. Figure 7.1 explains the process that was presented in (Lowe et al. 2017).

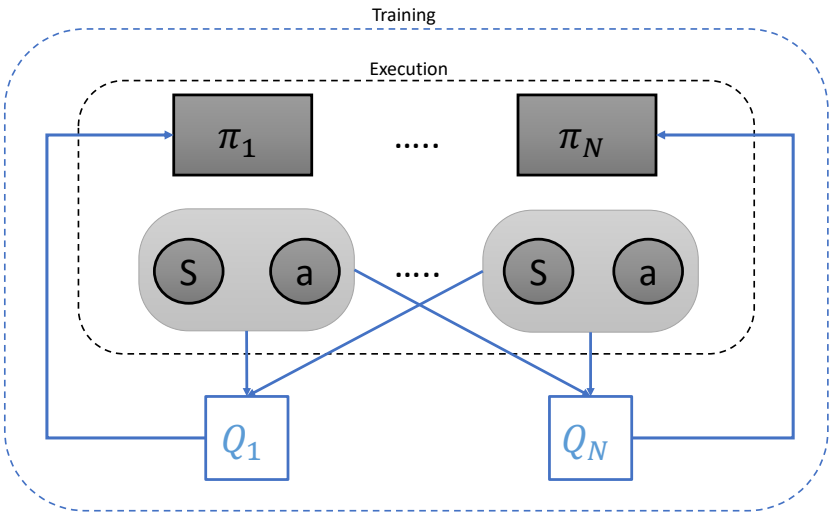


Figure 7.1: Overview of multi-agent decentralized actor, centralized critic approach

In this approach, each agent has its own policy function, represented by a set of parameters θ_i. The objective is to find the set of parameters that maximizes the expected return for each agent, denoted as $J(\theta_i)$.

To achieve this objective, the algorithm uses a centralized critic, which estimates the Q-function for the joint action of all agents. This Q-function is represented as $Q_i^{\pi}(\mathbf{x}, a_1, ..., a_N)$, where $\mathbf{x}$ is the state of the environment and $a_1, ..., a_N$ are the joint actions of all agents.

The policy of each agent is updated based on the gradient of its expected return, as expressed in the equation 7.8:

$$\nabla_{\theta_i} \mathcal{J}(\theta_i) = \mathbb{E}[\nabla_{\theta_i} \log \pi_i(a_i|s_i) Q_i^{\pi}(\mathbf{x}, a_1, ..., a_N)]. \tag{7.8}$$

Equation 7.8 computes the gradient of the expected return with respect to the policy parameters θ_i. It consists of two parts: the first part $\nabla_{\theta_i} \log \pi_i(a_i|s_i)$ is the gradient of the log probability of the agent's action, and the second part $Q_i^{\pi}(\mathbf{x}, a_1, ..., a_N)$ is the estimated Q-value of the joint action.

By updating the policy of each agent based on this gradient, the algorithm learns to maximize the expected return for each agent while taking into account the actions of all agents in the environment.

The critic network is shared among all agents and trained to minimize the mean squared error between the estimated state-action value and the true state-action value. This is done by computing the gradient of the critic network's parameters ϕ_i as:

$$\nabla \mathcal{J}(\phi) = \mathbb{E}(s_t, a_{1:t}, r_t, s_{t+1}) \sim \mathcal{D}\left[\left(Q^{\mu}(s_t, a_{1:t}, \mathbf{a_{-i}}; \phi) - y_t\right)^2\right], \tag{7.9}$$

where Q^{μ} is the Q-function and $y_t = r_t + \gamma Q^{\mu}(s_{t+1}, a_{1:t+1}, \mathbf{a_{-i}}, \mu_{i+1}(s_{t+1}); \phi)$ is the target value for the critic network, with γ being the discount factor. a_{-i} represents the actions taken by all other agents in the system, excluding the actions taken by agent i

After training, each agent executes its own learned policy independently based on its local observations and the joint Q-function learend. There is no longer any training involved, and the critic network is not used. Each agent's policy only depends on its own observations, and it does not require any information from the other agents' policies or actions. This is why the approach is referred to as "decentralized" - each agent's policy is independent and does not require coordination with the other agents during testing.

During testing, the agents interact with the environment and receive observations and rewards. Each agent then selects an action based on its own policy and the current observation, without any communication or coordination with the other agents. The agents' actions are sent to the environment, which produces a new observation and reward for each agent. The process continues until the episode ends, and the agents' individual returns are computed based on the total reward they received during the episode.

7.3 Waterflooding mechanistic model

The algorithm developed is utilized for optimizing the water-flooding process in the well-known Egg Model, which was originally created by Maarten Zandvliet and van Essen. Figure 7.2 displays the typical Egg model grids, which consist of 8 injection wells depicted in blue and 4 production wells depicted in red. The model consists of 25,200 grids with 18,553 active cells spread across 7 vertical layers. The inactive cells are located on the outer

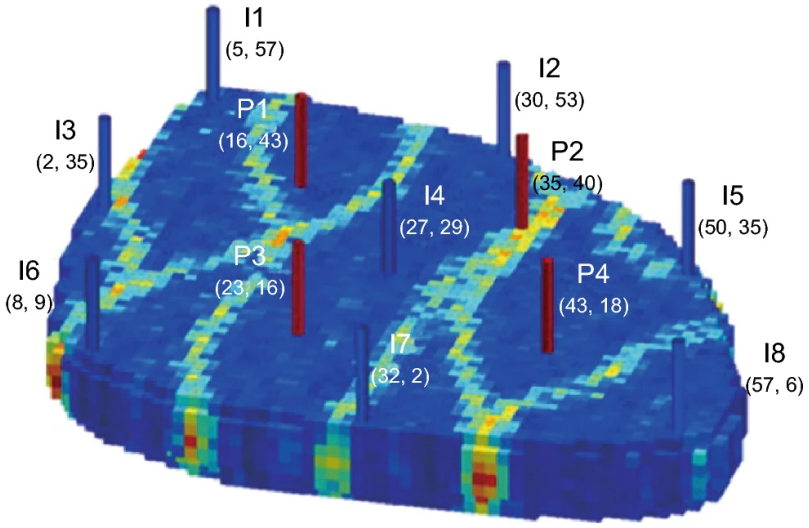


Figure 7.2: Standard Egg model reservoir (revised from J. D. Jansen et al. 2014) (P: production well, I: injection well)

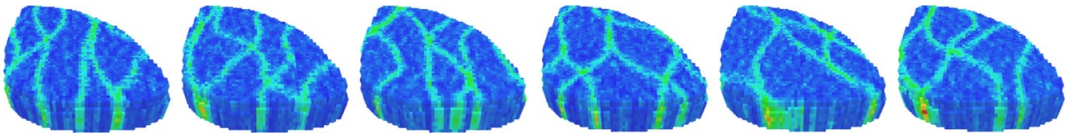


Figure 7.3: Random permeability realizations for the Egg model

edges of the model, resulting in an egg-shaped model comprising only active cells.

The high-permeability channels in a low-permeable background represent typical meandering river patterns as encountered in fluvial environments. Each field is unique, and, as a result, the permeability in each of the cell locations can be described with a probability distribution. However, there exists no underlying mathematical model to create additional permeability fields. The fields display a clear channel orientation with a typical channel distance and sinuosity. The seven layers have a strong vertical correlation, such that the permeability fields are almost two-dimensional. A random sample of six realizations is displayed in Fig. 7.3

The rest of the geological characteristics and fluid properties of the standard egg model are presented in table 7.1. Further geological characteristics of the standard Egg-Model as well as other parameters such as injection and production well locations and initial adjustments are available in (Essen et al. 2006; J. D. Jansen et al. 2014).

Table 7.1: Geological and fluid properties of standard Egg-Model

Symbol	Variable	Value	SI Units
h	Grid-Block height	4	m
$\Delta x, \Delta y$	Grid-Block height /width	8	m
C_o	Oil Compressiblity	3×10^{-6}	psi^{-1}
C_r	Rock Compressiblity	2.5×10^{-6}	psi^{-1}
C_w	Water Compressiblity	5×10^{-6}	psi^{-1}
μ_o	Oil Dynamic Viscosity	3	cp
μ_w	Water Dynamic Viscosity	0.5	cp
n_o	Corey Exponent, Oil	4.0	-
n_w	Corey Exponent, Water	3.0	-
P_c	Capillary Pressure	0.0	Pa
$S_{w,o}$	Initial Water Saturation	0.1	-
T	Simulation Time	3600	day

The waterflooding control problem involves optimizing the injection rate and timing to maximize oil recovery while minimizing water production. The optimization algorithm adjusts the water injection rate over time to minimize the objective function. The objective function for the optimization problem can be expressed mathematically as 7.10. The model's inputs include geological information, fluid properties, and well data, and the outputs provide a prediction of the reservoir's pressure, temperature, and fluid saturation over time.

$$\min_{q,t} J(q,t) = \int_0^T \left(\frac{1}{2} q^2(t) + \frac{1}{2} w(t)^2 \right) dt \tag{7.10}$$

where $q(t)$ is the water injection rate and $w(t)$ is the water production rate. The objective function seeks to minimize the sum of the square of the injection and production rates over time. To make the optimization problem well-posed, constraints must be added to ensure that the solution is physically feasible. For example, the water injection rate must be positive and the pressure in the reservoir must not exceed a maximum value to avoid fracture damage. The constraints can be expressed mathematically as 7.11:

$$\begin{aligned} q(t) &\geq 0 \\ BHFP(t) &\leq BHFP_{max} \end{aligned} \tag{7.11}$$

where $BHFP(t)$ is the bottom hole flowing pressure of the injector well at time t and $BHFP_{max}$ is the maximum allowed bottom hole flowing pressure.

The optimization algorithm iteratively adjusts the water injection rate until a solution that satisfies both the objective function and the constraints is found. The solution provides the optimal water injection rate and timing to maximize oil recovery while minimizing water production.

7.4 Waterflooding problem formulation based on RL

As mentioned previously, the primary goal of the waterflooding process is to maximize oil extraction while minimizing costs, ultimately resulting in maximum profits at the end of the depletion stage. To achieve this, reinforcement learning is utilized to optimize decision-making in injection and production wells. It is crucial to define the reinforcement learning problem and reward signal in a manner that aligns with the objective. The purpose of this section is to define each element of the reinforcement-learning architecture to ensure consistency with the overall goal.

7.4.1 The states set

In deep reinforcement learning, the state is the only aspect that an agent can observe while interacting with its environment. This information helps the agent to learn and make decisions that maximize its reward. Therefore, it is important to choose the relevant observation variables for the agent to learn from. The reinforcement learning function approximator, which is an artificial neural network, can be used to understand the interaction between the environment, state, and action. The network can take raw data and find patterns in hidden layers, leading to better estimates.

As mentioned previously, when applying deep reinforcement learning to reservoir simulation, several candidates can be considered as states:

- Wells operational information such as fluid production/injection rates, water cut, bhp, etc.
- Cumulative information as total oil/water/gas production, field water-cut, average pressure, etc.
- Temporal volumetric information such as the distribution of fluid pressure, saturation, relative permeability, etc.

Training with volumetric data enables agents to learn about relative well locations from changes in the variable distribution. Consequently, volumetric

data were used in this case. In a waterflooding environment, the pressure and water saturation are selected to represent an agent's state.

Specifically, a well window is chosen for each well, and for each window, two dynamic volumetric parameters are used (pressure and water saturation) to represent the agent's partial observation. This approach will enable us to optimize the injection rates based on their effect on the well windows around them and those in the vicinity of the production wells.

7.4.2 The actions set

In the oil recovery process, the water injection and production rates play a critical role in determining the economic return of an oil field. To control these rates, the injection and production rates of each well can be defined as the action vector: $a_t = [q_{inj,1}, \ldots, q_{inj,N_{inj}}, q_{prod,1}, \ldots, q_{prod,N_{prod}}]$. The optimization of these injection and production rates is a continuous problem. With eight injectors and four producers, the control of this optimization problem lies in a multidimensional continuous action space.

To address the multidimensional continuous action space problem, a deep deterministic policy gradient (DDPG) algorithm is employed to control both the injection and production rates. In this case, the actor network must ensure that negative values are not allowed, as negative injection rates can result in early termination of the episodes. To accomplish this, a rectified linear unit (ReLU) activation function is utilized in the actor network.

The ReLU has two advantages in this scenario. First, it ensures that only non-negative values are allowed and negative values are transformed to zero. Second, its 45-degree straight-line transformation allows for positive values to remain unchanged. Thus, the selected activation function allows for any continuous value in the range of $0 \leq q_{inj} \leq q_{inj}^{max}$.

7.4.3 The reward

As aforementioned, the rewards in reinforcement learning are numerical values that the agent receives after performing an action in a specific state in the environment. In the context of oil recovery using waterflooding, the ultimate goal is to maximize the total profit, which is why the net present value (NPV) can be used as an appropriate reward function. However, defining the limits on the injection rate can be difficult, so a hybrid reward function that combines both discrete and continuous elements is used. This function is presented by equation 7.12.

$$NPV_t = \begin{cases} \sum_{n=1}^{N} \frac{P_o q_o - C_{inj-water}\ q_{winj} - C_{prod-water}\ q_{wprod}}{1+i^{\frac{t-t_{ref}}{365}}} & if q_{inj} =< q_{inj}^{max} \\ 0 & if q_{inj} > q_{inj}^{max} \end{cases} \tag{7.12}$$

The terms in equation 7.12 have the following meanings:

- P_{oil}: the oil price in [USD/STB]
- $C_{inj-water}$: the cost of water injection in [USD/STB]
- $C_{prod-water}$: the cost of produced water handling in [USD/STB]
- q_o: the oil production rate in [STB/day]
- q_{winj}: the water injection rate in [STB/day]
- q_{wprod}: the water production rate in [STB/day]
- t: the current time
- t_{ref}: the reference time to which NPV is discounted
- i: the annual discount factor

The hybrid reward function calculates the NPV by subtracting the costs of water injection and produced water handling from the oil price and dividing by a factor that takes into account the time discounting. The function is zero if the injection rate exceeds the maximum allowable value. This helps ensure that the agent stays within the constraints of the system while maximizing the NPV.

The economical parameters are shown in Table 7.2.

Table 7.2: Economical factor for waterflooding use-case

Parameter	Value
P_{Oil}	100
$C_{InjectedWater}$	12
$C_{ProducedWater}$	3
$i_[Fraction]$	0.2

7.4.4 Neural networks for MADDPG

MADDPG algorithm utilizes neural networks as the function approximator for the agent policies and value functions. The actor network is responsible for choosing actions based on the state, while the critic network estimates the value function of the current state-action pair. The use of neural networks in MADDPG allows for the approximation of highly nonlinear and

complex functions, making it suitable for modeling complex multi-agent environments. Additionally, neural networks can adapt and improve over time as they receive feedback and learn from experience.

In reservoir simulation modeling, the use of LSTM in MADDPG networks offers numerous benefits. With a large number of input variables, LSTM networks can efficiently manage high-dimensional data, making them a perfect choice for such complex systems. Moreover, reservoir simulation models are inherently dynamic and involve complex, nonlinear interactions between various components. In this regard, the LSTM networks can capture temporal dependencies and long-term correlations in the data, which is particularly useful for modeling complex systems.

Additionally, reservoir simulation is a continuous process, and the use of LSTM networks can help in maintaining data continuity over time. This is particularly important when dealing with huge changes in the system such as changes breakthroughs. LSTM networks can retain the previous state information and update it with the current state, providing more accurate predictions and better control of the system.

Moreover, the use of LSTM in MADDPG networks allows agents to learn from their past experiences and adapt their behavior accordingly. This is particularly essential in reservoir simulation, where the system behavior can be highly non-linear and dependent on various factors, such as well placement, production rates, and fluid properties. LSTM networks enable agents to identify patterns and correlations in the data, thereby adjusting their decision-making processes accordingly. This leads to a more efficient reservoir management and improved production rates.

7.5 Summary

This chapter presents a novel approach for optimizing waterflooding in the oil and gas industry. Conventional methods rely on either model-based or steady-state formulations, which are insufficient for solving complex real-world problems. Model-based formulations can be mathematically complex and require increased complexity for more complicated reservoir models, whereas steady-state approaches only reach an optimized single point.

A multi-agent reinforcement learning approach is used. It is a model-free data-driven optimization algorithm in which multiple agents are trained to optimize their individual actions in response to the actions of other agents. The agents learn through trial and error and are motivated to achieve

a common goal, such as maximizing oil production. This cooperative - competitive approach allows agents to learn from each other and find an optimal solution through an iterative process.

DDPG and multi-agent decentralized actor-centralized critic frameworks are combined to discover various physical and informational strategies in the reservoir model. The proposed method is applied to a benchmark egg model, and the learning curve is analyzed. An analysis of the proposed optimization policy and a comparison with another data-driven optimization approach are presented in the results chapter.

Overall, this chapter highlights the implementation of cooperative competitive multi-agent reinforcement learning for waterflooding optimization and provides a valuable contribution to the literature on this topic.

8 Results and Discussion

8.1 Introduction

In this chapter, the results of several models developed for different problems are reported and their performance is discussed. Each model was evaluated using appropriate metrics relevant to its specific objective to assess its ability to achieve the desired outcomes.

First, the results of the virtual flow metering are presented. The developed model accurately predicted the liquid rate and water cut flow using pump sensor measurements. This eliminates the need for multiphase flow meters, which are costly and time consuming to install. The virtual flow metering model can be used in real time for the continuous monitoring and optimization of production rates. In addition, the model was validated using field data, demonstrating its effectiveness in a real-world application.

Subsequently, the results of the developed predictive maintenance models for electrical submersible pumps (ESPs) are presented. The ESP predictive maintenance model utilizes machine-learning algorithms to accurately predict the probability of ESP failure based on historical data. The effectiveness of the model was validated using field data, demonstrating its potential to reduce unplanned downtimes and associated costs by identifying potential failures before they occur.

Finally, the results of data-driven optimization for steam injection and water flooding projects are presented. The developed models aim to optimize the injection rates to improve oil recovery and maximize net present value. The models were trained using reinforcement-learning algorithms, and their ability to learn optimal policies for injection rates was demonstrated. The models were validated using field data, demonstrating their potential as valuable tools for optimizing the production rates in the oil and gas industry.

8.2 Results and discussion of the virtual flow meter

In this section, the models created for predicting sensor data measurements are evaluated. First, a detailed exploratory data analysis, outlier removal, data transformation, and feature ranking were performed. The models were then built using symbolic regression, XGBoosting, and Conv1D with LSTM algorithms. Finally, the developed models were used to present the predicted values versus the actual values for various well datasets.

8.2.1 Comparison of the applied algorithms

In this study, real field datasets were used to implement the various models. These datasets were divided into two sets: the training set and the testing set. This allowed for a comprehensive evaluation of the performance of the models, ensuring that they were tested on different subsets of the training data. The time series split technique was used to create these sets, wherein the future timeline data were excluded from the testing set. All the testing metrics were calculated based on the testing set. The training set was used for training and validation purposes. A 10-fold cross-validation method was used for the validation process. Utilization of the testing set and validation method enabled a thorough analysis of the algorithms implemented in this study.

When comparing different models, it is important to use objective metrics to quantify their performances. Three commonly used metrics are mean absolute error (MAE), mean squared error (MSE), and R-squared. These metrics were used to evaluate the accuracy of a model in predicting outcomes on a test dataset, which is a set of data that was not used during the model development process. In addition to the aforementioned three metrics, it is common to use a cross plot to compare the predicted values of each model with the actual values in the test data set. A more detailed explanation of the testing metrics and cross plots is provided to demonstrate their significance in model evaluation.

The mean absolute error (MAE) is a metric that calculates the average of the absolute differences between the actual and predicted values. It measures the average magnitude of the errors in the predictions and provides a measure of how close the predictions are to the actual values. A lower MAE indicates better model accuracy.

The mean squared error (MSE) is another metric that calculates the average of the squared differences between the actual and predicted values. This

metric is similar to MAE, but it penalizes larger errors more heavily than smaller ones. A lower MSE also indicates a better accuracy for the model.

R-squared (R^2) is a metric that measures the fit of predicted values to actual values. It is a statistical measure of the proportion of the variance in the dependent variable that is predictable from the independent variable(s). R2 ranges from 0 to 1, with a higher value indicating a better fit. A value of 1 means that the model perfectly predicts the outcome, while a value of 0 means that the model does not explain any of the variation in the outcome.

