Machine Learning and Internet of Things Enable Steam Flood Optimization for Improved Oil Production

Mi Yan
ExxonMobil Global Service Company
mi.yan1@exxonmobil.com

Jonathan C. MacDonald
Imperial Oil Resources Ltd.
jonathan.c.macdonald@esso.ca

Chris T. Reaume
ExxonMobil Global Service Company
chris.t.reaume@esso.ca

Wesley Cobb
ExxonMobil Global Service Company
wesley.cobb@exxonmobil.com

Tamas Toth
ExxonMobil Global Service Company
tamas.toth@exxonmobil.com

Sarah S. Karthigan
ExxonMobil Global Service Company
sarah.karthigan@exxonmobil.com

ABSTRACT

Recent developed machine learning techniques, in association with the Internet of Things (IoT) make possible a method of increasing oil production from existing heavy-oil wells. Steam flood injection, a widely used enhanced oil recovery technique, is use of thermal and gravitational potential to mobilize and dilute heavy oil *in situ* to increase oil production. In contrast to traditional steam flood simulations based on classic physics principles, we present a novel approach using cutting-edge machine learning techniques that have the potential to provide a better way to solve this problem. We propose a workflow for addressing a category of time-series data that can be analyzed with supervised machine learning algorithms and IoT. We demonstrate the effectiveness of the technique in the oil production forecast in steam flood scenarios. Moreover, we build an optimization system that allows one to recommend an optimal steam-allocation plan, and demonstrate a potential 3% improvement in oil production. We develop a minimum viable is pin a cloud platform that can achieve the real-time data collection, transfer, and storage, as well as the cloud-based machine learning model training and implementation. This workflow also offers an applicable solution to other problems with similar time-series data structures, like predictive maintenance.

CCS CONCEPTS

• **Applied computing → Physical sciences and engineering → Engineering**; • **Computing methodologies → Machine learning → Learning paradigms → Supervised learning → Supervised learning by regression**

KEYWORDS

Oil and gas industry, Internet of Things (IoT), XGBoost, Optimization, Time-series, Forecasting, Steam flood

ACM Reference format:

Mi Yan, Jonathan MacDonald, Chris Reaume, Wesley Cobb, Tamas Toth, and Sarah Karthigan. 2019. Machine Learning and Internet of Things Enable Steam Flood Optimization for Improved Oil Production. In *KDD '19: The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 4-8, 2019, Anchorage, Alaska, USA, 5 pages.* https://doi.org/xxxxx/yyyyy

1 INTRODUCTION

1.1 Background

Machine learning [1, 2] and the Internet of Things (IoT) [3-8] have proven successful across various industries. Recently they have gained more attention in the oil and gas industry. Machine learning offers an alternative solution to quite a few long-lasting questions in the oil and gas industry [9-15]. Bergen *et al.* [9] provided a perspective of applications of machine learning in geoscience, and summarized machine learning works finished by the geoscience community. Xu *et al.* [15] reviewed recent progresses in petrophysics with the aid of machine learning. On the other hand, the development of IoT helps achieve the real-time data acquisition via embedded sensors, as well as model building and its deployment at IoT edge devices or a cloud platform. It has wide applications in the oil and gas industry, particularly in the upstream industry, the oil exploration and production sector. Khan *et al.* [16] proposed a reliable and efficient IoT-based architecture for the oilfield environment. Aalsalem *et al.* [17] presented a review of recent advances and open challenges of wireless sensor networks in the oil and gas industry. Combining capabilities of machine learning and IoT, we propose an effective method in the oil production forecast in steam flood scenarios, as well as a steam-allocation optimization system predicting a potential 3% uplift in oil production.

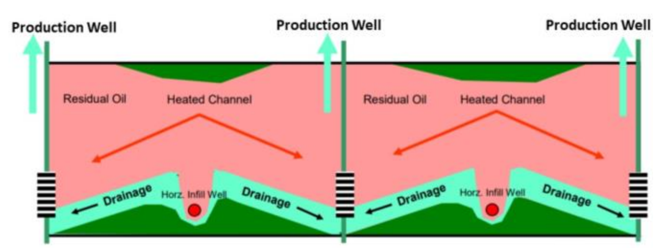

Figure 1: Diagram of steam flood process.