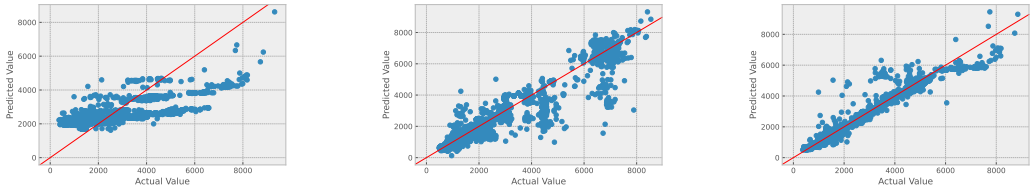
These metrics are important for the scientific evaluation of the effectiveness of a model because they provide objective measures of its performance. By comparing the MAE, MSE, and R-squared values of different models, scientists can determine which model is better suited for predicting outcomes in a particular domain.

Finally, a cross-plot is a graphical representation of the relationship between two variables. In scientific research, cross-plots are often used to assess the performance of different models by comparing the predicted values to the actual values. By comparing the distribution of the points on the graph to the 45-degree line, researchers can assess the accuracy of each model. A model with all points lying on the 45-degree line would have perfect accuracy, while a model with points scattered randomly around the graph would have poor accuracy.

Tables 8.1 and 8.2 present a comparison of the models built using symbolic regression, extreme gradient boosting, and a convolutional LSTM neural network to predict the liquid rate and basic sediment, and water (BS&W) in oil wells. The convolutional-LSTM models provided the best results for both the liquid rate and the BS&W prediction. However, the XGBoosting algorithm also performed well in both cases, particularly for BS&W prediction.

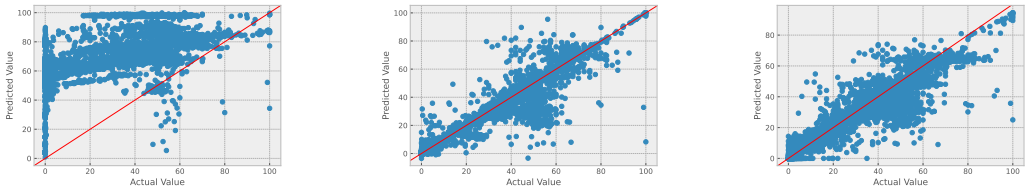
The Convolutional-LSTM model outperformed the other two models in liquid rate prediction, with a mean absolute error (MAE) of 0.113, the highest R-squared value of 0.95, and a mean squared error (MSE) of 0.041. The XGBoosting model also showed good results, with an MAE of 0.127, an R-squared value of 0.92, and an MSE of 0.046. However, the symbolic regression model had the highest MAE of 0.551, the lowest R-squared value of 0.44, and the highest MSE of 0.243.

For BS&W prediction, both the Convolutional-LSTM and XGBoosting models produced good results, with MAE values of 0.036 and 0.031, R-squared values of 0.905 and 0.908, and MSE values of 0.004 and 0.005, respectively. The symbolic regression model had the highest MAE (0.41 and the highest MSE (0.21). The negative predictions produced by the symbolic regression



a) SR cross-plot b) XGBoosting cross-plot c) CNN-LSTM cross-plot

Figure 8.1: Cross-plots for liquid production prediction using various algorithms



a) SR cross-plot b) XGBoosting cross-plot c) CNN-LSTM cross-plot

Figure 8.2: Cross-plots for basic sediments and water cut prediction using various algorithms

model for some data points made it impossible to calculate the R-squared value for this model.

Figures 8.1 and 8.2 show cross plots of the predicted versus actual values for the three algorithms used for the liquid rate and BS&W prediction, respectively. As previously mentioned, these plots provide a visual representation of the performance of each model and illustrate the closeness of the predicted values to the actual values.

Table 8.1: Models comparison for liquid rate prediction

Metric	Symbolic Regression	XGBoosting	CNN-LSTM
MAE	.551	0.137	0.133
R-squared	0.44	0.92	0.95
MSE	0.243	0.046	0.041

Table 8.2: Models comparison for basic sediments and water prediction

Metric	Symbolic Regression	XGBoosting	CNN-LSTM
MAE	—	0.031	0.036
R-squared	—	0.908	0.905
MSE	—	0.005	0.004

8.2.2 Deep learning results

Among the constructed models, those using CNN-LSTM showed good results on the test set. Figures 8.3 and 8.4 demonstrate the effectiveness of the model by quantitatively evaluating it using datasets that were not used during the training process. These datasets include the last 30% of the nine wells, which is approximately the same as the data from the last two years.

For liquid rate prediction, the predicted values were quite accurate for more stable wells (e.g., wells 2, 5, and 6). However, for wells like 1 and 3, the results had higher errors. Similar outcomes were observed for basic sediments and water percent prediction, with good results for most of the wells, except for those showing a drastic change in BS&W percent (e.g., well 2, as shown in figure 8.4). This could be due to wells having unstable conditions and many interventions or the model's predictability not being sufficient for such wells.

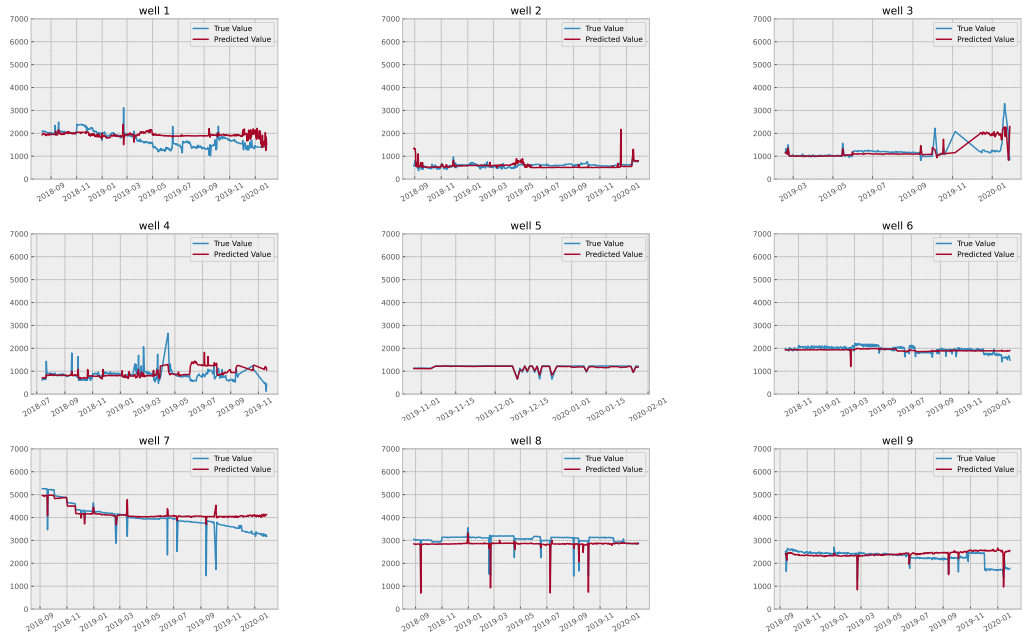


Figure 8.3: Liquid rate production per well, the predicted values and actual values

8.2.3 Discussion

The feasibility of building real-time models using machine learning approaches was demonstrated through this application. The model requires

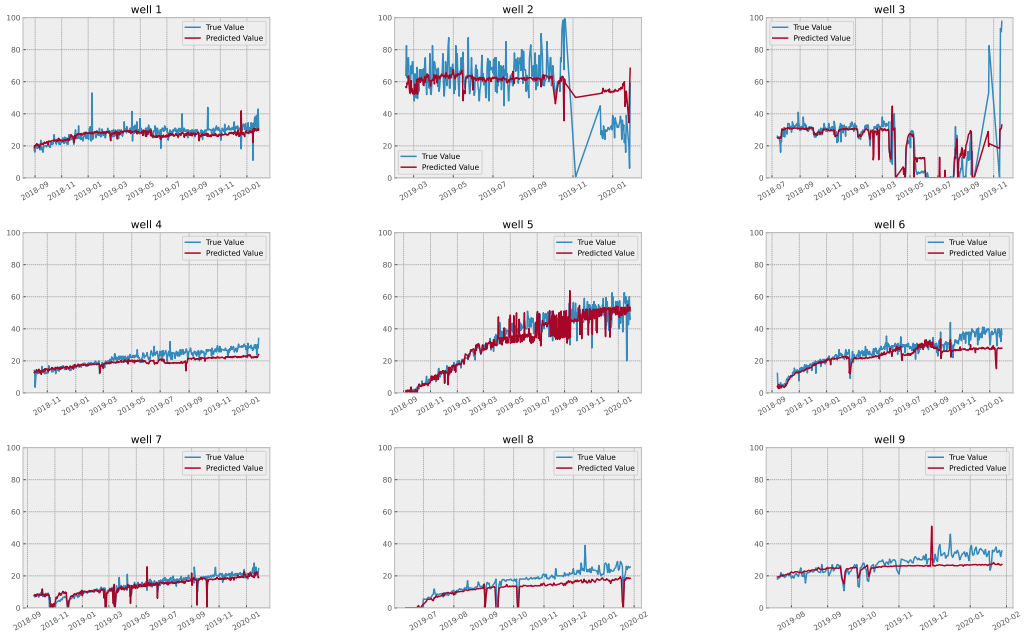


Figure 8.4: Basic sediments and water cut percent per well, the predicted values and actual values

only nine relatively straightforward measurements and provides instantaneous predictions due to the simplicity of the time series model.

Machine learning methods rely on the volume and granularity of data to develop the capability of predicting the production of an existing or new well based on past and offset production. The quality of the prediction is highly dependent on the quality of the input data. If the input is noisy and inconsistent, the prediction can be unreliable.

To ensure the versatility of the model, data from all wells were employed for training, and 70% of the data were used for model development to obtain a comprehensive representation of the entire data range. In order to check the model's generalizability, 30% of the available data were reserved for testing purposes across all well timelines. This allowed the evaluation of the model's performance on previously unseen data. The goal was to determine whether the model was making accurate predictions on new data or just fitting the training data too well. When the production performance is stable, the model predicts an R-squared value of 90%. Symbolic regression was also implemented on the dataset; however, the prediction outcome was poor.

It is concluded that the CNN-LSTM network architecture is quite prominent in the time-series analysis of sensor data. It provides good results for challenging sequential datasets with minimal feature engineering. The 1D-CNN layers were able to automatically extract features and create informative

representations of the time series. They are highly noise-resistant models and can extract informative deep features that are independent of time. Additionally, LSTM is effective at extracting patterns in the input feature space, where the input data spans over long sequences. The unique gated architecture of LSTM makes it well-suited for tasks that require the manipulation of long-term memory, particularly in sequential sensor datasets. Therefore, it is recommended that research studies utilize this network architecture in order to effectively handle sequential sensor datasets.

8.3 Results of predictive maintenance of ESPs

In this section, the outcomes of applying the predictive maintenance model to ESPs are comprehensively analyzed. The assessment of the model's performance is based on key metrics including precision, recall, and F1-score. Precision, indicating the accuracy of positive predictions, ensures the model's capability to correctly identify genuine maintenance events among its positive forecasts. It plays a pivotal role in minimizing unnecessary maintenance actions and associated costs related to false alarms. Similarly, recall evaluates the model's proficiency in identifying all relevant instances of maintenance events, with a higher recall value signifying the model's effectiveness in capturing a significant portion of actual maintenance occurrences. Additionally, the F1-score, a harmonic mean of precision and recall, offers a balanced evaluation of the model's overall performance.

Moreover, the results in the context of the specific characteristics and requirements of ESP systems are discussed to highlight the strengths and limitations of the models. The goal of this section is to provide an insightful evaluation of the performance of the model and highlight its potential for practical applications in the field of ESP predictive maintenance.

8.3.1 Using XGBoosting

To reduce the false alarms in our model, the raw sensor data have been pre-processed. Then, the "standardized" and "cleaned" time-series data with their moving difference are entered into feature engineering transformation through the use of PCA. Finally, a machine learning model is used to classify the operating point conditions.

The upcoming results are reported in two different ways. Firstly, validation results are reported using a 10-fold validation. Along with model training, model validation intends to locate an ideal model with the best execution. The model's performance is optimized using the training and validation datasets. Therefore, the Receiver operating characteristic (ROC) curves are reported for the 10 folds of the dataset and their mean value. On an ROC curve, the x-axis represents the false positive rate (FPR), while the y-axis represents the true positive rate (TPR). Then, the generalization performance of the model is tested using the testing set. The test dataset remains hidden during the model training and model performance evaluation stages. In this regard, a precision-recall curve is used.

Figure 8.5 displays the ROC curves. To interpret the ROC curves, a single score can be assigned to a classifier model using the "ROC area under curve"

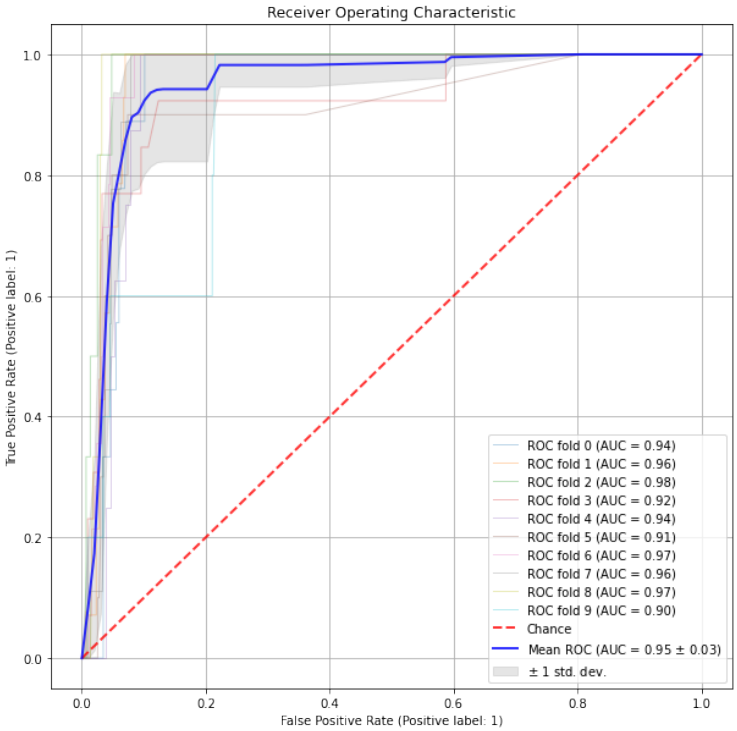


Figure 8.5: ROC for the proposed model

(AUC), which is obtained by integrating the area under the curve. The score ranges between 0.0 and 1.0, with 1.0 indicating a perfect classifier. Fig. 8.5 presents the ROC curves for the model with 10-fold validation sets and its mean curve. The mean ROC AUC value is 0.95.

As mentioned earlier, the second process is testing the proposed model against a testing set using the precision-recall curve (PRC), which is a valuable diagnostic tool particularly when classes are very imbalanced. The PRC trade-off between a classifier's precision, a measure of result relevancy and recall, a measure of completeness for every possible cut-off is depicted. Figure 8.6 shows a precision recall curve PRC for the Pre-Workover and Workover class.

The ESP dataset is unbalanced. For this reason, it is important to check the precision and recall for each class of the pumping conditions for better evaluation of the classifier. From Figure 8.6 and table 8.3, The precision and the recall for pre-workover and workover condition is less than those in normal conditions. This is mainly due to a higher number of data points supporting the normal labelled status. This is an effect of using an unbalanced dataset. One approach to address the imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples do not add any new information

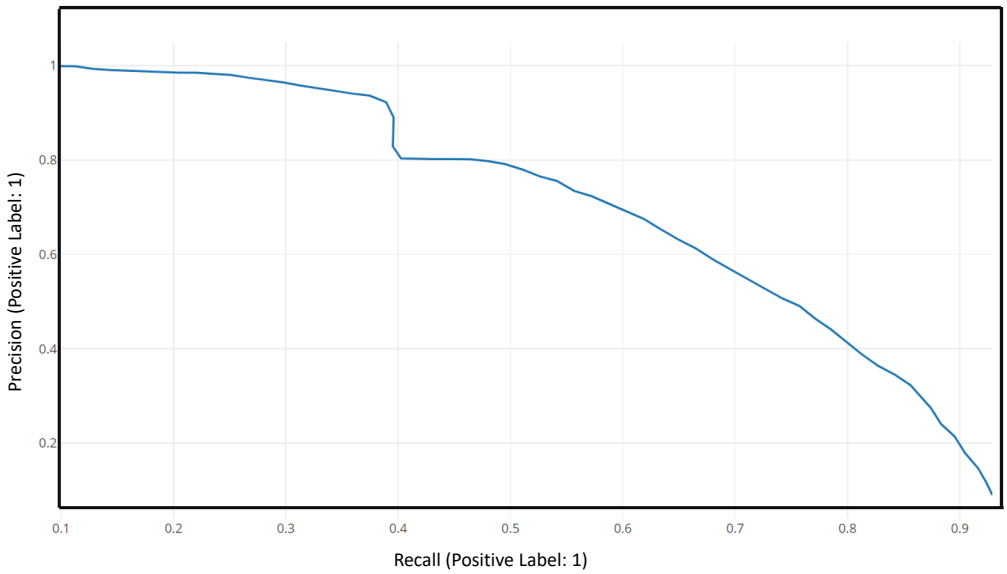


Figure 8.6: Precision Recall Curve

Table 8.3: Precision, Recall and F1 score

	Precision	Recall	F1-score	Support
Normal	0.99	1.00	1.00	101,726
seven days or less pre-event	0.80	0.60	0.685	0.604

to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the Synthetic Minority Oversampling Technique (SMOTE). This can be part of further work. However, such procedures are inherently dangerous because they may result in overfitting of the model.

8.3.2 The usage of 1D-CNN

The 1D-CNN architecture is applied to the time-series sensor data from the ESP system, in order to learn the patterns that lead to detect failures and degradation of the system. The input to the 1D CNN model is typically a set of time-series data such as vibration signals, temperature signals, and pressure signals, among others mentioned in chapter 5. These signals are processed through multiple convolutional layers, which learn to extract meaningful features from the input data. The extracted features are then fed into one or more fully connected layers, which make the final prediction of the system's state.

In deep learning predictive maintenance applications, the sequence of look back points refers to the number of past observations or measurements that are used as inputs to predict the future state of a system or equipment. These look back points allow the model to learn the patterns and trends in the data that can indicate an impending failure or degradation of the system, and make predictions based on those patterns. The appropriate number of look back points depends on the application itself and should be determined through experimentation. Table 8.4 presents the f1 scores for the predictive maintenance model using look back periods of 10, 20, and 30 days. The results show that the model achieved f1 scores of 0.7, 0.76, and 0.77, respectively, indicating improved performance with longer look back periods.

Table 8.4: F1 Scores for Different Look Back Days

Look Back Days	**F1 Score**
20	0.7
30	0.76
40	0.77

As previously mentioned, the evaluation results for a classification problems are obtained based on the precision, recall, and F1-score metrics. The results showed that the model was able to perform well in classifying both the normal and seven days or less pre-event conditions.

The precision for the normal class was 0.89, which means that the model correctly classified 89% of the instances as normal. The recall for the normal class was 0.78, which means that 78% of the actual normal instances were correctly classified by the model. The F1-score for the normal class was 0.83, which is the harmonic mean of precision and recall and provides a balance between the two.

For the seven days or less pre-event class, the precision was 0.79, which means that 79% of the instances that were classified as seven days or less pre-event were actually seven days or less pre-event. The recall was 0.70, which means that 70% of the actual seven days or less pre-event instances were correctly classified by the model. The F1-score was 0.75, which again provides a balance between precision and recall.

One of the key advantages of using a 1D CNN for ESP predictive maintenance is that it is able to handle high-dimensional time-series data and automatically extract relevant features from the data. Additionally, the 1D CNN model is able to learn from large amounts of data, allowing it to capture complex patterns and relationships in the data.

Table 8.5: Performance Comparison of 1D-CNN for Seven Days or Less Pre-event

	precision	recall	f1-score
Normal	0.89	0.78	0.83
Seven Days or Less Pre-Event	0.82	0.7	0.75

Table 8.6: Performance Comparison of Vanilla LSTM and LSTM with Attention for Seven Days or Less Pre-event

Algorithm	precision	recall
Vanilla LSTM	0.8 (normal) / 0.73 (pre-event)	0.89 (normal) / 0.69 (pre-event)
LSTM with Attention	0.88 (normal) / 0.78 (pre-event)	0.86 (normal) / 0.75 (pre-event)

Another advantage of using 1D CNN for ESP predictive maintenance is that it can be trained end-to-end, meaning that the entire model is trained simultaneously and no manual feature engineering is required. This makes it easier to implement and maintain and also allows for quick adaptation to new data as it becomes available.

In conclusion, the use of 1D CNN for ESP predictive maintenance can provide significant benefits over traditional machine learning approaches. With its ability to handle high-dimensional time-series data, learn from large amounts of data, and be trained end-to-end, 1D CNN is a powerful tool for improving the accuracy and reliability of ESP predictive maintenance.

8.3.3 Vanilla LSTM and LSTM with attention for predictive maintenance

In this study, the performance of Vanilla LSTM and LSTM with attention models for predictive maintenance of ESPs (Electrical Submersible Pumps) are evaluated. The results were compared based on prediction accuracy, precision, recall, and F1-score. The results obtained are shown in the table 8.6.

According to Table 8.6, the precision of the Vanilla LSTM model for normal and seven days or less before an event are 0.80 and 0.73, respectively, while the recall is 0.89 and 0.69. On the other hand, the LSTM with Attention model has a precision of 0.88 and 0.78 for the normal and pre-event conditions, respectively, with a recall of 0.86 and 0.75. The results indicate that the LSTM with Attention model has better precision and recall scores than the Vanilla LSTM model for the pre-event condition, but the Vanilla LSTM model performs slightly better for the normal condition in terms of precision.

Hence, these results indicate that the LSTM with attention model was better at correctly identifying the instances of failure and hence, was able to predict the maintenance needs of ESPs more accurately. The higher precision and recall values of the LSTM with attention indicate that the model was able to correctly identify the instances of failure without raising many false alarms and was able to correctly predict most of the actual failures.

8.3.4 Discussion of the findings

The findings of this study suggest that, LSTM with attention emerged as the best performing model for predictive maintenance of electrical submersible pumps compared with Vanilla LSTM, 1D CNN, and PCA-XGBoosting. The attention mechanism in LSTM helps the model to selectively focus on the most relevant features and information, thus enhancing its performance compared to the traditional Vanilla LSTM model.

The improved performance of LSTM with attention can be attributed to its ability to dynamically weigh and allocate attention to different parts of the input sequence, which is crucial for analyzing sequential data in predictive maintenance. This ability enables the model to effectively capture complex dependencies and patterns in the data, leading to improved accuracy and reliability in predictions.

Also, the application of focal loss in predictive maintenance of electrical submersible pumps (ESPs) showed promising results. The use of focal loss helped in improving the performance of the classification model, particularly in handling the imbalanced nature of the dataset. The focal loss allowed the model to focus more on the pre-events data points, which is the minor class, resulting in better precision and recall for the failure class.

Despite LSTM with attention being the best performing model in this study, 1D CNN also showed promising results and came in second place. The 1D CNN architecture used in this study consisted of two convolutional and max pooling layers, and utilized batch normalization to aid in training. This suggests that 1D CNN is also a viable option for predictive maintenance in ESPs, though it may not be as effective as LSTM with attention. The results of this study demonstrate the potential for utilizing deep learning models in predictive maintenance and highlight the importance of choosing the right model for a specific problem. In this case, LSTM with attention was found to be the best option for predictive maintenance of ESPs, but it is possible that different models may be more effective for different types of predictive maintenance problems.

8.4 Results and discussion of steam injection optimization using RL

In this section, the results of the learning process and the optimization of the steam injection rate over time are examined. Specifically, the agent's learning curve and the optimal steam injection rate policy are of interest. To evaluate the effectiveness of the agent's policy in optimizing steam injection rates, a quasi-experimental comparison is conducted with the base case where the injection rate is held constant at 138 bbl/day. This quasi-validation approach will allow us to assess the relative performance of the agent's policy in maximizing the cumulative net present value over the production horizon.

Figure 8.7 illustrates the learning curve for our agent, where the line represents the cumulative net present value for each episode after 820 days. Large variations in the NPV of initial episodes can be observed, which can be attributed to poor approximations of the actor-network. Typically, the critic is a state-value function that evaluates the new state after each actor selection to determine whether things have improved or become worse than expected. The decisive parameter for this evaluation is called the temporal difference error (TD). As the number of episodes increases, the critic learns about the policy currently being followed by the actor. This critique takes the form of a TD error, which is the sole output of the critic and drives all learning in both the actor and critic.

Figure 8.8 shows the optimal steam injection curves for the A2C agent and the base model over time. Four specific regions can be distinguished in the optimal injection policy: from 0 to 380 days, the injection rate has slightly changed between 98 and 118 bbl/day. The second region is from 380 to 640 days where the injection rate increased from 118 bbl/day to a maximum value of 148 bbl/day. The third region (from 640 to 780) shows a steep decrease in the injection rate from 148 bbl/day to 98 bbl/day. The fourth region is from 780 to 820 showing a slight change between 98 bbl/day and 108 bbl/day. Based on Figure 8.7, and from an economic perspective, it is clear that the policy defined by the agent was able to achieve the highest net present value. However, the question that arises here is how such a policy can be justified from a physical point of view.

The main factor that explains the success of this policy is its ability to minimize the cumulative heat loss to the surrounding rocks. Figures 8.9b and 8.9a show the rate of heat loss and cumulative loss to the surrounding rocks. It is evident that the policy defined by the A2C agent to change the steam injection rate results in less heat loss to the surrounding rocks

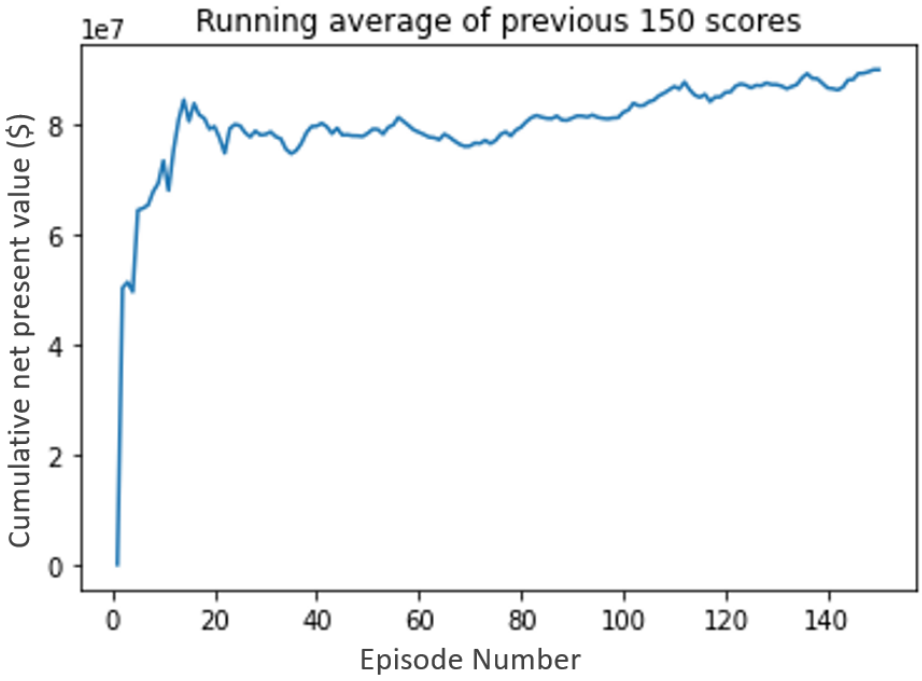


Figure 8.7: Learning curve of RL

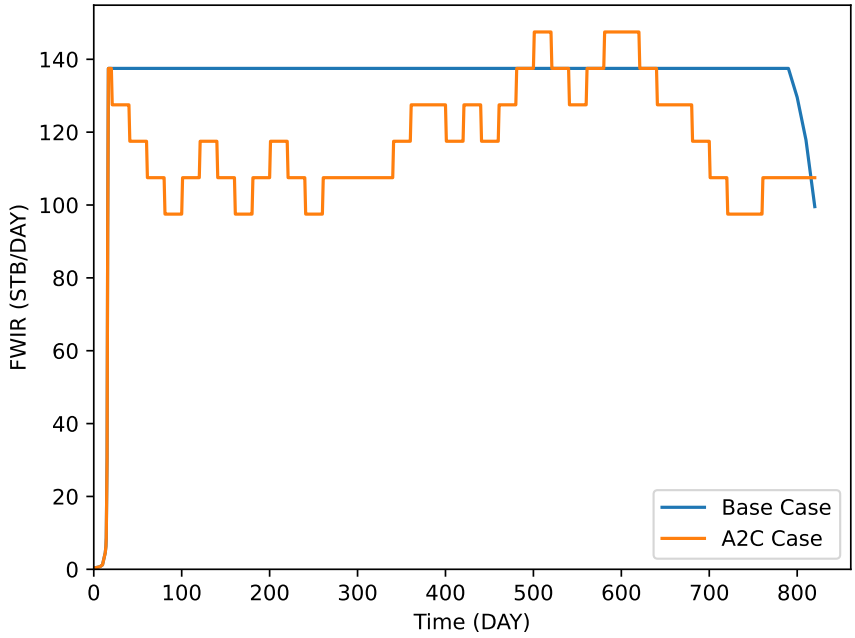


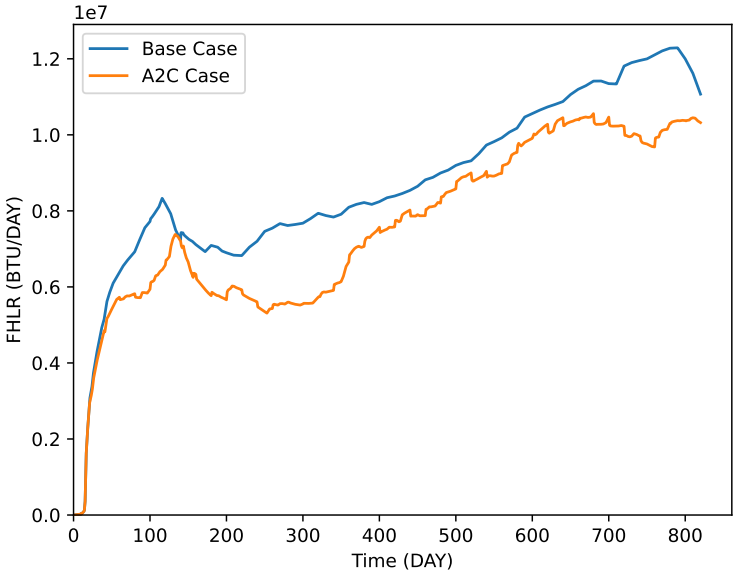
Figure 8.8: Field water injection total versus time

compared to the base case. This effect explains the success of this policy in optimizing oil recovery through water injection.

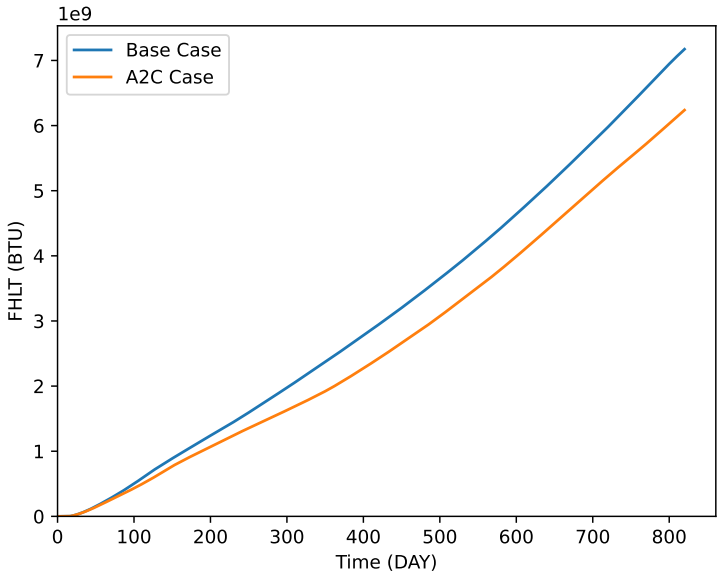
Also, it is important to study the effect of this injection policy on the produced oil and water. Figures 8.10 - 8.11 show that the total water cut and the cumulative water production are lower in the agent policy case than in the base case. This may be due to slightly smaller injection rates at the first region of the production horizon, which lead to the steam chamber growing vertically.

Figure 8.12a-8.12b show the oil rate and cumulative oil production over time. The agent policy shifted the profile of oil production rate, with the second peak moving from day 260 to day 400. In the A2C policy, this led to a reduction in the sharp decrease happening afterwards and a delay in the steam breakthrough in the near and far wells. Figures 8.13a and 8.13b show the total steam production from the near and far wells in both the A2C and base cases. While the near well started to produce steam after 160 days in the A2C case, it started after 140 days in the base case. The same happened for the far well, with an approximately 80-day difference.

To conclude, the aim of this work is to propose the application of an actor-critical reinforcement learning (RL) approach to optimize steam injection rates based on compositional modeling of fluids flowing through porous media. The objective is to find a policy capable of maximizing the cumulative net present value at the end of the production horizon by optimizing well water injection rates. The actor network's purpose is to estimate the value function by mimicking the environment's inputs and outputs' physics. The policy distribution is then updated in the critic network's suggested direction. After successive interactions between the agent (actor-critic network) and the reservoir simulation model, the agent chooses a series of actions (steam injection rate over time) aimed at maximizing the project's net present value. The results from both the base and optimum policy cases are compared from a physical perspective to provide an explanation for the increase in the net present value of the entire project. This work represents the first step in implementing a cooperative-competitive multi-agent actor-critic reinforcement learning framework to optimize multi-pad oil wells.



a) Heat losses per day versus time



b) Cumulative heat losses versus time

Figure 8.9: Field Heat losses versus time

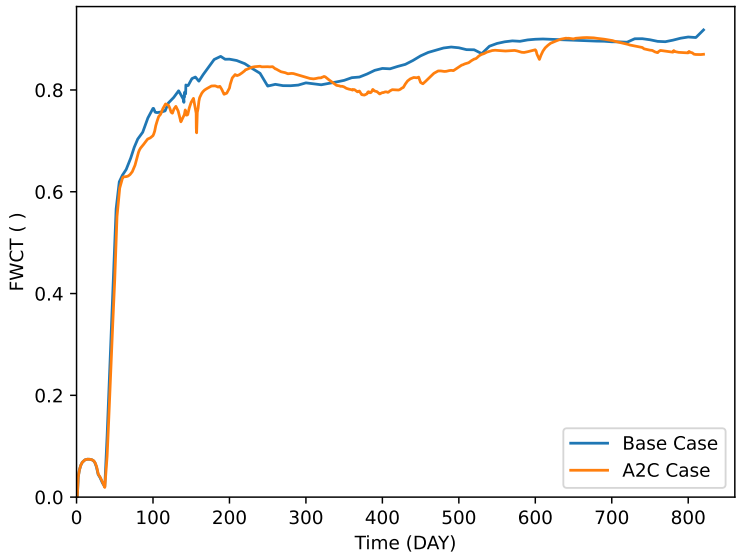


Figure 8.10: Field water cut versus time

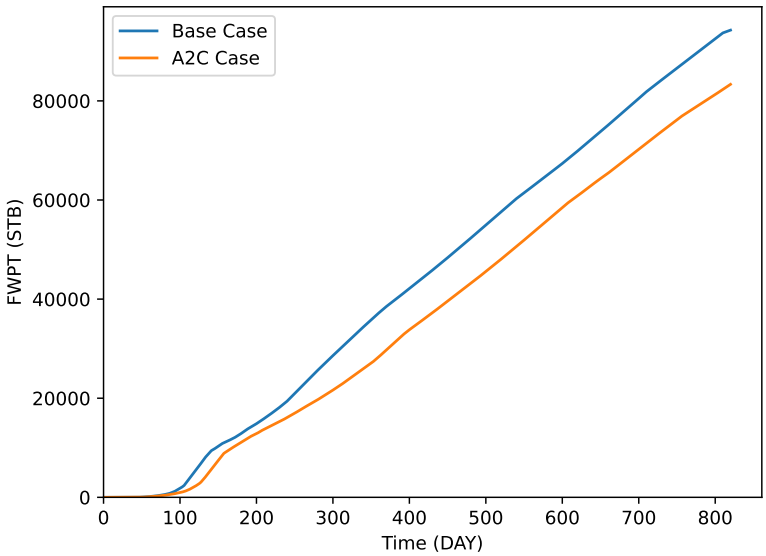
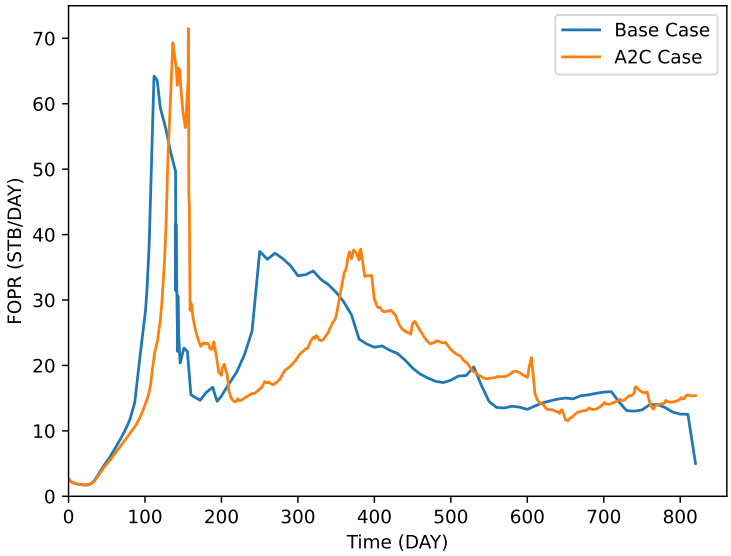
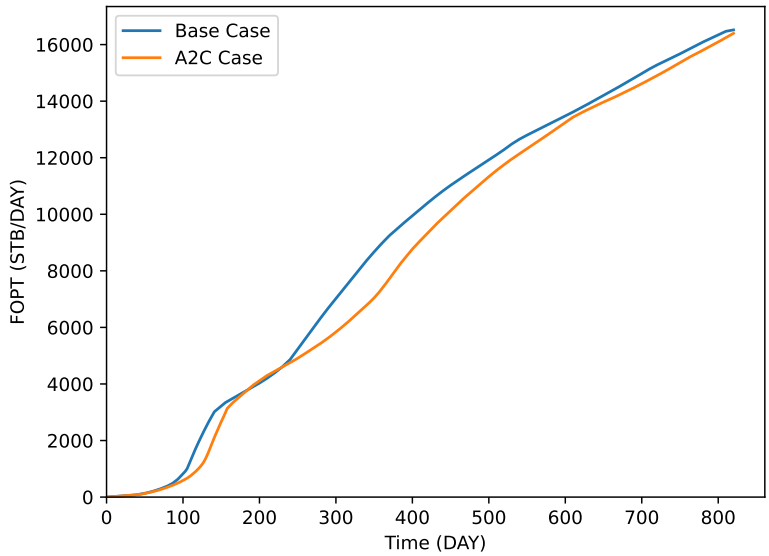


Figure 8.11: Cumulative water production versus time

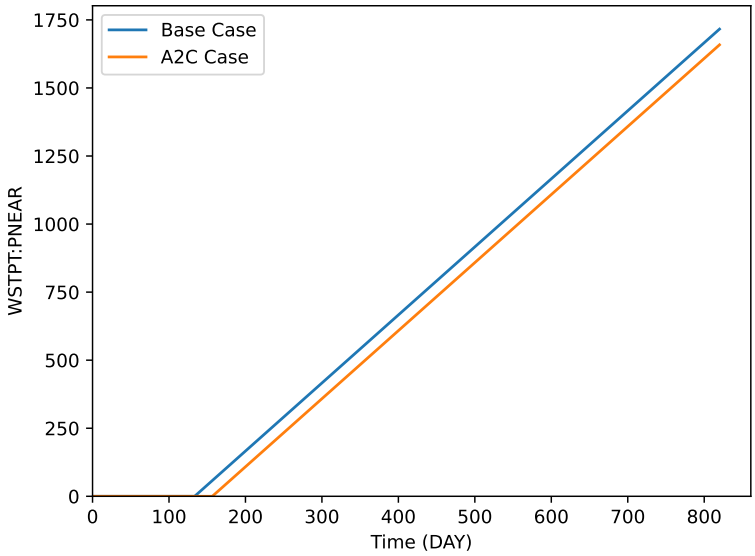


a) Daily oil production rate

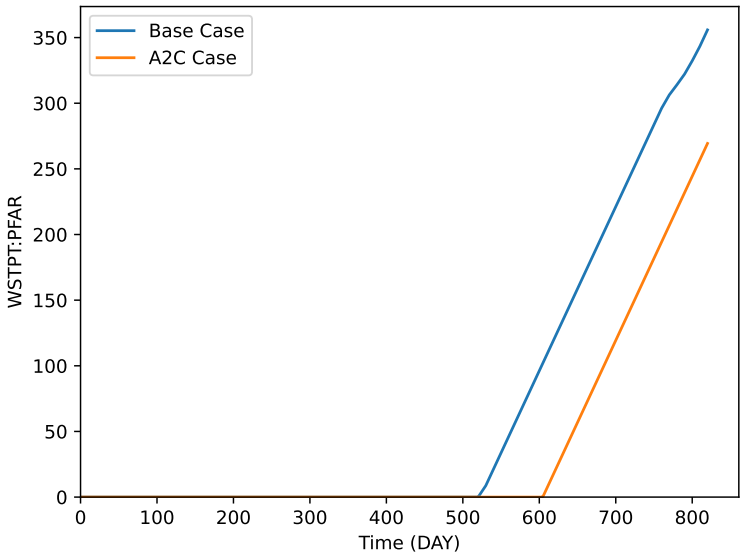


b) Cumulative oil production

Figure 8.12: Field oil production versus time



a) Steam production in the near production well



b) Steam production in the far production well

Figure 8.13: Cumulative steam production per well versus time

8.5 Results of the waterflooding optimization using multi-agent RL

The optimization of waterflooding using multi-agent deep deterministic policy gradient (DDPG) on the Egg model is a recent advancement in the field of enhanced oil recovery (EOR). The Egg model is a simplified representation of a reservoir used to study and optimize oil production processes. The results of waterflooding optimization using multi-agent DDPG on the Egg model have shown promising improvements in oil production compared to traditional methods. The optimization process has been shown to increase the total oil production while reducing the amount of injected water and the pressure drop in the reservoir. This resulted in a more efficient and cost-effective solution for enhanced oil recovery. In the following sections, the detailed results of applying Multi-agent deep deterministic policy gradient (MADDPG) to the Egg model are presented. Subsequently, a comparison between the MADDPG injection policy and the Multi-Objective Particle Swarm Optimization (MOPSO) injection policy is presented.