How to increase oil production to meet growing global energy demands is a hot topic. Enhanced oil recovery (EOR) [18] is a widely used technique, which helps increase the oil production in the post-natural-extraction process. Usually the natural pumping stage results in as much as 70% residual crude oil due to low well pressure. In order to efficiently recover oil, three primary EOR methods, thermal injection [19], gas injection, and chemical injection are carried out depending on specific field conditions. Steam flood injection [20-22] is a major thermal injection technique in which steam is injected into infill wells to mobilize and dilute heavy oil using thermal and gravitational potential, so that production wells can easily extract oil from reservoirs (Fig. 1).

1.2 Related work

The oil production forecast in steam flood fields has been studied for decades. Traditional analytical models [21, 22] were built using physics principles and reservoir conditions to describe the performance of steam flood and predict oil production with given steam-allocation plans (Fig. 2). However, there were considerable discrepancies between real productions and predictions. There has been very few machine-learning-based studies of steam flood injection reported. Hama *et al.* [23] employed hierarchical clustering, an unsupervised machine learning algorithm, to create a new steam flood screening criteria and help choose which method of EOR to use for different reservoir conditions. Nevertheless, it did not cover the oil production forecast.

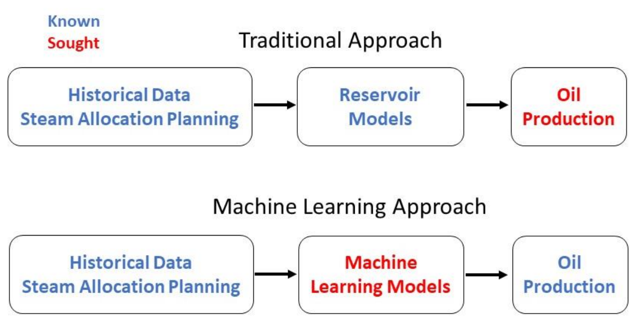

Figure 2: Comparison of traditional and machine learning approaches in the oil production forecast.

1.3 Why can machine learning and IoT help?

In a traditional way, an output, i.e. the oil production here, is predicted by known models and input data. Alternatively, machine learning methods offer a different and potentially better way to solve this problem. In the machine learning approach, input and output in a training dataset are used to train candidate models (Fig. 2). The optimal model is determined according to predefined criteria. This model is evaluated on a test dataset that has never been seen before. In other words, machine learning methods are data driven, whereas traditional approaches are physics principle driven. In contrast to traditional techniques, geological parameters are not necessary for building the machine learning models. Moreover, some decision-tree-based machine learning algorithms, e.g. Random Forests [24] and XGBoost [25], are able to assist in recognizing relative importance of each factor in a complex non-linear system. With these powerful capabilities, machine learning can help a steam-flood surveillance team quantitatively explore potential opportunities to improve oil production.

Machine learning and IoT are complementary to each other. Machine learning is able to address a tremendous amount of data collected by IoT edge devices to reveal hidden patterns that could not be identified before, while IoT makes possible the real-time data acquisition and storage, as well as the machine learning model training and implementation in a cloud platform or IoT edge devices. This work benefits from the development of both machine learning and IoT.

2 DATA & METHODS

2.1 Data collection and introduction

The raw data is collected by edge sensors in fields and is located in five different data sources. The sources have different schema designs, primary keys and sampling frequency. Daily Extract, Transform and Load (ETL) jobs collect and cross reference data, preform error correction and consolidate it to one location. It is then transferred to a cloud storage to be ready for the data engineering process.

Table 1 displays the structure of a daily dataset at the well level in a pad composed of a group of adjacent dependent wells. Our goal is to build one model for one pad to forecast daily oil production of each production well. There are 16 features in the dataset. *Well Name* has two categories: *infill wells* and *production wells*. *Sensor Data* is the real-time measurements of temperature and pressure in fields. *Steam Volume* is the daily steam volume injected into each infill well, which is only valid for *infill wells*, whereas *Well Status*, *Sensor Data* and *Oil Volumes* are only meaningful for *production wells*. Here *Oil Volume* is the daily oil production of each production well, which is the output variable. As expected, missing data is inevitable in the real-world dataset. Before feeding data into machine learning algorithms, the data engineering stage is necessary.