8.5.1 Performance of applying MADDPG on the Egg model

The purpose of this study was to evaluate the performance of the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm on a multiple-well simulation model. The results of the experiments are presented. The impact of different configurations of the critic networks' are investigated. These parameters are the hidden layer sizes and the noise standard deviation values on the training and performance of the algorithm. Finally, the performance of the MADDPG algorithm is evaluated in optimizing waterflooding based on several metrics, including cumulative oil production, water cut, and injection rate.

Experiments Results

Firstly, experiments are conducted to evaluate the efficacy of the MADDPG algorithm on a simulation egg model for water flooding by varying the configurations of the deterministic policy gradient algorithm. Those are the noise standard deviation values that were used in exploration and the critic networks' hidden layer sizes. Specifically, the impact of these parameters is investigated on the training and performance of the MADDPG algorithm and provides important insights into the optimal configurations for achieving the best results.

The noise standard deviation is a parameter that is used to incorporate randomness into the training process, which can help to prevent the agents from becoming overly reliant on a particular set of actions. Adding Noise has two different impacts. Noise can be harmful as it can lead to systematic overestimation. On the other hand, adding noise can encourage exploration by introducing randomness into the system, which is useful for discovering new information or improving performance. To evaluate the impact of different noise standard deviation values, experiments with values of 0.4, 0.3, 0.2 and 0.1 are coducted.

As shown in Figure 8.14, the configuration with a noise standard deviation of 0.2 outperforms the other two configurations, achieving a higher average reward and more consistent learning by the agents. There are several reasons why using such a moderate level of noise can lead to more stable and consistent learning:

- Exploration-Exploitation Trade-off: In any learning algorithm, there is a trade-off between exploration and exploitation. Exploration allows the agents to discover new strategies and policies, while exploitation allows them to refine and optimize their existing strategies. Adding noise to the environment can encourage the agents to explore more, without completely abandoning their existing strategies. Using a moderate level of noise strikes a balance between exploration and exploitation, allowing the agents to learn more effectively.
- Avoiding Local Optima: In a complex environment, there may be multiple local optima, which are suboptimal solutions that are still better than most other alternatives. Without enough exploration, the agents may converge to one of these suboptimal solutions and get stuck there. Adding noise to the environment can help the agents to escape from local optima and find better solutions.
- Robustness: A moderate level of noise can make the agents more robust to changes in the environment. In a dynamic environment, the optimal strategy may change over time, and the agents need to be able to adapt quickly. By learning to deal with a moderate level of noise, the agents can become more resilient to unexpected changes in the environment.

Overall, using a moderate level of noise can help to promote more stable and consistent learning by the agents in a MARL algorithm by balancing exploration and exploitation, avoiding local optima, and increasing the agents' robustness.

Aside from evaluating the effect of noise, the impact of the hidden layer size is also investigated. The critic network plays a critical role in the MADDPG algorithm, as it estimates the state and action values of the agents based on

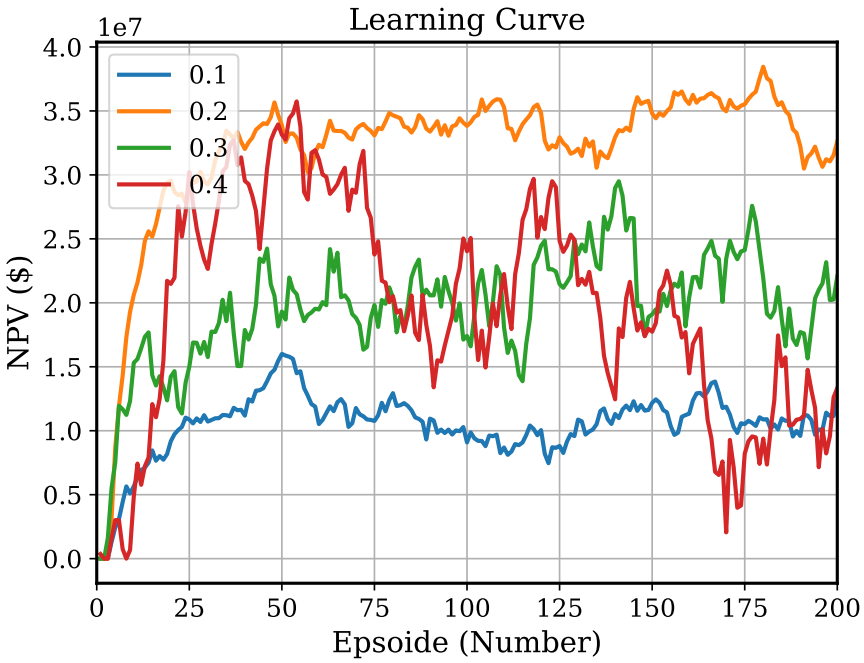


Figure 8.14: Learning Curve for MADDPG for various standard deviation values

their observed environment. The number of neurons in the critic network's hidden layers determines the complexity of the function approximation and can significantly impact the performance of the algorithm. To evaluate the effect of different hidden layer sizes, experiments with three configurations are conducted. They consist of 64, 256, and 512 neurons. As shown in figure 8.15, the experiments revealed that the 256*256 configuration of the critic networks yielded the best results in terms of learning the optimal flooding strategy. This configuration was found to be particularly effective in capturing the complex relationships between the various parameters involved in the flooding process, which, in turn, allowed the agents to learn the optimal policy more efficiently. overall, the agents achieved a stable average reward that was close to the optimal value. This indicates that the algorithm was able to learn the optimal policy for the water injection simulation environment.

Overall, our experiments demonstrate the importance of carefully selecting the configurations of the critic networks' hidden layer sizes and noise standard deviation values in the MADDPG algorithm for water flooding simulation. The results suggest that 256*256 hidden layer sizes and 0.2 noise standard deviation values can help to promote more stable and consistent learning. These insights can be valuable for optimizing the Multi-agent reinforcement learning (MARL) algorithm for different applications and do-

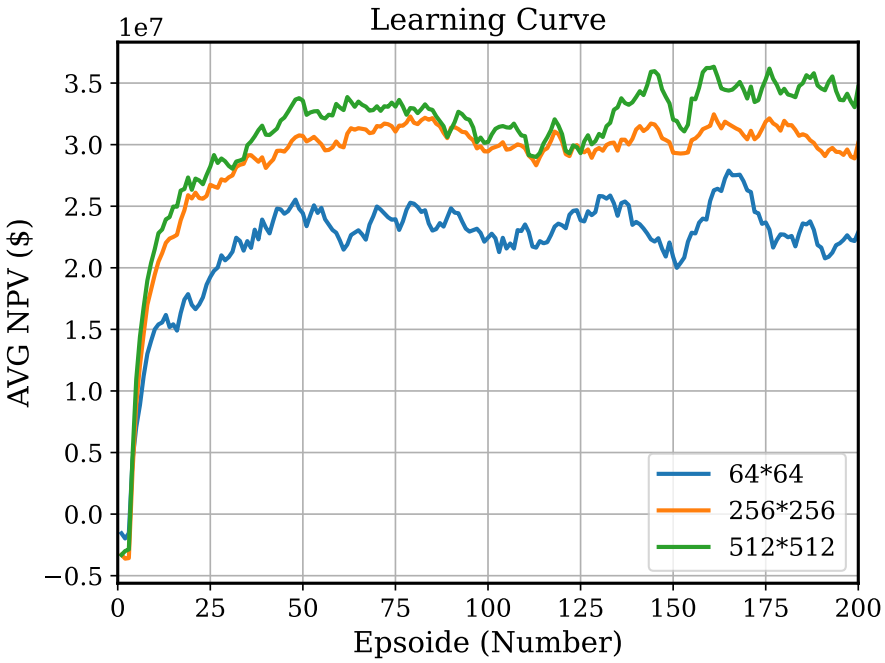


Figure 8.15: Learning Curve for MADDPG for various network structures

mains, and can contribute to the development of more effective and efficient multi-agent systems.

Performance Evaluation

In this step, the efficacy of the MADDPG algorithm is assessed in optimizing waterflooding using several key metrics, including cumulative oil production, water cut, and injection rate. The results from each testing metric will be compared against a baseline strategy in a quasi-experimental design.

The cumulative oil production, which represents the total amount of oil recovered from the reservoir during the waterflooding process, was used to evaluate the effectiveness of the optimized waterflooding strategy. The optimized strategy resulted in a significant increase in cumulative oil production, as shown in Figure 8.16. The cumulative oil production achieved with the optimized strategy was higher than that achieved with the baseline strategy.

The cumulative water production was used to evaluate the effectiveness of the optimized strategy in reducing the amount of water produced during the waterflooding process. As shown in Figure 8.17, the optimized strategy resulted in a slight reduction in water cut.

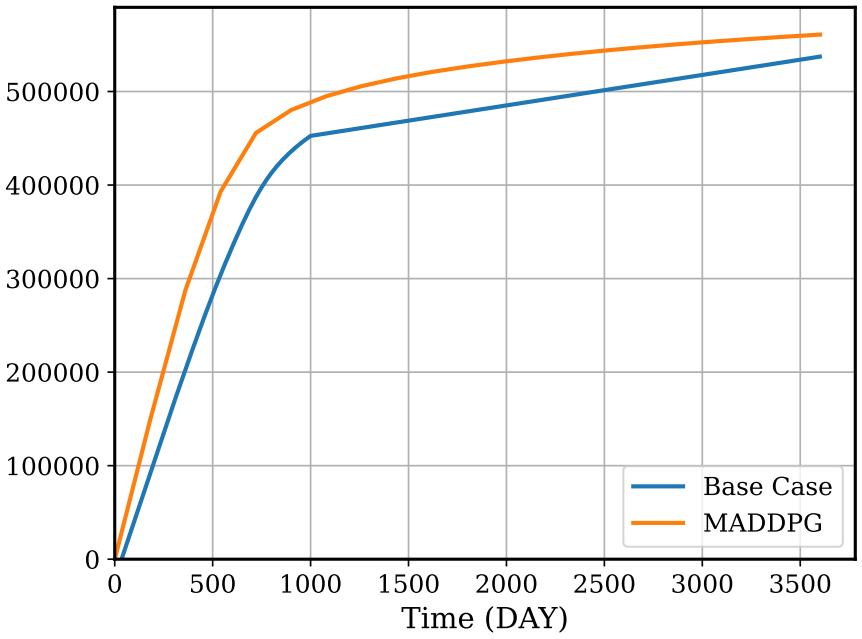


Figure 8.16: Cumulative Oil Production

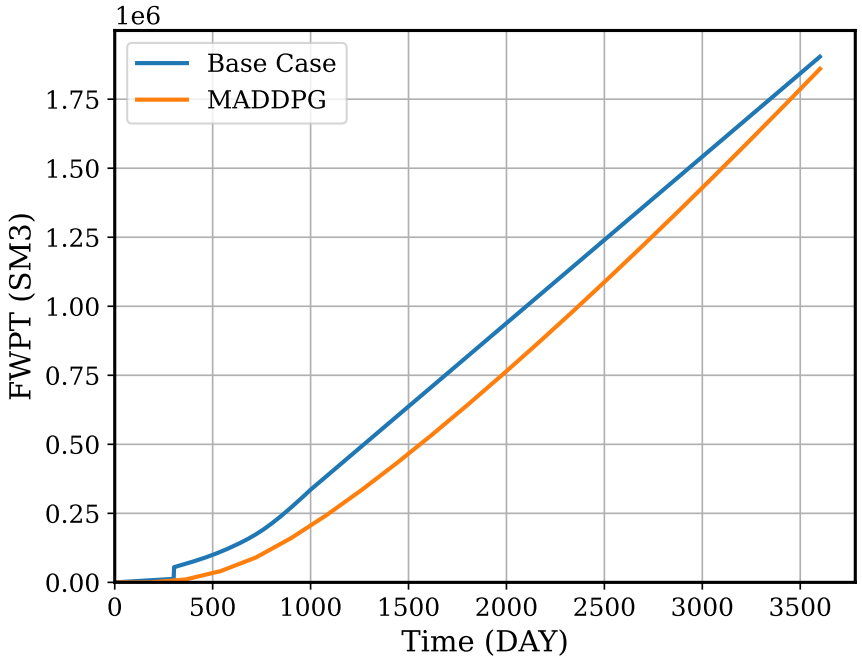


Figure 8.17: Cumulative Water Production

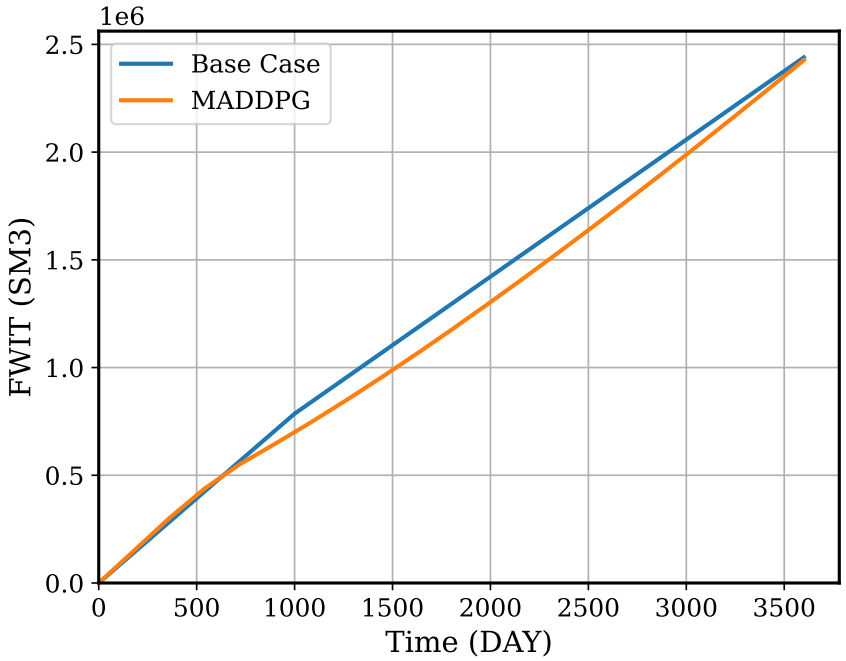


Figure 8.18: Cumulative Water Injected

The injection rate was used also to evaluate the effectiveness of the optimized strategy in maximizing oil recovery while minimizing water production. As shown in Figure 8.18, the optimized strategy resulted in a slight reduction in injection rate with respect to the baseline strategy while maintaining high oil production rates.

During the simulation, the agent learned to maximize the instant NPV of oil production by applying a high injection capacity in the initial months. Fig. 8.19 shows the agent's strategy. This strategy can be explained by the fact that injecting water into the reservoir can increase the pressure and push the oil towards the production wells, resulting in higher oil production rates. However, injecting too much water can also increase the water cut values in the production wells, which means that more water than oil is being produced. This can lead to higher operational costs.

Therefore, the agent adapted its strategy over time by decreasing the injection rates as the water cut values in the production wells increased. This is a rational decision, as reducing the injection rates can help to control the water cut values and to optimize the production of oil. However, the agent also slightly increased the injection rates again later on. This can be explained by the fact that injecting some water is still necessary to maintain the pressure in the reservoir and to avoid the premature abandonment of the wells.

Moreover, this strategy demonstrated a noticeable jump in the value of accumulative NPV during the production period. indicating it provides a more efficient sweeping policy and hence results in higher profits. The rate of produced water for each production well are demonstrated in Figs. 8.20.

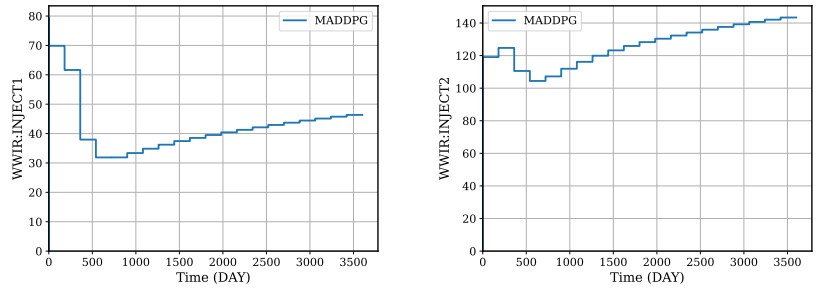
Finally, the status of oil and water distribution in the reservoir during the production related to the studied strategies can be observed every four months in Fig. 8.21. These snapshots are for the optimum policy. The figure illustrates the spatial distribution of oil and water in the reservoir over time as the injectors and producers operate according to the studied strategies. The reservoir is divided into a grid of cells, and the color of each cell in the figure represents the fraction of oil present in that cell. By observing the change in color over time, one can see how the oil is being displaced and recovered by the injected water. In conclusion, the results of waterflooding optimization using MADDPG can be promising but the specifics of the results depend on the specific details of the problem and the algorithm used.

8.5.2 Comparison between Multi-agents reinforcement learning (MADDPG) and Multi objective particle swarm optimization (MOPSO)

In this section, the performance of two optimization methods is compared, namely MADDPG algorithm and MOPSO, in the context of waterflooding optimization. First, a brief overview of MOPSO and its applications are presented. Then, the advantages and disadvantages of both algorithms in production optimization are demonstrated. Finally, the results of both algorithms are compared.

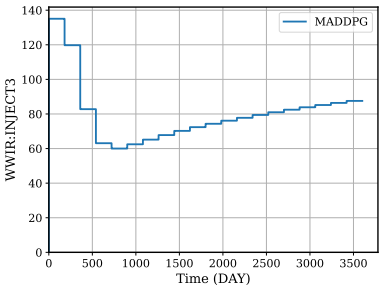
MOPSO is a widely-used optimization algorithm that has been applied in various fields, including petroleum engineering. In the context of oil recovery, MOPSO has been used for optimizing various parameters, such as well placement, injection rates, and production schedules.

Previous studies have demonstrated the effectiveness of MOPSO in optimizing waterflooding operations. For example, in a study by (M. M. Farahi et al. 2021; M. Farahi et al. 2021), MOPSO was used to optimize the water injection rate. The results showed that MOPSO was able to significantly improve the oil recovery factor compared to the initial baseline scenario. These studies demonstrated the potential of MOPSO for optimizing waterflooding operations and highlight the importance of considering multiple objectives,

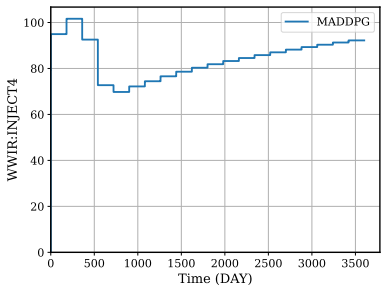


a) Well 1

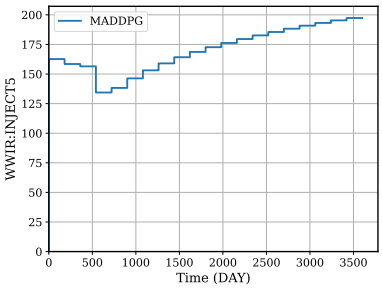
b) Well 2



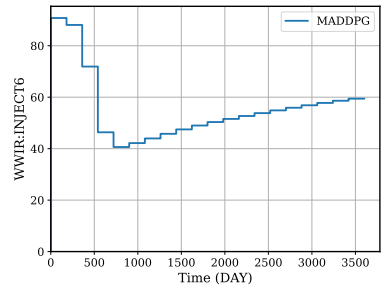
c) Well 3



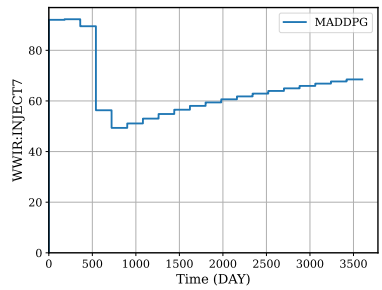
d) Well 4



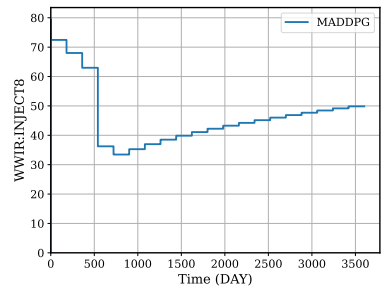
e) Well 5



f) Well 6

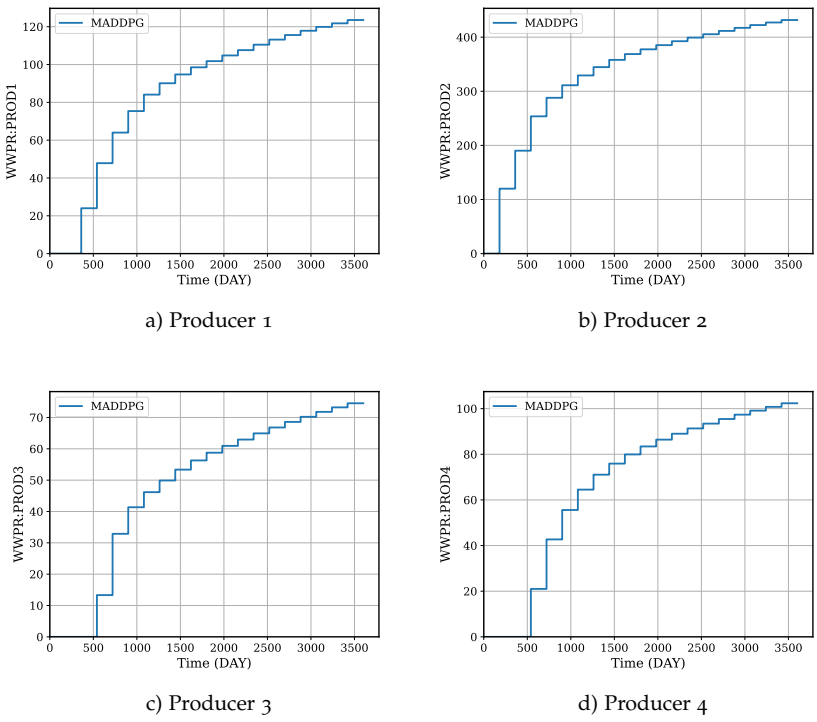


g) Well 7



h) Well 8

Figure 8.19: Optimal injection rates in m3/day for eight wells resulted from MADDPG



a) Producer 1 b) Producer 2

c) Producer 3 d) Producer 4

Figure 8.20: Water Production Per Well

such as maximizing oil production and minimizing water injection, in the optimization process.

The comparison between the results of waterflooding optimization using MADDPG and MOPSO can provide insight into the relative strengths and weaknesses of these two approaches. RL algorithms are often used in dynamic decision-making problems, where the model adapts to changing conditions over time. In waterflooding, RL was used to optimize the injection and production strategies based on the changing state of the reservoir simulation. On the other hand, PSO algorithms are optimization techniques inspired by the behavior of bird flocks and fish schools. They have been applied to various optimization problems, including waterflooding (M. M. Farahi et al. 2021). PSO algorithms typically perform well in multi-dimensional optimization problems and can find optimal solutions quickly.

To make a comprehensive evaluation, it is important to consider multiple performance metrics, as each one provides a different perspective on the effectiveness of the optimization process. The most commonly used metrics for waterflooding optimization include the total oil production and the amount of injected water. In addition, the optimum strategy for each algorithm is presented.

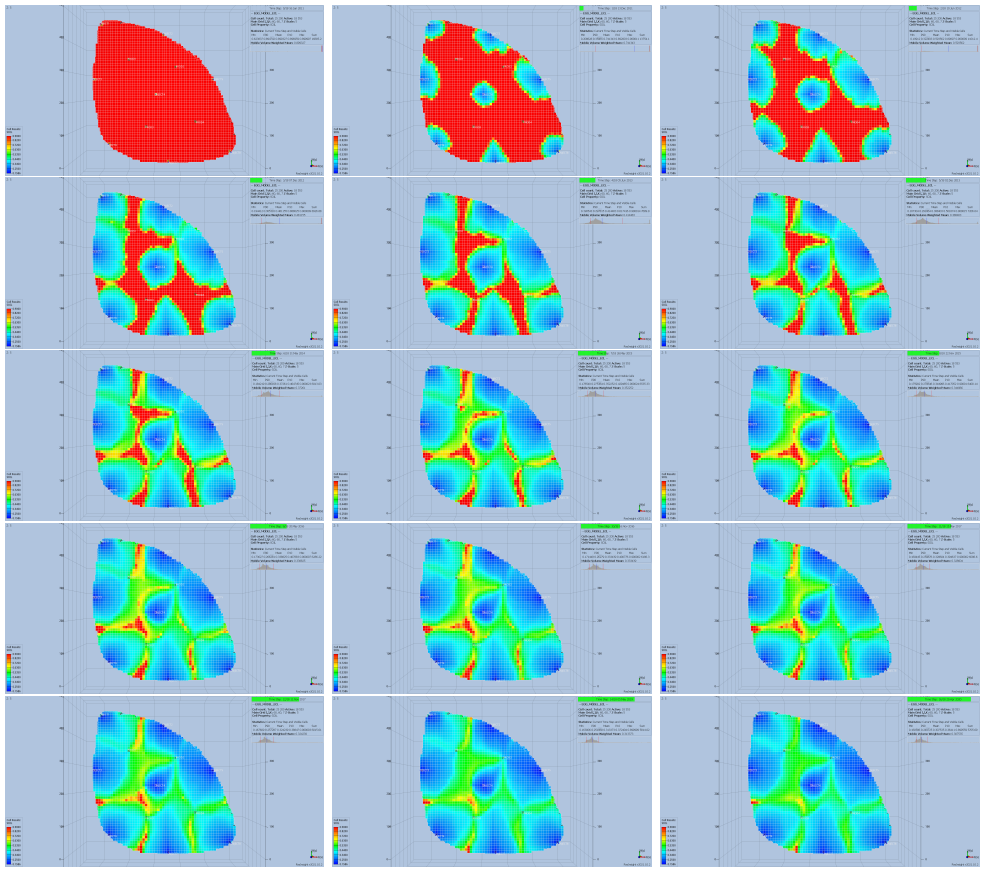


Figure 8.21: status of the reservoir during water-flooding process

In terms of total oil production, the results of waterflooding optimization using multi-agent DDPG on the Egg model have shown significant improvements compared to MOPSO methods. Fig 8.22 shows the total oil production of MOPSO vs MADDPG.

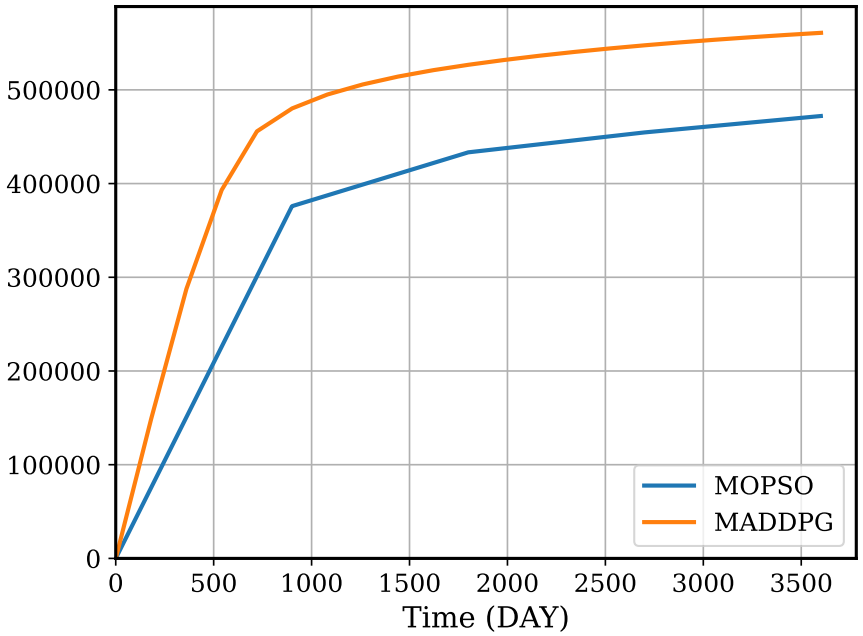


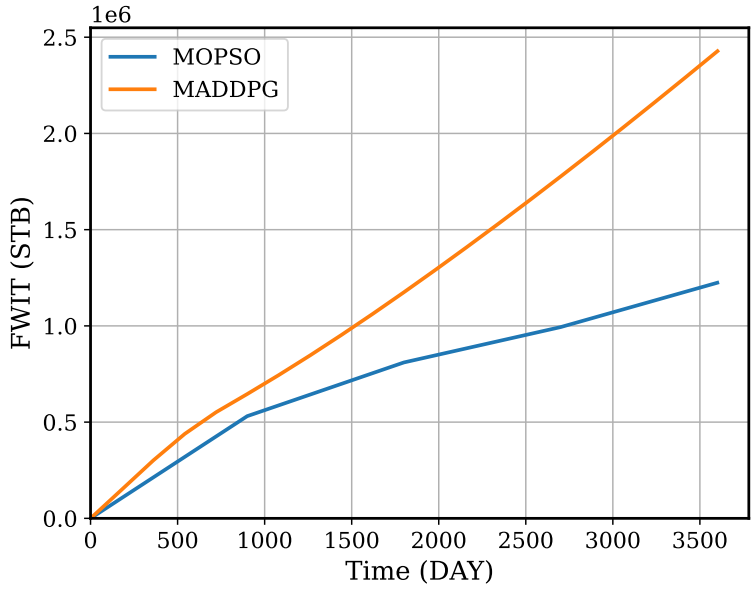
Figure 8.22: Total oil production MADDPG Vs MOPSO

The amount of injected water is another important metric to consider in this comparison. The results of waterflooding optimization using MOPSO have shown a reduction in the amount of injected and produced water compared to MADDPG. Fig 8.23a,8.23b shows a comparison of MOPSO vs MADDPG in terms of the amount of injected and produced water.

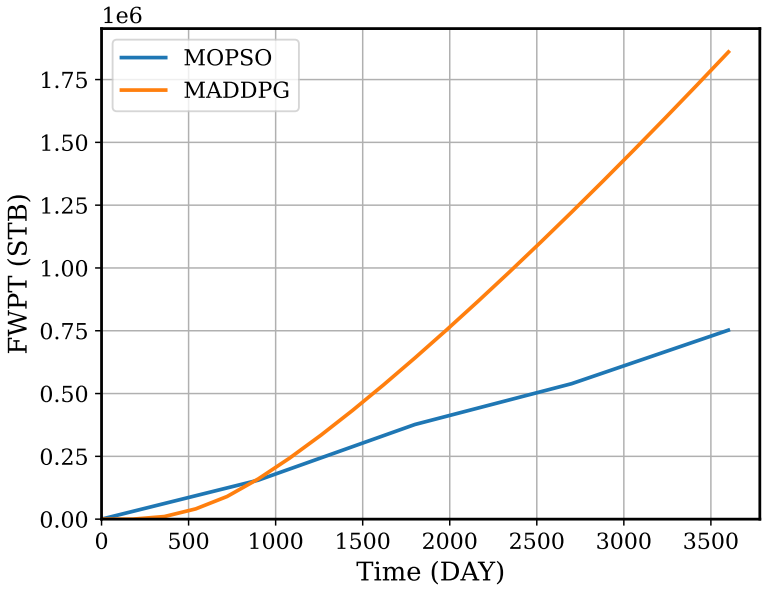
Based on the information provided, it appears that the multi-agent DDPG approach on the Egg model has led to higher total oil production compared to the MOPSO method. However, the MOPSO method resulted in a reduction in the amount of injected water compared to MADDPG. These differences suggest that the two methods have different strengths and weaknesses and may be more suitable for different optimization objectives.

It is also possible that the MOPSO method has reached a sub-optimal point in terms of the total oil production, resulting in a reduced amount of injected water. Further analysis would be necessary to confirm this hypothesis.

Therefore, comparing the NPV of the two methods can provide valuable insights into their economic feasibility and profitability. It is possible that



a) Total injected water MADDPG Vs MOPSO



b) Total produced water MADDPG Vs MOPSO

Figure 8.23: comparison MOPSO vs MADDPG in terms of the amount of injected and produced water

the method that performs better in terms of the total oil production or water injection rate may not necessarily result in the highest NPV. Fig. 8.24 shows a comparison between MOPSO and MADDPG in terms of NPV.

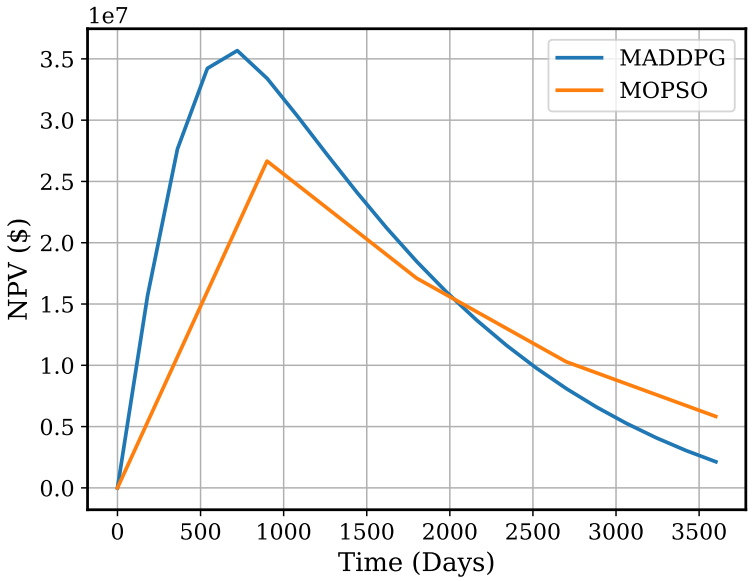


Figure 8.24: NPV comparison MOSPO Vs MADDPG

Based on the comparison of net present value (NPV) between MADDPG and MOPSO, it appears that the MADDPG method has resulted in higher economic viability and profitability than MOPSO. This suggests that MADDPG has optimized the waterflooding process more efficiently, resulting in a better balance between the production of oil and the injection of water.

However, it is important to note that the MOPSO method may have reached a sub-optimal point in terms of NPV, and it is possible that further exploration of the search space may yield better results. Additionally, the difference in NPV between the two methods may be due to the fact that they optimize multiple objectives, which may have different trade-offs.

It is important to note that these results may vary depending on the specific conditions of the reservoir and the optimization process. However, the comparison between waterflooding optimization using multi-agent DDPG and MOPSO has shown that the multi-agent DDPG approach can result in significant improvements in the total oil production, the amount of injected water, and the pressure drop in the reservoir compared to MOPSO.

In conclusion, the comparison between waterflooding optimization using multi-agent DDPG and MOPSO highlights the potential of the multi-agent DDPG approach for improving the efficiency and effectiveness of enhanced

oil recovery processes. Further research and development in this area is likely to result in even more significant improvements in the future.

8.5.3 Discussion of the Results and Findings

The use of a multi-agent deep deterministic policy gradient (MADDPG) in this optimization process allows for a more efficient and effective solution than traditional methods. This is because the multi-agent approach allows for the simultaneous control of multiple wells in the reservoir rather than controlling each well individually. The MADDPG algorithm, which is a variant of reinforcement learning, allows agents to learn from their actions and improve their decision making over time.

The MADDPG algorithm has shown a great potential for solving the complex problem of waterflooding in a cooperative competitive setting. However, the performance of these algorithms depends significantly on the noise level of the environment in which the agents operate. The effects of different noise standard deviation values on the performance of the MADDPG algorithm were investigated. The results showed that a moderate level of noise, represented by a noise standard deviation of 0.2, led to the best performance, achieving a higher average reward and a lower standard deviation. This suggests that using a moderate level of noise can promote more stable and consistent learning by the agents.

The results of waterflooding optimization using MADDPG and MOPSO showed promising results in terms of maximizing oil recovery and minimizing water production. The MADDPG algorithm showed superior performance compared to MOPSO, as it achieved a higher total oil production and a lower water cut.

One possible explanation for the superior performance of MARL could be the ability of the multi-agent system to learn and adapt to the changing conditions of the reservoir, as well as the behavior of other agents in the system. This adaptability allows agents to explore the solution space better and find optimal solutions.

Another advantage of MADDPG over MOPSO is its ability to capture the complex interdependencies between the injection and production rates of different wells. This allows the agents to optimize the waterflooding process on a global level, considering the impact of each well on the overall system performance.

Overall, the results suggest that MADDPG can be an effective approach for waterflooding optimization, with the potential to outperform traditional

optimization methods, such as MOPSO. However, it is important to note that the results and conclusions drawn from the comparison between MADDPG and MOPSO for waterflooding optimization may depend on the specific details of the problem and the algorithms used. Therefore, future research can explore the potential of other MADDPG algorithms for waterflooding optimization under uncertainties such as reservoir heterogeneity, uncertain production data, and variations in operational constraints.

9 Conclusions and Recommendations

9.1 Conclusions

A framework of data analytics applications for subsurface energy systems was used to outline various applications. A proof-of-concept project was created for each study to demonstrate the impact of these applications on subsurface energy system production. For real-time descriptive applications, liquid rate and water cut predictions using ESP sensor data were implemented and explained. A predictive maintenance model for the early diagnosis of ESP systems was introduced for predictive analytics. Finally, steam flooding and water flooding models were used for prescriptive analytics to optimize injection policies using reinforcement learning. This work elaborate on a wide range of applications of artificial intelligence on the upstream sector. In the following section, the most important points of the various applications involved are concluded:

- In the descriptive study, a new real-time data-driven model was introduced to calculate the multi-phase flow rate using ESP-assisted well data sets. The study began with a detailed exploratory data analysis at both the univariate and multivariate levels on the ESP sensor dataset, followed by feature prioritization experiments to identify the most dominant parameters affecting rate prediction. The algorithms implemented involved symbolic regression, XGBoost, and deep learning techniques, including a pipeline of a Convolutional Neural Network (CNN) and a Long Short-Term Memory (LSTM) algorithm, which were successfully used for real-time prediction of the liquid rate and water cut based on ESP sensor data. Finally, the mean absolute error, mean squared error, and R-squared were compared across the different models.
  The feasibility of building a predictor model using machine learning approaches for this specific application was demonstrated. Because the model uses only nine relatively straightforward variables, predictions are provided instantaneously, owing to the simplicity of the time-series model. However, the accuracy of the prediction is highly dependent on

the quality of the input data. If the input data are noisy and inconsistent, prediction may be unreliable. When the production performance is stable, the model predicts with an R-squared accuracy above 90%. Symbolic regression was also applied to the dataset; however, the prediction outcome was poor.
The use of the CNN-LSTM network architecture is quite prominent in the time-series analysis of sensor data. The 1D-CNN layers are capable of automatically extracting features and creating informative representations of time series. They are highly noise-resistant and can extract informative deep features that are independent of time. LSTM networks are highly effective and reliable for extracting patterns in the input feature space, where the input data span long sequences. The unique gated LSTM architecture can manipulate its memory state in such a way that long-term data relations can be stored. Therefore, the implementation of such a network architecture for future research studies using sequential sensor datasets is highly recommended.