Date	***Well Name***	***Well Status***	***Sensor Data***	***Steam Volume***	***Oil Volume***
4/17/2019	Prod Well 1	Pump	100	NA	23
4/18/2019	Prod Well 1	Shut-In	200	NA	31
…	…	…	…	…	…
4/17/2019	Infill Well 1	NA	NA	6	NA
4/18/2019	Infill Well 1	NA	NA	9	NA
…	…	…	…	…	…

Table 1: Structure of a daily dataset at the well level.

2.2 Workflow

As illustrated in Figure 3, there are five sections in the workflow: data collection and transfer, data engineering, model building, data visualization and optimization system. We perform the missing data imputation using forward copy or backward copy for different features. No individual records are dropped, since a well's complete history is needed for the data engineering stage later. As introduced before, there are two categories of wells so the dataset is separated into two subsets accordingly. Starting from the infill-well subset, we create a new data structure where each row corresponds to records on one day, and columns/features are the daily steam volume injected into each infill well. In the production-well subset, two groups of new features are added. One group are features like *gas_day_rate* when taking into account the effective working time of pumps. The other group are one-hot encodings of the production wells derived from categorical features, e.g. *Well Name*, *Well Status*, and so on. We merge the reorganized infill-well subset to the new production-well subset aligned by *date* to build a new dataset.

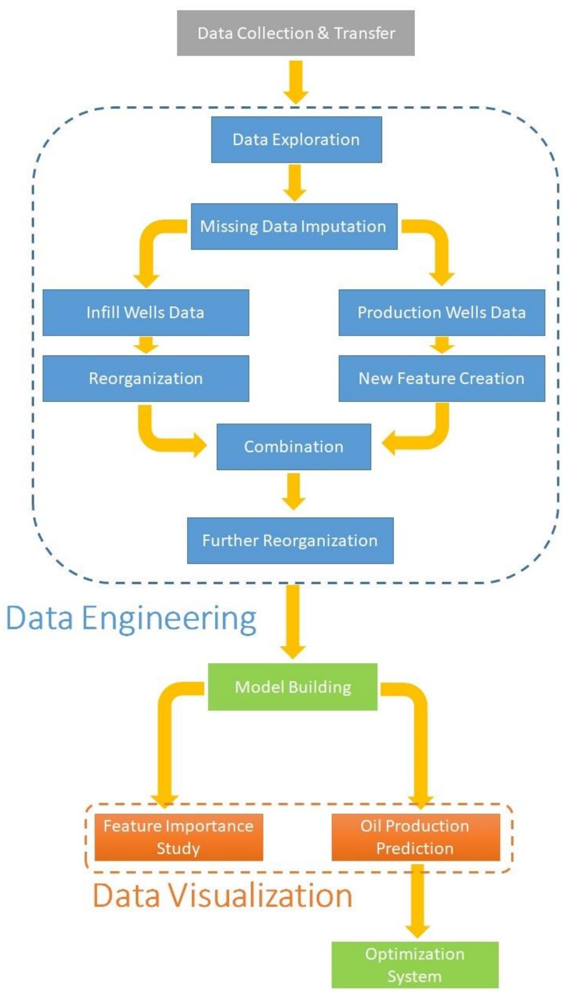

Figure 3: Schematic of workflow of steam flood optimization.

There are two questions to answer: (1) what will be the future oil production with a given steam-allocation plan; (2) how does the historical data influence future production. For a t-day-prior daily production prediction with k-day historical data as the input, we create two groups of new features to account for these two questions, respectively:

prior_m_day_infill_well_x_steam,
where $m = 1, 2, \ldots, t - 1, t$, with infill well names x,
&
prior_n_day_sensor_y_value,
where $n = t + 1, \ldots, t + k$, with sensor names y.