- In predictive analytics, a study for predictive maintenance of electrical submersible pumps was presented. It processes the sensor measurements through two pipelines. The first pipeline involves dimensionality reduction techniques, projecting the dataset onto new, lower dimensions. The transformed data was then fed into the XGBoost supervised algorithm, using input features (PCA-projected features) paired with labeled outputs indicating the seven days before reported failures for ESP. Each input-output pair could be considered a "data point" for training. The model's performance was assessed on a validation set, achieving a mean AUC of 0.95 for a 10-fold validation.
  The second pipeline of the study focused on exploring deep learning techniques, specifically 1D-CNN, LSTM, and LSTM with attention, for predicting pre-failure events. The performance of the algorithms was evaluated with data shifted by different lookback periods, and the best shift was used to build all models.
  The study concluded that LSTM with attention is the best performing model for predictive maintenance of electrical submersible pumps, outperforming Vanilla LSTM, 1D CNN, and PCA-XGBoosting. The attention mechanism in LSTM allows the model to selectively focus on relevant features, enabling it to capture complex patterns and dependencies in the data. Focal loss also showed promise in improving model performance. The study highlights the potential of deep learning models in predictive maintenance and emphasizes the importance of choosing the right model for a specific problem.
- For prescriptive analytics studies, RL algorithms have been successfully able to interact with complex problems in steam-flooding and water-flooding environments. The first application studied a steam-flooding

environment using a single RL agent. The state of the environment was defined by the cumulative oil production, cumulative steam injection, and cumulative water production. The agent was able to learn an optimal injection rate policy that maximized the net present value.
The second application addressed the more complex problem of waterflooding with multiple agents, where each agent had a partial observation of the well window surrounding it. The agents learned how to manipulate the injection rate policy to maximize the net present value, resulting in improved oil recovery and reduced water production.
The use of MADDPG, a multi-agent reinforcement learning algorithm, has shown promising results in optimizing waterflooding processes, allowing for simultaneous control of multiple wells and capturing complex interdependencies. It outperformed traditional optimization methods and achieved higher oil production and lower water cut. The algorithm's performance depends on the noise level of the environment, and moderate noise levels can promote more stable and consistent learning by agents.

9.2 Recommendations

The application of machine learning in the digital oilfield holds significant potential for further optimizing the efficiency and performance in the oil and gas industry. However, ongoing research is crucial for further advancement. The proposed framework of descriptive, predictive, and prescriptive modeling can serve as a guide for future studies.

Descriptive modeling has been used to expand the application of virtual sensing, particularly in the context of virtual flow metering. The integration of data exploration and machine learning algorithms has significantly improved the technology and empowered the virtual flow meters. This framework can be applied to other artificial lift systems that provide data to create virtual flowmeters.

Predictive modeling is a crucial aspect of data analytics in the digital oilfield, and predictive maintenance is considered one of its most critical applications. Deep learning performs a crucial role in predictive maintenance applications that handle complex and imbalanced datasets for machinery failure prediction. This is because deep learning algorithms can extract complex features and patterns from unstructured or high-dimensional data, enabling accurate predictions of equipment failure. Moreover, deep learning algorithms can effectively learn from unbalanced datasets, which allows them to correctly identify rare events, and thereby enhance the accuracy of predictive

maintenance applications. Researchers can also focus on exploring new algorithms, such as transfer learning, which can further improve the accuracy of predictions. Additionally, the application of hybrid systems for system dynamics identification and data-driven modeling can be combined with physics-informed machine learning algorithms to predict critical parameters for specific flow phenomena, such as liquid loading, or to predict pump problems.

Prescriptive modeling, which involves the use of machine learning algorithms to make decisions and control processes, also holds great potential in the digital oilfield. Deep reinforcement learning has shown promising results in solving complex decision-making problems in physics and engineering, but its application in the digital oil field remains to be fully explored. Future research can focus on addressing the challenges posed by high-dimensional states in reinforcement learning as well as the application of these algorithms in controlling fluid flow.

In conclusion, the proposed framework of descriptive, predictive, and prescriptive modeling represents a significant step forward in the application of data analytics in the digital oilfield. By integrating these modeling approaches, it is possible to develop more accurate and efficient solutions for monitoring and controlling oil and gas operations.

The continued exploration and development of new algorithms and techniques has the potential to further optimize the efficiency and performance of the oil and gas industry. The use of machine learning can lead to improvements in various areas, including production optimization, equipment maintenance, and safety management.

Furthermore, the success of machine learning in the digital oilfield can serve as a model for other industries seeking to incorporate data-driven decision making. As the amount of data generated by industrial processes continues to grow, the need for sophisticated machine-learning algorithms and models will increase. By investing in this technology, subsurface energy production industries can improve their efficiency, productivity, and profitability while reducing their environmental impacts and ensuring the safety of their workers.

Abbreviations

CHP	Casing head pressure
CNN	Convolutional neural network
Current	Variable speed drive output current
DDPG	Deep deterministic policy gradient
EDHF	Electrical downhole failures
ESP	Electrical submersible pump
FLP	Flow line pressure
FRQ	Pump frequency
LSTM	Long short term memory algorithm
MADDPG	Multi-agent deep deterministic policy gradient
MAE	Mean absolute error
MARL	Multi-agent reinforcement learning
MDHF	Motor downhole failures
MDP	Morkov decision process
MSE	Mean squared error
MOPSO	Multi-Objective Particle Swarm Optimization
MT	Motor temperature
NN	neural network
NPV	Net present value
OU	Ornstein-Uhlenbeck
PCA	Principle component analysis
PIP	Pump intake pressure
PDP	Pump discharge pressure
RL	Reinforcement learning
RNN	Recurrent neural network
ROC	Receiver operating characteristic
SVM	Support vector machine
SRP	Sucker rod pump
VFM	Virtual flow meter
WHP	Well head pressure
WHT	Well head temperature
WOR	water-oil ratio
XGBoosting	Extreme gradient boosting

Bibliography

Abdalla, Ramez Maher, Mahmoud Abu El Ela, and Ahmed El Banbi (2020). "Identification of Downhole Conditions in Sucker Rod Pumped Wells Using Deep Neural Networks and Genetic Algorithms (includes associated discussion)." In: *SPE Production & Operations* 35.02, pp. 435–447. ISSN: 1930-1855. DOI: 10.2118/200494-PA.

Abdelaziz, Mohannad, Rafael Lastra, and J. J. Xiao, eds. (2017). *ESP Data Analytics: Predicting Failures for Improved Production Performance*. Vol. Day 3 Wed, November 15, 2017. Abu Dhabi International Petroleum Exhibition and Conference. DOI: 10.2118/188513-MS.

Abdeljaber, Osama, Onur Avcı, Mustafa Serkan Kiranyaz, Boualem Boashash, Henry A. Sodano, and Daniel J. Inman (2018). "1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data." In: *Neurocomputing* 275, pp. 1308–1317.

Abdeljaber, Osama, Onur Avcı, Serkan Kiranyaz, M. Gabbouj, and Daniel J. Inman (2017). "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks." In: *Journal of Sound and Vibration* 388, pp. 154–170.

Abdeljaber, Osama, Sadok Sassi, Onur Avcı, Serkan Kiranyaz, Abdelrahman Aly Ibrahim, and Moncef Gabbouj (2019). "Fault Detection and Severity Identification of Ball Bearings by Online Condition Monitoring." In: *IEEE Transactions on Industrial Electronics* 66, pp. 8136–8147.

Abhulimen, Kingsley Eromoses, K. E. Abhulimen, and A. D. Oladipupo (2023). "Modelling of liquid loading in gas wells using a software-based approach." In: *Journal of Petroleum Exploration and Production Technology* 13.1, pp. 1–17. ISSN: 2190-0558. DOI: 10.1007/s13202-022-01525-x.

Adesanwo, Moradeyo, Tommy Denney, Sony Lazarus, and Oladele Bello, eds. (2016). *Prescriptive-Based Decision Support System for Online Real-Time Electrical Submersible Pump Operations Management*. Vol. All Days. SPE Intelligent Energy International Conference and Exhibition. DOI: 10.2118/181013-MS.

Airaldi, Filippo, Bart De Schutter, and Azita Dabiri (2022). *Learning safety in model-based Reinforcement Learning using MPC and Gaussian Processes*. DOI: 10.48550/ARXIV.2211.01860. URL: https://arxiv.org/abs/2211.01860.

Akhiiartdinov, Anvar, Eduardo Pereyra, Cem Sarica, and Jose Severino, eds. (2020). *Data Analytics Application for Conventional Plunger Lift Modeling and Optimization*. Vol. Day 1 Tue, November 10, 2020. SPE Artificial Lift Conference and Exhibition - Americas. DOI: 10.2118/201144-MS.

Robust Data Driven Well Performance Optimization Assisted by Machine Learning Techniques for Natural Flowing and Gas-Lift Wells in Abu Dhabi (Oct. 2020). Vol. Day 4 Thu, October 29, 2020. SPE Annual Technical Conference and Exhibition. D041S046R002. DOI: 10.2118/201696-MS. eprint: https://onepetro.org/SPEATCE/proceedings-pdf/20ATCE/4-20ATCE/D041S046R002/2476357/spe-201696-ms.pdf. URL: https://doi.org/10.2118/201696-MS.

AlAjmi, Mohammed D., Abdulazeez Abdulraheem, Abdulrahman T. Mishkhes, and Mubarak J. Al-Shammari, eds. (2015). *Profiling Downhole Casing Integrity Using Artificial Intelligence*. Vol. Day 1 Tue, March 03, 2015. SPE Digital Energy Conference and Exhibition.

Alhashem, Mayadah, ed. (2020). *Machine Learning Classification Model for Multiphase Flow Regimes in Horizontal Pipes*. Vol. Day 2 Tue, January 14, 2020. IPTC International Petroleum Technology Conference. DOI: 10.2523/IPTC-20058-Abstract.

Ali, S. M. Farouq and Robert F. Meldau (1979). "Current Steamflood Technology." In: *Journal of Petroleum Technology* 31, pp. 1332–1342.

Application of Data Mining Tools for Analysis and Prediction of Hydraulic Fracturing Efficiency for the BV8 Reservoir of the Povkh Oil Field (Oct. 2014). Vol. All Days. SPE Russian Petroleum Technology Conference. SPE-171332-MS. DOI: 10.2118/171332-MS. eprint: https://onepetro.org/SPERPTC/proceedings-pdf/14ROGC/All-14ROGC/SPE-171332-MS/1523011/spe-171332-ms.pdf. URL: https://doi.org/10.2118/171332-MS.

Ameli, Forough and Kaveh Mohammadi (2018). "A novel optimization technique for Fast-SAGD process in a heterogeneous reservoir using discrete variables and repetition inhibitory algorithm." In: *Journal of Petroleum Science and Engineering* 171, pp. 982–992. ISSN: 0920-4105. DOI: https://doi.org/10.1016/j.petrol.2018.08.008. URL: https://www.sciencedirect.com/science/article/pii/S0920410518306740.

Anderson, Roger, Boyi Xie, Leon Wu, Arthur Kressner, Joseph Jr, Matthew Ockree, Kenneth Brown, Peter Carragher, and Mark McLane (Jan. 2016). "Using Machine Learning to Identify the Highest Wet Gas Producing Mix of Hydraulic Fracture Classes and Technology Improvements in the Marcellus Shale." In: DOI: 10.15530/urtec-2016-2430481.

Andrade Marin, Antonio, Salim Busaidy, Mohammed Murad, Issa Balushi, Abdullah Riyami, Sultan Jahwari, Adnan Ghadani, Erik Ferdiansyah, Ghosin Shukaili, Faham Amri, Nitish Kumar, Eduardo Marin, Rahul Gala, Rakesh Rai, Bimal Venkatesh, Brian Bai, Arun Kumar, Edward Ang, and George Jacob, eds. (2019). *ESP Well and Component Failure Prediction in Ad-*

vance using Engineered Analytics - A Breakthrough in Minimizing Unscheduled Subsurface Deferments. Vol. Day 2 Tue, November 12, 2019. Abu Dhabi International Petroleum Exhibition and Conference. DOI: 10.2118/197806-MS.

Andrianov, Nikolai (2018). "A Machine Learning Approach for Virtual Flow Metering and Forecasting." In: *IFAC-PapersOnLine* 51.8. 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production OOGP 2018, pp. 191–196. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.06.376. URL: https://www.sciencedirect.com/science/article/pii/S2405896318307067.

ArnØ, Mikkel, John-Morten Godhavn, and Ole Morten Aamo (Nov. 2, 2020). "Deep Reinforcement Learning Applied to Managed Pressure Drilling." In: *Day 2 Tue, November 03, 2020*. SPE Norway Subsurface Conference. Virtual: SPE, D021S007R001. DOI: 10.2118/200757-MS. URL: https://onepetro.org/SPEBERG/proceedings/20BERG/2-20BERG/Virtual/448669 (visited on 05/27/2022).

Arteaga-Arteaga, Harold Brayan, Alejandro Mora-Rubio, Frank Florez, Nicolas Murcia-Orjuela, Cristhian Eduardo Diaz-Ortega, Simon Orozco-Arias, Melissa delaPava, Mario Alejandro Bravo-OrtÃz, Melvin Robinson, Pablo Guillen-Rondon, and Reinel Tabares-Soto (Nov. 29, 2021). "Machine learning applications to predict two-phase flow patterns." In: *PeerJ Computer Science* 7, e798. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.798. URL: https://peerj.com/articles/cs-798 (visited on 10/30/2022).

Audet, C. and J.E. Dennis Jr (2004). "Generalized pattern search for global optimization." In: *Journal of Optimization Theory and Applications* 123.1, pp. 168–191.

– (2006). "Mesh adaptive direct search algorithms for constrained optimization." In: *SIAM Journal on Optimization* 17.1, pp. 188–217.

Avci, Onur, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, and Daniel J Inman (2018). "Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks." In: *Journal of Sound and Vibration* 424, pp. 158–172.

Avcı, Onur, Osama Abdeljaber, Mustafa Serkan Kiranyaz, Boualem Boashash, Henry A. Sodano, and Daniel J. Inman (2018). "Efficiency Validation of One Dimensional Convolutional Neural Networks for Structural Damage Detection Using A SHM Benchmark Data." In.

Avcı, Onur, Osama Abdeljaber, Serkan Kiranyaz, and Daniel J. Inman (2017). "Structural Damage Detection in Real Time: Implementation of 1D Convolutional Neural Networks for SHM Applications." In.

Awaid, A., H. Al-Muqbali, A. Al-Bimani, Z. Yazeedi, H. Al-Sukaity, K. Al-Harthy, and Alastair Baillie (Jan. 19, 2014). "ESP Well Surveillance using Pattern Recognition Analysis, Oil Wells, Petroleum Development Oman." In: International Petroleum Technology Conference, IPTC–17413–MS. DOI:

10.2523/IPTC-17413-MS. URL: https://doi.org/10.2523/IPTC-17413-MS (visited on 05/25/2022).

Awange, Joseph L., Béla Paláncz, Robert H. Lewis, and Lajos Völgyesi (2018). "Symbolic Regression." In: *Mathematical Geosciences: Hybrid Symbolic-Numeric Methods*. Cham: Springer International Publishing, pp. 321–357. ISBN: 978-3-319-67371-4. DOI: 10.1007/978-3-319-67371-4_11. URL: https://doi.org/10.1007/978-3-319-67371-4_11.

Aziz, K., A.B. Ramesh, and P.T. Woo (Dec. 1, 1987). "Fourth SPE Comparative Solution Project: Comparison of Steam Injection Simulators." In: *Journal of Petroleum Technology* 39.12, pp. 1576–1584. ISSN: 0149-2136. DOI: 10.2118/13510-PA. URL: https://doi.org/10.2118/13510-PA (visited on 05/28/2022).

Barrios, Lissett and Vinay Pydah (Oct. 4, 2021). "Perdido GoM Total Field ESP Integrated Full-Field Modeling for Surveillance and Forecast Prediction." In: SPE Gulf Coast Section Electric Submersible Pumps Symposium, D041S011R001. DOI: 10.2118/204498-MS. URL: https://doi.org/10.2118/204498-MS (visited on 04/01/2022).

Bermudez, Fernando, Noor Al Nahhas, Hafsa Yazdani, Michael LeTan, and Mohammed Shono (Nov. 15, 2021). "Unlocking the Potential of Electrical Submersible Pumps: The Successful Testing and Deployment of a Real-Time Artificially Intelligent System, for Failure Prediction, Run Life Extension, and Production Optimization." In: D042S278R001. DOI: 10.2118/207839-MS. URL: https://doi.org/10.2118/207839-MS (visited on 04/09/2022).

Bhardwaj, Abhijeet Sandeep, Rahul Saraf, Geetha Gopakumar Nair, and Sridharan Vallabhaneni, eds. (2019). *Real-Time Monitoring and Predictive Failure Identification for Electrical Submersible Pumps*. Vol. Day 2 Tue, November 12, 2019. Abu Dhabi International Petroleum Exhibition and Conference. DOI: 10.2118/197911-MS.

Bikmukhametov, Timur and Johannes JÃ¤schke (Jan. 2020). "First Principles and Machine Learning Virtual Flow Metering: A Literature Review." In: *Journal of Petroleum Science and Engineering* 184, p. 106487. ISSN: 09204105. DOI: 10.1016/j.petrol.2019.106487. URL: https://linkinghub.elsevier.com/retrieve/pii/S0920410519309088 (visited on 11/20/2022).

Bilgin, Enes (Dec. 18, 2020). *Mastering Reinforcement Learning with Python: Build next-generation, self-learning models using reinforcement learning techniques and best practices*. S.l.: Packt Publishing. 544 pp. ISBN: 978-1-83864-414-7.

Bilogan, Andriy, Vitaly Elichev, Huzaifah Binn-Tahir, Rinat Khabibullin, and Konstantin Litvinenko, eds. (2019). *A Real-time Well Integrity Monitoring Workflow Based on Artificial Intelligence and Dynamic Flow Modeling*. Vol. All Days. International Conference on Multiphase Production Technology.

Binder, Benjamin J.T., Alexey Pavlov, and Tor A. Johansen (2015). "Estimation of Flow Rate and Viscosity in a Well with an Electric Submersible Pump using Moving Horizon Estimationâˆ—âˆ—This work is funded by the Research Council of Norway and Statoil through the PETROMAKS project No. 215684: Enabling High-Performance Safety-Critical Offshore and Subsea Automatic Control Systems Using Embedded Optimization (emOpt)." In: *IFAC-PapersOnLine* 48.6. 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production OOGP 2015, pp. 140–146. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.08.022. URL: https://www.sciencedirect.com/science/article/pii/S2405896315008897.

Borden, Zachary H., Amr El-Bakry, and Peng Xu, eds. (2016). *Workflow Automation for Gas Lift Surveillance and Optimization, Gulf of Mexico*. Vol. All Days. SPE Intelligent Energy International Conference and Exhibition. DOI: 10.2118/181094-MS.

Brunton, Steven L. and J. Nathan Kutz (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press. DOI: 10.1017/9781108380690.

Camilleri, L., M. El Gindy, and A. Rusakov, eds. (2016). *ESP Real-Time Data Enables Well Testing with High Frequency, High Resolution, and High Repeatability in an Unconventional Well*. Vol. All Days. SPE/AAPG/SEG Unconventional Resources Technology Conference. DOI: 10.15530/URTEC-2016-2471526.

Camilleri, L.., M.. El Gindy, A.. Rusakov, and S.. Adoghe, eds. (2015). *Converting ESP Real-Time Data to Flow Rate and Reservoir Information for a Remote Oil Well*. Vol. Day 2 Wed, September 16, 2015. SPE Middle East Intelligent Oil and Gas Symposium. DOI: 10.2118/176780-MS.

Camilleri, L.., M.. El Gindy, A.. Rusakov, I.. Ginawi, H.. Abdelmotaal, E.. Sayed, T.. Edris, and M.. Karam, eds. (2017). *Increasing Production With High-Frequency and High-Resolution Flow Rate Measurements from ESPs*. Vol. Day 2 Tue, April 25, 2017. SPE Gulf Coast Section Electric Submersible Pumps Symposium. DOI: 10.2118/185144-MS.

Camilleri, L.., M.. El-Gindy, A.. Rusakov, F.. Bosia, P.. Salvatore, and G.. Rizza, eds. (2016). *Testing the Untestable... Delivering Flowrate Measurements with High Accuracy on a Remote ESP Well*. Vol. Day 2 Tue, November 08, 2016. Abu Dhabi International Petroleum Exhibition and Conference. DOI: 10.2118/183337-MS.

Camilleri, Lawrence and Wentao Zhou, eds. (2011). *Obtaining Real-Time Flow Rate, Water Cut, and Reservoir Diagnostics from ESP Gauge Data*. Vol. All Days. SPE Offshore Europe Conference and Exhibition. DOI: 10.2118/145542-MS.

Carpenter, Chris (2019). "Analytics Solution Helps Identify Rod-Pump Failure at the Wellhead." In: *Journal of Petroleum Technology* 71.05, pp. 63–64. ISSN: 0149-2136. DOI: 10.2118/0519-0063-JPT.

Cornelio, J., S. Mohd Razak, A. Jahandideh, Y. Cho, H-H. Liu, R. Vaidya, and B. Jafarpour, eds. (2021). *Physics-Assisted Transfer Learning for Production Prediction in Unconventional Reservoirs*. Vol. Day 3 Wed, July 28, 2021. SPE/AAPG/SEG Unconventional Resources Technology Conference. DOI: 10.15530/urtec-2021-5688.