The difference of ranges of m and n is because m focuses on impacts of future steam plans, while n indicates those of historical records. Please note that k is a hyperparameter depending on data. Up to now, the input dataset is ready for building daily production forecast models at the well level, with a typical dimension of 100,000 rows by 1000 columns/features.

All models are trained with GPU-supported XGBoost algorithm [25]. The optimal model is finalized by the hyperparameter grid search with the k-fold time series cross validation.

3 RESULTS

3.1 Importance Study

Here we display results from an optimal 30-day-prior prediction model. Figure 4 lists the eight most important features given by the model. A surveillance team can have a qualitative impression of how important each feature is, which could help them in planning and decision making.

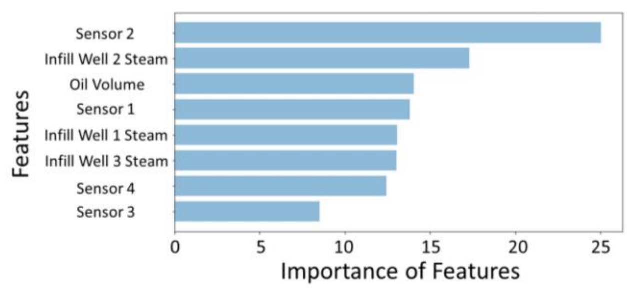

Figure 4: Importance study results (Top 8) given by the optimal model.

3.2 Oil Production Forecast

The dataset is split into training (80%) and test (20%) datasets. Figure 5 compares the normalized real monthly oil production with the predicted production. A ±10% relative error band with respect to the real production is displayed for reference. The real daily production and prediction are both at the well level, which are accumulated by *month* over all production wells in one pad for visualization. The optimal model is selected from the 5-fold time-series cross validation in terms of the metric of *root mean square* *error* (RMSE). The predictions on the test dataset are all within the ±10% relative error range of the real productions, which is a tremendous improvement compared to previous works [21, 22]. Table 2 summarizes performances of the optimal model and the baseline model on the training and test datasets, respectively. The baseline model predicts the future daily oil production by copying the latest real daily production, i.e. the 30-day-prior daily production here. The XGBoost model significantly outperforms the baseline model in both RMSE and R^2.

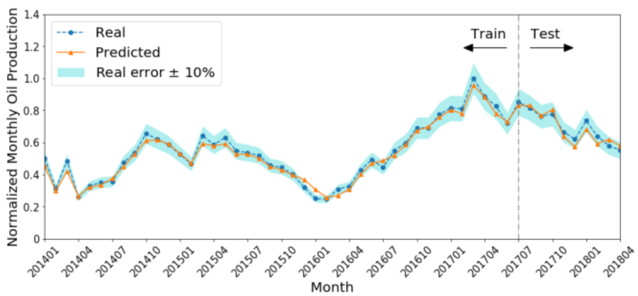

Figure 5: Comparison of the real productions with the predictions from the optimal 30-day-prior prediction XGBoost model.

Metric	Train	Test
RMSE	2.375	3.334
Baseline RMSE	4.295	4.361
$\boldsymbol{R^2}$	0.805	0.631
Baseline $\boldsymbol{R^2}$	0.361	0.368

Table 2: Comparison of performances of the optimal model and the baseline model on training and test datasets.

3.3 Optimization System

With a model capable of accurately predicting the oil production in different scenarios, optimization of the steam flood allocation will be straightforward. Making one pad with three infill wells as an example, all steam volume was injected into Infill Well 2 in reality, and the total real monthly oil production is 4202 m^3 (Table 3). Given this real input, the model predicts the oil production of 4242 m^3, 0.9% higher than the real value. To recommend an optimal steam-allocation plan to maximize the production, a brutal-force search of all possible scenarios is a simple solution. Considering a fixed total steam volume and three infill wells, there are two independent parameters, e.g. steam volumes to Infill Well 1 and Infill Well 2. Figure 6 plots the oil production as functions of relative steam volumes of Infill Well 1 and Infill Well 2 in percentage. The bottom right dashed circle corresponds to the actual scenario, while the left circle indicates the optimal scenario of 27% steam injected to Infill Well 1, 4% to Infill Well 2, and 69% to Infill Well 3, where the model predicted the maximum oil production of 4340 m^3 (Table 3), a 3.3% improvement compared to the real production.

In future works, considering the brutal-force searching time will exponentially increase with the number of infill wells, other optimization algorithms, e.g. gradient descent searching, could be an alternative solution especially for pads with a large number of infill wells. Moreover, this optimization system is built with the objective function of maximizing oil production. If other objective functions are defined, such as minimizing steam-oil ratio to save fuel costs, different optimization strategy could be employed.

	Real	Optimal
Infill Well 1	0%	27%
Infill Well 2	100%	4%
Infill Well 3	0%	69%
Real Oil Production	4202	NA
Predicted Oil Production	4242	4340

Table 3: Real and optimal scenario comparison.

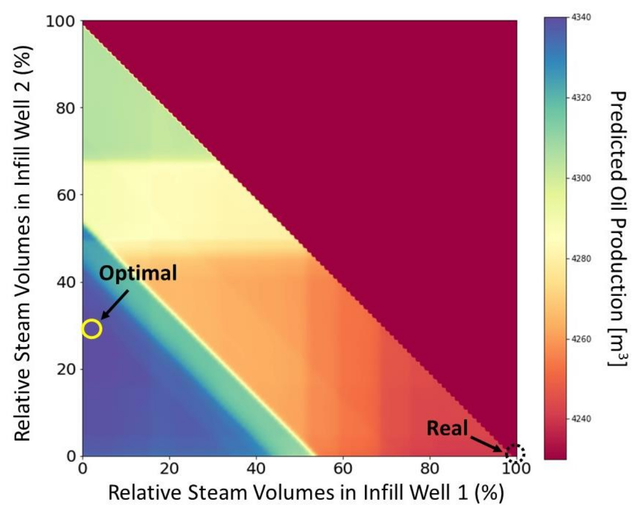

Figure 6: Variation of the oil production with relative steam volumes injected into Infill Well 1 and Infill Well 2.

4 CONCLUSION

We have designed a workflow including data engineering process addressing a category of time-series data and a machine learning algorithm XGBoost. This model can predict the oil productions in specific steam flood scenarios with an unprecedented accuracy, compared to traditional methods. Furthermore, we build an optimization system that can recommend the optimal steam-allocation plan with a potential 3% uplift in oil production.

Benefiting from the development of cloud platforms for IoT, we develop a cloud-based minimum viable product of steam flood optimization, which can achieve the real-time data collection, transfer and storage, as well as the machine learning model training and implementation in cloud platforms. This work opens up a new road for the study of the steam flood in the oil and gas industry.

It will be very interesting to explore applications of this workflow in other datasets with similar time-series data structures. For example, in a predictive maintenance case [26, 27], given historical records and future work statuses, this workflow can help build a machine learning model to predict the quantified abrasion level of a machine part.

ACKNOWLEDGMENTS

The authors would like to thank Charles A. Crawford, Richard J. Smith, Ryan C. Bouma, Dustin J. Bespalko, and Brian P. Vu for helpful discussions.