Cramer, Ron and Keat-Choon Goh, eds. (2009). *Data Driven Surveillance and Optimization for Gas, Subsea and Multizone Wells*. Vol. All Days. SPE Digital Energy Conference and Exhibition. DOI: 10.2118/122554-MS.

Cui, Yao, Minze Zhang, and Qi Tang (Oct. 2020). "SYMBOLIC REGRESSION FOR FORMULATION OF SHEAR RESISTANCE OF BEARING-TYPE BOLTED CONNECTIONS." In.

Datta-Gupta, A., M.J. King, and P.B. Bhattad (2010). "Equalizing water breakthrough of producing wells based on time of flight concept in streamline simulators." In: *SPE Journal* 15.4, pp. 1028–1038.

Dawar, K. (2021). *Reinforcement Learning for Well Location Optimization*. Pennsylvania State University. URL: https://books.google.de/books?id=0gSUzgEACAAJ.

De Paola, Giorgio, Cristina Ibanez-Llano, Jesus Rios, and Georgios Kollias (Oct. 26, 2020). "Reinforcement Learning For Field Development Policy Optimization." In: SPE Annual Technical Conference and Exhibition, D041S046R003. DOI: 10.2118/201254-MS. URL: https://doi.org/10.2118/201254-MS (visited on 04/18/2022).

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering." In: *NIPS*.

Dehdari, Vahid, Akhil Datta-Gupta, and Albert C Reynolds (2011). "Ensemble-based optimization of oil reservoir production under uncertainty." In: *Computers & Geosciences* 37, pp. 1558–1569.

Dethlefs, Jerald C. and Bettina Chastain (2012). "Assessing Well-Integrity Risk: A Qualitative Model." In: *Spe Drilling & Completion* 27, pp. 294–302.

Eaton, Ammon N., Logan D. R. Beal, Samuel D. Thorpe, Casey B. Hubbell, John D. Hedengren, Roar Nybø, and Manuel Aghito (Feb. 2, 2017). "Real time model identification using multi-fidelity models in managed pressure drilling." In: *Computers & Chemical Engineering* 97, pp. 76–84. ISSN: 0098-1354. DOI: 10.1016/j.compchemeng.2016.11.008. URL: https://www.sciencedirect.com/science/article/pii/S0098135416303428 (visited on 07/23/2022).

Elichev, Vitaly, Andriy Bilogan, Konstantin Litvinenko, Rinat Khabibullin, Alexey Alferov, and Alexey Vodopyan, eds. (2019). *Understanding Well Events with Machine Learning*. Vol. Day 1 Tue, October 22, 2019. SPE Russian Petroleum Technology Conference. DOI: 10.2118/196861-MS.

Eren, Levent (2017). "Bearing Fault Detection by One-Dimensional Convolutional Neural Networks." In: *Mathematical Problems in Engineering* 2017, pp. 1–9.

Eren, Levent, Turker Ince, and Serkan Kiranyaz (2019). "A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier." In: *Journal of Signal Processing Systems* 91, pp. 179–189.

Ertekin, Turgay and Qian Sun (2019). "Artificial Intelligence Applications in Reservoir Engineering: A Status Check." In: *Energies* 12.15. DOI: 10.3390/en12152897. URL: https://www.mdpi.com/1996-1073/12/15/2897.

Robust Waterflooding Optimization of Multiple Geological Scenarios (Sept. 2006). Vol. All Days. SPE Annual Technical Conference and Exhibition. SPE-102913-MS. DOI: 10.2118/102913-MS. eprint: https://onepetro.org/SPEATCE/proceedings-pdf/06ATCE/All-06ATCE/SPE-102913-MS/2816523/spe-102913-ms.pdf. URL: https://doi.org/10.2118/102913-MS.

Falcone, G., G. F. Hewitt, C. Alimonti, and B. Harrison (Sept. 2001). "Multiphase Flow Metering: Current Trends and Future Developments." In: SPE Annual Technical Conference and Exhibition All Days. SPE-71474-MS. DOI: 10.2118/71474-MS. URL: https://doi.org/10.2118/71474-MS.

Falcone, Gioia (2009). "Chapter 3 Multiphase Flow Metering Principles." In: *Multiphase Flow Metering*. Ed. by Gioia Falcone, G.F. Hewitt, and Claudio Alimonti. Vol. 54. Developments in Petroleum Science. Elsevier, pp. 33–45. DOI: https://doi.org/10.1016/S0376-7361(09)05403-X. URL: https://www.sciencedirect.com/science/article/pii/S037673610905403X.

Farahi, M.M. Moshir, M. Ahmadi, and B. Dabir (2021). "Model-based waterflooding optimization using multi-objective approach for efficient reservoir management." In: *Journal of Petroleum Science and Engineering* 196, p. 107988. ISSN: 0920-4105. DOI: https://doi.org/10.1016/j.petrol.2020.107988. URL: https://www.sciencedirect.com/science/article/pii/S0920410520310433.

Farahi, Mohammad, Mohammad Ahmadi, and Bahram Dabir (Jan. 2021). "Model-based production optimization under geological and economic uncertainties using multi-objective particle swarm method." In: *Oil & Gas Science and Technology* 76, p. 60. DOI: 10.2516/ogst/2021039.

Foss, Bjarne A., Bjarne Grimstad, and Vidar Gunnerud (2015). "Production Optimization - Facilitated by Divide and Conquer Strategies." In: *IFAC-PapersOnLine* 48, pp. 1–8.

Gao, Qian, Shaobo Sun, and Jianchao Liu (2015). "Working Condition Detection of Suck Rod Pumping System via Extreme Learning Machine." In: *Proceedings of the 2nd International Conference on Civil, Materials and Environmental Sciences*. Advances in Engineering Research. Atlantis PressParis, France. DOI: 10.2991/cmes-15.2015.120.

Garcia, Carlos A., Akhan Mukhanov, and Henry Torres, eds. (2019). *Chan Plot Signature Identification as a Practical Machine Learning Classification Problem*. Vol. Day 3 Thu, March 28, 2019. IPTC International Petroleum Technology Conference. DOI: 10.2523/IPTC-19143-MS.

Ghadami Jadval Ghadam, Aboutaleb (May 2015). "Prediction of Gas Critical Flow Rate for Continuous Lifting of Liquids from Gas Wells Using Comparative Neural Fuzzy Inference System." In: *Journal of Applied Environmental and Biological Sciences* 5, pp. 196–202.

Gryzlov, Anton, Sergey Safonov, Muqbil Alkhalaf, and Muhammad Arsalan, eds. (2020). *Novel Methods for Production Data Forecast Utilizing Machine Learning and Dynamic Mode Decomposition*. Vol. Day 1 Mon, November 09, 2020. Abu Dhabi International Petroleum Exhibition and Conference. DOI: 10.2118/202792-MS.

Guevara, J.L., Rajan Patel, and Japan Trivedi (Nov. 1, 2021). "Optimization of steam injection in SAGD using reinforcement learning." In: *Journal of Petroleum Science and Engineering* 206, p. 108735. ISSN: 0920-4105. DOI: 10.1016/j.petrol.2021.108735. URL: https://www.sciencedirect.com/science/article/pii/S0920410521003958.

Guo, Dong, Cauligi S. Raghavendra, Ke-Thia Yao, Mark Harding, Amir Anvar, and Anil Patel, eds. (2015). *Data Driven Approach to Failure Prediction for Electrical Submersible Pump Systems*. SPE Western Regional Meeting. DOI: 10.2118/174062-MS.

Guo, Yanbao, Min Zhang, Hui Yang, Deguo Wang, Melvin A. Ramos, Travis Shihao Hu, and Quan Xu (2022). "Friction Challenge in Hydraulic Fracturing." In: *Lubricants* 10.2. ISSN: 2075-4442. DOI: 10.3390/lubricants10020014. URL: https://www.mdpi.com/2075-4442/10/2/14.

Gupta, Supriya, Michael Nikolaou, Luigi Saputelli, and Cesar Bravo, eds. (2016). *ESP Health Monitoring KPI: A Real-Time Predictive Analytics Application*. SPE Intelligent Energy International Conference and Exhibition. DOI: 10.2118/181009-MS.

Hadi, Farqad, Andreas Eckert, and Faleh Almahdawi, eds. (2019). *Real-Time Pore Pressure Prediction in Depleted Reservoirs Using Regression Analysis and Artificial Neural Networks*. Vol. Day 3 Wed, March 20, 2019. SPE Middle East Oil and Gas Show and Conference. DOI: 10.2118/194851-MS.

Haili Dong, Bingsheng Wang, Chang-Tien Lu (n.d.). "Deep Sparse Coding Based Recursive Disaggregation Model for Water Conservation." In: ().

Haouche, Mohamed, Adrien Tessier, Younes Deffous, and Jean-Francois Authier, eds. (2012a). *Virtual Flow Meter pilot: based on Data Validation and Reconciliation Approach*. Vol. All Days. SPE International Production and Operations Conference and Exhibition. DOI: 10.2118/157283-MS.

Haouche, Mohamed, Adrien. Tessier, Younes. Deffous, J-F.. Authier, J-P Couput, R.. Caulier, and B.. Vrielynck (2012b). "Smart Metering: An

Online Application of Data Validation and Reconciliation Approach." In: SPE-149908–MS. DOI: 10.2118/149908-MS.

He, Z. and L.J. Durlofsky (2011). "Reduced order models." In: *Computational Geosciences* 15.2, pp. 221–246.

Hong, Bing-Yuan, Sheng-Nan Liu, Xiao-Ping Li, Di Fan, Shuai-Peng Ji, Si-Hang Chen, Cui-Cui Li, and Jing Gong (2022). "A liquid loading prediction method of gas pipeline based on machine learning." In: *Petroleum Science* 19.6, pp. 3004–3015. ISSN: 1995-8226. DOI: https://doi.org/10.1016/j.petsci.2022.05.002. URL: https://www.sciencedirect.com/science/article/pii/S1995822622001017.

Horowitz, Paul, Lorenz T Biegler, and Jean-Luc Vay (2013). "Simulation-based optimization of oil reservoir production." In: *Journal of Petroleum Science and Engineering* 111, pp. 180–194.

Hou, Jian, Kang Zhou, Xian-Song Zhang, Xiao-Dong Kang, and Hai Xie (2015). "A review of closed-loop reservoir management." In: *Petroleum Science* 12.1, pp. 114–128. ISSN: 1672-5107. DOI: 10.1007/s12182-014-0005-6.

Hourfar, Farzad, Hamed Jalaly Bidgoly, Behzad Moshiri, Karim Salahshoor, and Ali Elkamel (Jan. 1, 2019). "A reinforcement learning approach for waterflooding optimization in petroleum reservoirs." In: *Engineering Applications of Artificial Intelligence* 77, pp. 98–116. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2018.09.019. URL: https://www.sciencedirect.com/science/article/pii/S0952197618302057 (visited on 05/27/2022).

Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017). "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." In: *ArXiv* abs/1704.04861.

Ince, Turker, Serkan Kiranyaz, Levent Eren, Murat Askar, and M. Gabbouj (2016). "Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks." In: *IEEE Transactions on Industrial Electronics* 63, pp. 7067–7075.

Ishak, Mohd Azmin, Tareq Aziz Al-qutami Hasan, Halvard Ellingsen, Torgeir Ruden, and Hatef Khaledi, eds. (2020). *Evaluation of Data Driven Versus Multiphase Transient Flow Simulator for Virtual Flow Meter Application*. Vol. Day 2 Tue, November 03, 2020. Offshore Technology Conference Asia. DOI: 10.4043/30422-MS.

J.L. Guevara, Rajan Patel, and Japan Trivedi (2021). "Optimization of steam injection in SAGD using reinforcement learning." In: *Journal of Petroleum Science and Engineering* 206, p. 108735. ISSN: 0920-4105. DOI: 10.1016/j.petrol.2021.108735. URL: https://www.sciencedirect.com/science/article/pii/S0920410521003958.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013). *An Introduction to Statistical Learning: with Applications in R*. 1st. New York, NY: Springer.

Jansen, J. D., R. M. Fonseca, S. Kahrobaei, M. M. Siraj, G. M. Van Essen, and P. M. J. Van den Hof (2014). "The egg model â€" a geological ensemble for reservoir simulation." In: *Geoscience Data Journal* 1.2, pp. 192–195. DOI: https://doi.org/10.1002/gdj3.21. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/gdj3.21. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/gdj3.21.

Jansen, JD (2011). "Efficient production optimization using adjoint-based gradient computation." In: *SPE Journal* 16.2, pp. 401–412.

Khamehchi, E., S. V. Yasrebi, and A. Ebrahimi (2014). "Prediction of the Influence of Liquid Loading on Wellhead Parameters." In: *Petroleum Science and Technology* 32.14, pp. 1680–1689. DOI: 10.1080/10916466.2011.603003. eprint: https://doi.org/10.1080/10916466.2011.603003. URL: https://doi.org/10.1080/10916466.2011.603003.

Khamehchi, Ehsan, Iman Rahimzadeh Kivi, and Mohammadreza Akbari (2014). "A novel approach to sand production prediction using artificial intelligence." In: *Journal of Petroleum Science and Engineering* 123. Neural network applications to reservoirs: Physics-based models and data models, pp. 147–154. ISSN: 0920-4105. DOI: https://doi.org/10.1016/j.petrol.2014.07.033. URL: https://www.sciencedirect.com/science/article/pii/S0920410514002320.

Khan, Hassan and Clifford Louis, eds. (2021). *An Artificial Intelligence Neural Networks Driven Approach to Forecast Production in Unconventional Reservoirs – Comparative Analysis with Decline Curve*. Vol. Day 10 Thu, April 01, 2021. IPTC International Petroleum Technology Conference. DOI: 10.2523/IPTC-21350-MS.

Kim, Yoon (2014). "Convolutional Neural Networks for Sentence Classification." In: *Conference on Empirical Methods in Natural Language Processing*.

Kiranyaz, Serkan, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman (2021a). "1D convolutional neural networks and applications: A survey." In: *Mechanical Systems and Signal Processing* 151, p. 107398. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2020.107398. URL: https://www.sciencedirect.com/science/article/pii/S0888327020307846.

– (2021b). "1D convolutional neural networks and applications: A survey." In: *Mechanical Systems and Signal Processing* 151, p. 107398. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2020.107398. URL: https://www.sciencedirect.com/science/article/pii/S0888327020307846.

Kiranyaz, Serkan, Adel Gastli, Lazhar Ben-Brahim, Nasser Ahmed Al-Emadi, and M. Gabbouj (2019). "Real-Time Fault Detection and Identification for MMC Using 1-D Convolutional Neural Networks." In: *IEEE Transactions on Industrial Electronics* 66, pp. 8760–8771.

Kiranyaz, Serkan, Turker Ince, and M. Gabbouj (2017). "Personalized Monitoring and Advance Warning System for Cardiac Arrhythmias." In: *Scientific Reports* 7.

– (2016). "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks." In: *IEEE Transactions on Biomedical Engineering* 63, pp. 664–675.

Kiranyaz, Serkan, Turker Ince, Ridha Hamila, and M. Gabbouj (2015). "Convolutional Neural Networks for patient-specific ECG classification." In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2608–2611.

Krikunov, Dmitry, Semen Kosyachenko, Dmitry Lukovkin, Alexander Kunchinin, Roman Tolmachev, and Roman Chebotarev, eds. (2019). *AI-Based ESP Optimal Control Solution to Optimize Oil Flow Across Multiple Wells*. Vol. Day 2 Tue, October 22, 2019. SPE Gas. DOI: 10.2118/198673-MS.

Kubota, Leonardo and Danilo Reinert, eds. (2019). *Machine Learning Forecasts Oil Rate in Mature Onshore Field Jointly Driven by Water and Steam Injection*. Vol. Day 2 Tue, October 01, 2019. SPE Annual Technical Conference and Exhibition. DOI: 10.2118/196152-MS.

Lea, James, Henry Nickens, and Michael Wells (2003). "Chapter 2 - Recognizing Symptoms of Liquid Loading in Gas Wells." In: *Gas Well Deliquification*. Ed. by James Lea, Henry Nickens, and Michael Wells. Burlington: Gulf Professional Publishing, pp. 13–26. ISBN: 978-0-7506-7724-0. DOI: https://doi.org/10.1016/B978-075067724-0/50002-8. URL: https://www.sciencedirect.com/science/article/pii/B9780750677240500028.

Li, X. G. (2010). "Research on the fault diagnosis of the submersible pump units based on analysisof vibration signals." In: *Qingdao: China University of Petroleum (East China)*.

Li, Xiangyu, Xianwen Gao, Yongbin Cui, and Kun Li (2013). "Dynamic liquid level modeling of sucker-rod pumping systems based on Gaussian process regression." In: *2013 Ninth International Conference on Natural Computation (ICNC)*, pp. 917–922. DOI: 10.1109/ICNC.2013.6818107.

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra (2015). *Continuous control with deep reinforcement learning*. DOI: 10.48550/ARXIV.1509.02971. URL: https://arxiv.org/abs/1509.02971.

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2017). "Focal loss for dense object detection." In: *IEEE transactions on pattern analysis and machine intelligence* 42.2, pp. 318–327.

Liu, Hao, Dandan Zhu, Yi Liu, Aimin Du, Dong Chen, and Zhihui Ye (Dec. 14, 2018). "A Reinforcement Learning Based 3D Guided Drilling Method: Beyond Ground Control." In: *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*. ICNCC 2018. New York, NY, USA: Association for Computing Machinery, pp. 44–48.

ISBN: 978-1-4503-6553-6. DOI: 10.1145/3301326.3301374. URL: https://doi.org/10.1145/3301326.3301374 (visited on 05/27/2022).

Liu, S.., C. S. Raghavendra, Y.. Liu, K.. Yao, O.. Balogun, L.. Olabinjo, R.. Soma, J.. Ivanhoe, B.. Smith, B.. Seren, T. L. Lenz, C. G. Dinesh Babu, and I.. Ershaghi, eds. (2011). *Automatic Early Fault Detection for Rod Pump Systems*. Vol. All Days. SPE Annual Technical Conference and Exhibition. DOI: 10.2118/146038-MS.

Lowe, Ryan, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch (2017). *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. DOI: 10.48550/ARXIV.1706.02275. URL: https://arxiv.org/abs/1706.02275.

Ma, Hongze, Gaoming Yu, Yuehui She, and Yongan Gu (Sept. 30, 2019). "Waterflooding Optimization under Geological Uncertainties by Using Deep Reinforcement Learning Algorithms." In: SPE Annual Technical Conference and Exhibition, D031S043R001. DOI: 10.2118/196190-MS. URL: https://doi.org/10.2118/196190-MS (visited on 04/18/2022).

Makhotin, Ivan, Dmitry Koroteev, and Evgeny Burnaev (Mar. 2019). "Gradient boosting to boost the efficiency of hydraulic fracturing." In: *Journal of Petroleum Exploration and Production Technology* 9. DOI: 10.1007/s13202-019-0636-7.

Manikonda, Kaushik, Abu Rashid Hasan, Chinemerem Edmond Obi, Raka Islam, Ahmad Khalaf Sleiti, Motasem Wadi Abdelrazeq, and Mohammad Azizur Rahman (Dec. 9, 2021). "Application of Machine Learning Classification Algorithms for Two-Phase Gas-Liquid Flow Regime Identification." In: *Day 4 Thu, November 18, 2021*. Abu Dhabi International Petroleum Exhibition & Conference. Abu Dhabi, UAE: SPE, D041S121R004. DOI: 10.2118/208214-MS. URL: https://onepetro.org/SPEADIP/proceedings/21ADIP/4-21ADIP/D041S121R004/473840 (visited on 10/31/2022).

Mask, Gene, Xingru Wu, and Kegang Ling (Mar. 22, 2019). "An Improved Model for Gas-Liquid Flow Pattern Prediction Based on Machine Learning." In: *Day 2 Wed, March 27, 2019*. International Petroleum Technology Conference. Beijing, China: IPTC, D021S026R005. DOI: 10.2523/IPTC-19174-MS. URL: https://onepetro.org/IPTCONF/proceedings/19IPTC/2-19IPTC/Beijing,%20China/154169 (visited on 10/31/2022).

Miftakhov, Ruslan, Abdulaziz Al-Qasim, and Igor Efremov (Jan. 13, 2020). "Deep Reinforcement Learning: Reservoir Optimization from Pixels." In: *Day 2 Tue, January 14, 2020*. International Petroleum Technology Conference. Dhahran, Kingdom of Saudi Arabia: IPTC, D021S052R002. DOI: 10.2523/IPTC-20151-MS. URL: https://onepetro.org/IPTCONF/proceedings/20IPTC/2-20IPTC/Dhahran,%20Kingdom%20of%20Saudi%20Arabia/154747 (visited on 05/27/2022).