REFERENCES

[1] M. I. Jordan, T. M. Mitchell (2015). Machine Learning: Trends, perspectives, and prospects. Science, 349, 255-260.
[2] I. Goodfellow, Y. Bengio, A. Courville (2016). *Deep Learning*, MIT Press
[3] A. Colakovic, M. Hadzialic (2018). Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. Computer Networks, 144, 17-39.
[4] D. Singh, G. Tripathi, A. J. Jara (2014). A survey of internet-of-things: future vision, architecture, challenges and services. Proc. IEEE World Forum on Internet of Things, 287-292.
[5] M. H. Miraz, M. Ali (2015). A review on Internet of Things (loT), Internet of Everything (IoE) and Internet of Nano Things (IoNT). Proc. Internet Technologies and Applications (ITA), 219-224.
[6] N. Benamar, A. Jara, L. Ladid, D. E. Ouadghiri (2014). Challenges of the Internet of Things: IPv6 and network management. Proceedings of the 8th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 328-333.
[7] L. Atzori, A. Iera, G. Morabito (2010). The Internet of Things: a survey. Computer Networks, 54, 2787-2805.
[8] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami (2013). Internet of Things (IoT): a vision, architectural elements, and future directions. Future Generation Computer Systems. February 1645-1660.
[9] K. J. Bergen, P. A. Johnson, M. V. de Hoop, G. C. Beroza (2019). Machine learning for data-driven Discovery in solid Earth geoscience. Science, 363(6433), eaau0323.
[10] S. Kuhn, M. J. Cracknell, A. M. Reading (2018). Lithologic mapping using Random Forests applied to geophysical and remote-sensing data: A demonstration study from the Eastern Goldfields of Austrilia. Geophysics, 83, B183-B193.
[11] M. Valera, Z. Guo, P. Kelly, S. Matz, V. A. Cantu, A. G. Percus, J. D. Hyman, G. Srinvasan, H. S. Viswanathan (2018). Machine learning for graph-based representations of three-dimensional discrete fracture networks. Computational Geosciences, 22, 695-710.
[12] S. M. Mousavi, S. P. Horton, C. A. Langston, B. Samer (2016). Seismic feature and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression. Geophysical Journal International, 207, 29-46.
[13] P. M. R. DeVries, T. B. Thompson, B. J. Meade (2017). Enabling large-scale viscoelastic calculations via neural network acceleration. Geophysical Research Letters, 44, 2662-2669.
[14] M. Araya-Polo, J. Jennings, A. Adler, T. Dahlke (2018). Deep-learning tomography. Leading Edge (Tulsa Okla.), 37, 58-66.
[15] C. Xu, S. Misra, P. Srinivasan, S. Ma (2019). When Petrophysics Meets Big Data: What can Machine Do? SPE Middle East Oil and Gas Show and Conference, 195068.
[16] W. Z. Khan, M. Y. Aalsalem, M. K. Khan, M. S. Hossain, M. Atiquzzaman (2017). A Reliable Internet of Things based Architecture for Oil and Gas Industry. International Conference on Advanced Communications Technology, 19-22.
[17] M. Y. Aalsalem, W. Z. Khan, W. Gharibi, M. K. Khan, Q. Arshad (2018). Wireless Sensor Networks in oil and gas industry: Recent advances, taxonomy, requirements, and open challenges. Journal of Network and Computer Applications, 113, 87-97.
[18] V. Alvarado, E. Manrique (2010). Enhanced Oil Recovery: An Update Review. Energies, 3(9), 1529-1575.
[19] E. M. A. Mokheimer, M. Hamdy, Z. Abubakar, M. R. Shakeel, M. A. Habib, M. Mahmoud (2019). A Comprehensive Review of Thermal Enhanced Oil Recovery: Techniques Evaluation. Journal of Energy Resources Technology, 141, 030801.
[20] S. M. Farouq Ali, R. F. Meldau (1979). Current Steamflood Technology. Journal of Petroleum Technology, 31(10), 1332-1342.
[21] A. Dutt, A. Mandal (2012). Modified analytical model for prediction of steam flood performance. Journal of Petroleum Exploration and Production Technology, 2, 117-123.
[22] Z. Ning, H. Liu, H. Zhang (2007). Steam flooding after steam soak in heavy oil reservoirs through extended-reach horizontal wells. Petroleum Science, 4(2), 71-74.
[23] M. Q. Hama, M. Wei, L. D. Saleh, B. Bai (2014). Updated Screening Criteria for Steam Flooding Based on Oil Field Projects Data. Paper SPE, 170031.
[24] L. Breiman (2001). Random Forests. Machine Learning, 45(1), 5-32.
[25] T. Chen, C. Guestrin (2016). XGBoost: A scalable tree boosting system. In Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.
[26] R. K. Mobley (2002). *An Introduction to Predictive Maintenance*, Butterworth-Heinemann.
[27] A. Widodo, B. Yang (2007). Support vector machine in machine condition monitoring and fault diagnosis. Mechanical System and Signal Processing, 21, 2560-2574.