Frac-Hit Dynamic Modeling using Artificial Intelligence Machine Learning (July 2020). Vol. Day 3 Wed, July 22, 2020. SPE/AAPG/SEG Unconventional

Resources Technology Conference, D033S060R002. DOI: 10.15530/urtec-2020-2647. eprint: https://onepetro.org/URTECONF/proceedings-pdf/20URTC/3-20URTC/D033S060R002/2265163/urtec-2020-2647-ms.pdf. URL: https://doi.org/10.15530/urtec-2020-2647.

Molnar, Gergely (2022). "Economics of Gas Transportation by Pipeline and LNG." In: *The Palgrave Handbook of International Energy Economics*. Ed. by Manfred Hafner and Giacomo Luciani. Cham: Springer International Publishing, pp. 23–57. ISBN: 978-3-030-86884-0. DOI: 10.1007/978-3-030-86884-0_2. URL: https://doi.org/10.1007/978-3-030-86884-0_2.

Morteza Zadkarami, Ali Akbar Safavi, Mohammad Taheri, and Fabienne Fariba Salimi (2020). "Data driven leakage diagnosis for oil pipelines: An integrated approach of factor analysis and deep neural network classifier." In: *Transactions of the Institute of Measurement and Control* 42.14, pp. 2708–2718. DOI: 10.1177/0142331220928145.

Nande, Soumitra (2018). "Application of Machine Learning for Closure Pressure Determination." In: *Day 2 Tue, September 25, 2018*.

Noshi, Christine Ikram, Marco Risk Eissa, and Ramez Maher Abdalla, eds. (2019). *An Intelligent Data Driven Approach for Production Prediction*. Vol. Day 4 Thu, May 09, 2019. OTC Offshore Technology Conference. DOI: 10.4043/29243-MS.

Prediction of Critical Gas Flow Rate for Gas Wells Unloading (Oct. 2002). Vol. All Days. Abu Dhabi International Petroleum Exhibition and Conference. SPE-78568-MS. DOI: 10.2118/78568-MS. eprint: https://onepetro.org/SPEADIP/proceedings-pdf/02ADIPEC/All-02ADIPEC/SPE-78568-MS/2912083/spe-78568-ms.pdf. URL: https://doi.org/10.2118/78568-MS.

Patel, Kalpesh, Elvira Marie B. Aske, and Morten Fredriksen (2014). "Use of Model-Predictive Control for Automating SAGD Well-Pair Operations: A Simulation Study." In: *Spe Production & Operations* 29, pp. 105–113.

Peng, Long, Guoqing Han, Arnold Landjobo Pagou, Liying Zhu, He Ma, Jiayi Wu, and Xiaolong Chai, eds. (2021). *A Predictive Model to Detect the Impending Electric Submersible Pump Trips and Failures*. SPE Annual Technical Conference and Exhibition. DOI: 10.2118/206150-MS.

Peng, Xing-yu, Peng Zhang, and Li-qiong Chen (2009). "Long-Distance Oil/Gas Pipeline Failure Rate Prediction Based on Fuzzy Neural Network Model." In: *2009 WRI World Congress on Computer Science and Information Engineering*. Vol. 5, pp. 651–655. DOI: 10.1109/CSIE.2009.738.

Peng, Yi, Ruidong Zhao, Xishun Zhang, Junfeng Shi, Shiwen Chen, Qingming Gan, Gangyao Li, Xiaoxiong Zhen, and Tao Han, eds. (2019). *Innovative Convolutional Neural Networks Applied in Dynamometer Cards Generation*. Vol. Day 3 Thu, April 25, 2019. SPE Western Regional Meeting. DOI: 10.2118/195264-MS.

Popa, Andrei, Jacob Umbriaco, and Alexia Tirtawidjaja (Apr. 2022). *Effective Neural Networks Models for Inferred Production Prediction in ESP Equipped*

Wells. Vol. Day 3 Thu, April 28, 2022. SPE Western Regional Meeting. DOI: 10.2118/209266-MS. URL: https://doi.org/10.2118/209266-MS.

Popa, Florentina, Serkan Dursun, and Brent Houchens (Sept. 28, 2015). "Fuzzynistic Models for Multiphase Flow Pattern Identification." In: *Day 3 Wed, September 30, 2015*. SPE Annual Technical Conference and Exhibition. Houston, Texas, USA: SPE, D031S039R004. DOI: 10.2118/174812-MS. URL: https://onepetro.org/SPEATCE/proceedings/15ATCE/3-15ATCE/Houston,%5C%20Texas,%5C%20USA/180293.

Purkayastha, Sagar N., Ian D. Gates, and Milana Trifkovic (Jan. 1, 2015). "Model-Predictive-Control (MPC) of Steam Trap Subcool in Steam-Assisted Gravity Drainage (SAGD)." In: *9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015* 48.8, pp. 539–544. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2015.09.023. URL: https://www.sciencedirect.com/science/article/pii/S2405896315011040.

Rammay, Muzammil Hussain and Sami Alnuaim (Dec. 6, 2015). "Flow Regime Prediction Using Fuzzy Logic and Modification in Beggs and Brill Multiphase Correlation." In: *Day 2 Mon, December 07, 2015*. International Petroleum Technology Conference. Doha, Qatar: IPTC, D021S013R003. DOI: 10.2523/IPTC-18267-MS. URL: https://onepetro.org/IPTCONF/proceedings/15IPTC/2-15IPTC/Doha,%20Qatar/153603 (visited on 10/31/2022).

Rao, Subba Rama and Richard Mohan David, eds. (2015). *Integrated Production Testing Framework to Improve Next Generation Production Workflows*. Vol. Day 3 Wed, November 11, 2015. Abu Dhabi International Petroleum Exhibition and Conference. DOI: 10.2118/177940-MS.

Roxas, Ricardo, Matthew Angelo Evangelista, Jalen Aeron Sombillo, Somtochukwu Godfrey Nnabuife, and Karl Ezra Pilario (June 2022). "Machine Learning Based Flow Regime Identification using Ultrasonic Doppler Data and Feature Relevance Determination." In: *Digital Chemical Engineering* 3, p. 100024. ISSN: 27725081. DOI: 10.1016/j.dche.2022.100024. URL: https://linkinghub.elsevier.com/retrieve/pii/S2772508122000151 (visited on 10/30/2022).

Ruiz-Diaz, C M, J A GÃ³mez-Camperos, and M M HernÃ¡ndez-Cely (Jan. 1, 2022). "Flow pattern identification of liquid-liquid (oil and water) in vertical pipelines using machine learning techniques." In: *Journal of Physics: Conference Series* 2163.1, p. 012001. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/2163/1/012001. URL: https://iopscience.iop.org/article/10.1088/1742-6596/2163/1/012001 (visited on 10/31/2022).

Sabaa, Ahmed, Mahmoud Abu El Ela, Ahmed H. El-Banbi, and Mohamed H. M. Sayyouh (Sept. 2022). "Artificial Neural Network Model to Predict Production Rate of Electrical Submersible Pump Wells." In: *SPE Production & Operations*, pp. 1–10. ISSN: 1930-1855. DOI: 10.2118/212284-PA. eprint: https://onepetro.org/PO/article-pdf/doi/10.2118/212284-PA/

2977648/spe-212284-pa.pdf. URL: https://doi.org/10.2118/212284-PA.

Saputelli, L., M. Nikolaou, and M. J. Economides (Dec. 15, 2005). "Self-Learning Reservoir Management." In: *SPE Reservoir Evaluation & Engineering* 8.6, pp. 534–547. ISSN: 1094-6470. DOI: 10.2118/84064-PA. URL: https://doi.org/10.2118/84064-PA (visited on 05/27/2022).

Sengel, Ayhan and Gulcan Turkarslan, eds. (2020). *Assisted History Matching of a Highly Heterogeneous Carbonate Reservoir Using Hydraulic Flow Units and Artificial Neural Networks*. Vol. Day 1 Tue, December 01, 2020. SPE Europec featured at EAGE Conference and Exhibition. DOI: 10.2118/200541-MS.

Shafiei, Ali and Maurice B. Dusseault (2013). "Geomechanics of thermal viscous oil production in sandstones." In: *Journal of Petroleum Science and Engineering* 103, pp. 121–139.

Shami, Tareq M., Ayman A. El-Saleh, Mohammed Alswaitti, Qasem Al-Tashi, Mhd Amen Summakieh, and Seyedali Mirjalili (2022). "Particle Swarm Optimization: A Comprehensive Survey." In: *IEEE Access* 10, pp. 10031–10061. DOI: 10.1109/ACCESS.2022.3142859.

Sherif, Sanusi, Omisore Adenike, Eremiokhale Obehi, Adebowale Funso, and Blankson Eyituoyo, eds. (2019). *Predictive Data Analytics for Effective Electric Submersible Pump Management*. SPE Nigeria Annual International Conference and Exhibition. DOI: 10.2118/198759-MS.

Sibaweihi, N.., R. G. Patel, J. L. Guevara, I. D. Gates, and J. J. Trivedi, eds. (2019). *Real-Time Steam Allocation Workflow Using Machine Learning for Digital Heavy Oil Reservoirs*. Vol. Day 2 Wed, April 24, 2019. SPE Western Regional Meeting. DOI: 10.2118/195312-MS.

Sibaweihi, Najmudeen, Rajan G. Patel, J. L. Guevara, Ian D. Gates, and Japan J. Trivedi (2021). "Real-time steam allocation workflow using machine learning for digital heavy oil reservoirs." In: *Journal of Petroleum Science and Engineering* 199, p. 108168.

Song, Jinze, Yuhao Li, Shuai Liu, Youming Xiong, Weixin Pang, Yufa He, and Yaxi Mu (Sept. 2022). "Comparison of Machine Learning Algorithms for Sand Production Prediction: An Example for a Gas-Hydrate-Bearing Sand Case." In: *Energies* 15.18, pp. 1–32. URL: https://ideas.repec.org/a/gam/jeners/v15y2022i18p6509-d908095.html.

Sun, A. (2020). "Optimal carbon storage reservoir management through deep reinforcement learning." In: DOI: 10.1016/j.apenergy.2020.115660.

Sutton, R. and A. Barto (2005). "Reinforcement Learning: An Introduction." In: *IEEE Transactions on Neural Networks*. DOI: 10.1109/TNN.1998.712192.

Takacs, Gabor (2018). "Chapter 1 - Introduction." In: *Electrical Submersible Pumps Manual (Second Edition)*. Ed. by Gabor Takacs. Second Edition. Gulf Professional Publishing, pp. 1–10. ISBN: 978-0-12-814570-8. DOI: https://doi.org/10.1016/B978-0-12-814570-8.00001-5. URL: https://www.sciencedirect.com/science/article/pii/B9780128145708000015.

Theodoridis, Sergios (2020). "Chapter 18 - Neural Networks and Deep Learning." In: *Machine Learning (Second Edition)*. Ed. by Sergios Theodoridis. Second Edition. Academic Press, pp. 901–1038. ISBN: 978-0-12-818803-3. DOI: https://doi.org/10.1016/B978-0-12-818803-3.00030-1. URL: https://www.sciencedirect.com/science/article/pii/B9780128188033000301.

Thuerey, Nils, Philipp Holl, Maximilian Mueller, Patrick Schnell, Felix Trost, and Kiwon Um (Apr. 25, 2022). *Physics-based Deep Learning*. version: 3. arXiv: 2109.05237[physics]. URL: http://arxiv.org/abs/2109.05237 (visited on 07/24/2022).

Tseng, Shih-Hsien and Khoa-Dang Tran (Jan. 2023). "Predicting maintenance through an attention long short-term memory projected model." In: *Journal of Intelligent Manufacturing*, pp. 1–18. DOI: 10.1007/s10845-023-02077-5.

Turner, R.G., M.G. Hubbard, and A.E. Dukler (Nov. 1969). "Analysis and Prediction of Minimum Flow Rate for the Continuous Removal of Liquids from Gas Wells." In: *Journal of Petroleum Technology* 21.11, pp. 1475–1482. ISSN: 0149-2136. DOI: 10.2118/2198-PA. eprint: https://onepetro.org/JPT/article-pdf/21/11/1475/2221983/spe-2198-pa.pdf. URL: https://doi.org/10.2118/2198-PA.

van Jansen Rensburg, Nico, Lisa Kamin, and Skip Davis, eds. (2019). *Using Machine Learning-Based Predictive Models to Enable Preventative Maintenance and Prevent ESP Downtime*. Abu Dhabi International Petroleum Exhibition and Conference. DOI: 10.2118/197146-MS.

Vembadi, Shiv S., Rajan G. Patel, and Vinay Prasad (2018). "Real-time feedback control of SAGD wells using model predictive control to optimize steam chamber development under uncertainty." In: *Canadian Journal of Chemical Engineering* 96, pp. 1290–1305.

Vieira, Ronald E. and Siamack A. Shirazi (May 2022). *Predictive Model for Particle Transport Velocities in Multiphase Gas-Liquid Flows using Artificial Intelligence*. Vol. All Days. PSIG Annual Meeting.

Vinogradov, Dmitrii and Daniil Vorobev, eds. (2020). *Virtual Flowmetering NovyPort Field Examples*. Vol. Day 3 Wed, October 28, 2020. SPE Russian Petroleum Technology Conference. DOI: 10.2118/201876-MS.

Wang, Cai, Chunming Xiong, Hanjun Zhao, Ruidong Zhao, Junfeng Shi, Jianjun Zhang, Xishun Zhang, Hongxing Huang, Shiwen Chen, Yi Peng, and Yizhen Sun, eds. (2020a). *Well Condition Diagnosis of Sucker-Rod Pumping Wells Based on the Machine Learning of Electrical Power Curves in the Context of IoT*. Vol. Day 2 Tue, November 03, 2020. Offshore Technology Conference Asia. DOI: 10.4043/30326-MS.

– eds. (2020b). *Well Condition Diagnosis of Sucker-Rod Pumping Wells Based on the Machine Learning of Electrical Power Curves in the Context of IoT*. Vol. Day 2 Tue, November 03, 2020. Offshore Technology Conference Asia. DOI: 10.4043/30326-MS.

Wen, Chen, Ibrahim Hoteit, and Louis J Durlofsky (2014). "Maximizing reservoir performance by solving coupled optimization problems with production control." In: *SPE Journal* 19.4, pp. 615–628.

Wu, Xiongjun, Guoxiang Liu, and Vyacheslav Romanov (Aug. 2022). *Machine learning applications for frac-hit identification: A field data use case*. Vol. Day 1 Sun, August 28, 2022. SEG International Exposition and Annual Meeting. DOI: 10.1190/image2022-3742713.1. URL: https://doi.org/10.1190/image2022-3742713.1.

Xi, W. J. (2008). "Research on fault diagnosis of electric submersible pumps based on vibration detection." In: *Dongying: China University of Petroleum (East China)* 5.

Xiaoxiao, Lv and Wang Hanxiang (2020). "A Self-Evolution Fault Diagnosis Method of Sucker Rod Pump Based on Simulating Dynamometer Card." In: *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 1–6. DOI: 10.1109/ICMTMA50254.2020.00009.

Xu, Cui, Zhihao Xu, and Zhifeng Yang (2020). "Reservoir operation optimization for balancing hydropower generation and biodiversity conservation in a downstream wetland." In: *Journal of Cleaner Production* 245, p. 118885. ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2019.118885. URL: https://www.sciencedirect.com/science/article/pii/S0959652619337552.

Yakoot, Mostafa Sa'eed, Adel Mohamed Salem Ragab, and Omar Mahmoud (2021). "Machine Learning Application for Gas Lift Performance and Well Integrity." In: *Day 2 Tue, October 19, 2021*.

Real Time Calculation of Fluid Level Using Dynamometer Card of Sucker Rod Pump Well (Dec. 2014). Vol. All Days. IPTC International Petroleum Technology Conference. IPTC-17773-MS. DOI: 10.2523/IPTC-17773-MS. eprint: https://onepetro.org/IPTCONF/proceedings-pdf/14IPTC/All-14IPTC/IPTC-17773-MS/1484489/iptc-17773-ms.pd. URL: https://doi.org/10.2523/IPTC-17773-MS.

Innovative Deep Autoencoder and Machine Learning Algorithms Applied in Production Metering for Sucker-Rod Pumping Wells (July 2019). Vol. Day 1 Mon, July 22, 2019. SPE/AAPG/SEG Unconventional Resources Technology Conference. D013S011R004. DOI: 10.15530/urtec-2019-1090. eprint: https://onepetro.org/URTECONF/proceedings-pdf/19URTC/1-19URTC/D013S011R004/1138218/urtec-2019-1090-ms.pdf. URL: https://doi.org/10.15530/urtec-2019-1090.

Yu, Yingwei, Wei Chen, Qiuhua Liu, Minh Chau, Velizar Vesselinov, and Richard Meehan (Mar. 8, 2021). "Training an Automated Directional Drilling Agent with Deep Reinforcement Learning in a Simulated Environment." In: *Day 4 Thu, March 11, 2021*. SPE/IADC International Drilling Conference and Exhibition. Virtual: SPE, D041S013R002. DOI: 10.2118/

204105-MS. URL: https://onepetro.org/SPEDC/proceedings/21DC/4-21DC/Virtual/460374 (visited on 05/27/2022).

Yuan, Ye, Kang Li, Liang Li, Xiaoxia Yin, and Xiao Liu (2020). "A general end-to-end diagnosis framework for manufacturing systems." In: *National Science Review* 7.2, pp. 418–429. DOI: 10.1093/nsr/nwz190.

Zhang, Jiaming and Zhangxin Chen (Jan. 1, 2018). "Chapter Nine - Formation Damage by Thermal Methods Applied to Heavy Oil Reservoirs." In: *Formation Damage During Improved Oil Recovery*. Ed. by Bin Yuan and David A. Wood. Gulf Professional Publishing, pp. 361–384. ISBN: 978-0-12-813782-6. DOI: 10.1016/B978-0-12-813782-6.00009-9. URL: https://www.sciencedirect.com/science/article/pii/B9780128137826000099 (visited on 06/17/2022).

Zhang, Shirong and Yuling Tang (2008). "Indirect measurement of dynamometer card of pumping unit." In: *2008 7th World Congress on Intelligent Control and Automation*, pp. 4952–4955. DOI: 10.1109/WCICA.2008.4593728.

Zhao, P. (2011). "Study on the vibration fault diagnosis method of centrifugal pump and system implementation." In: *Beijing: North China Electric Power University*, pp. 56–58.

Zhao, Yulong, Albert C Reynolds, and Akhil Datta-Gupta (2011). "Optimization based on quadratic interpolation models for production problems under geological uncertainty." In: *SPE Journal* 16.1, pp. 136–149.

Zhao, Zuoan and Dali Wang, eds. (2021). *Machine Learning for Productivity Prediction in Heterogeneous Carbonate Gas Reservoirs, Central Sichuan Basin, China*. Vol. Day 4 Thu, May 20, 2021. SPWLA Annual Logging Symposium. DOI: 10.30632/SPWLA-2021-0090.

Zhou, Zhenning, Peiyuan Ni, Xiaoxiao Zhu, and Qixin Cao (2021). "Compliant Robotic Assembly based on Deep Reinforcement Learning." In: *2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, pp. 6–9. DOI: 10.1109/MLISE54096.2021.00009.

Zhu, Dandan, Xiaoting Luo, Zhanmin Zhang, Xiangyi Li, Gao Peng, Liping Zhu, and Xuefeng Jin (2021). "Full Reproduction of Surface Dynamometer Card Based on Periodic Electric Current Data." In: *SPE Production & Operations* 36.03, pp. 594–603. ISSN: 1930-1855. DOI: 10.2118/205396-PA.

Zhu, David, Sergey Alyamkin, Laura Sesack, Jeffrey Bridges, and Jamie Letzig, eds. (2016). *Electrical Submersible Pump Operation Optimization with Time Series Production Data Analysis*. Vol. All Days. SPE Intelligent Energy International Conference and Exhibition. DOI: 10.2118/181030-MS.

Zhu, Kai, Liang Wang, Yonghui Du, Cong Jiang, and Zhongwei Sun (2020). "DeepLog: Identify Tight Gas Reservoir Using Multi-Log Signals by a Fully Convolutional Network." In: *IEEE Geoscience and Remote Sensing Letters, title=Prediction of Subsurface NMR T2 Distributions in a Shale Petroleum System Using Variational Autoencoder-Based Neural Networks* 17.4, pp. 568–571. ISSN: 1558-0571. DOI: 10.1109/LGRS.2019.2930587.