SPE-210983-MS

Cloud-Based Machine Learning System for Large Scale Implementation of Virtual Meters for Production Optimization

Yujun Johnny Chen, Nikhil Gulati, and Subbu Annamalai, Baker Hughes; Amir Delgoshaie, Zhaoyang Larry Jin, and Riyad Muradov, C3.ai

This paper was prepared for presentation at the ADIPEC held in Abu Dhabi, UAE, 31 October – 3 November 2022.

Abstract

This paper proposes a virtual metering (VM) system using machine learning and well production data for estimation of emulsion, produced gas, and vapors for steam-assisted gravity drainage (SAGD) production system. The system consists of four types of virtual meter models for every well: Emulsion Rate VM, Produced Gas Rate VM, Steam Vapor Rate VM, and Flash Vapor Rate VM. Emulsion Rate VM is used to estimate the oil rate output, while the other three VM types are used as the risk level indicators. For optimal well production and well management, operations need to consider both oil production and risks at the same time. Providing both well rate estimation and ability to assess risk at near-real time is a challenging task for large production fields, especially when the data is sparse and there is uncertainty in the well data. We propose a comprehensive system that utilizes a scalable framework to develop and deploy data-driven but process-informed virtual meters at scale and at near-real time. Specifically, in this paper, we describe a) a robust well data-preprocessing pipeline developed to continuously process the stream of input and target data for online learning. b) a state-of-the-art modeling framework designed to combine and segment large amounts of training data from the production filed. c) a scalable and flexible framework to ensure stable model performance with low data latency. d) a streamlined domain-specific model monitoring and MLOps process to enable tuning and retraining of the models by monitoring the performance and data drift.

Introduction

The ever-increasing energy demand is promoting the evolution of oil and gas (O&G) production methods to become more data driven. Physics-based approaches with equations derived from first principles are reliable but they are time-consuming to develop and maintain. These techniques require constant calibration to tune for the operational regime change or require expensive physical meters. More ML use cases in oil and gas have emerged in the last decade, such as predictive maintenance [1] and production forecast and planning [2]. All those applications aim to provide better resource optimization, higher operation efficiency, as well as lower carbon footprint for the oil and gas industry.

For production optimization, multi-phase virtual flow meters are proven to be an effective solution to --a well-test-based operations [3,4]. These operations rely on periodic production tests to update the flow rate records. Parameters such as emulsion rate, gas, and vapor rate, are only available from test separators. It is impractical to install a test separator for every well. Field engineers must rely on the test results to estimate past flow rates and use empirical methods to forecast production and do planning. Empirical methods are difficult to standardize and promote across the organization. It could result in drastically different performance across fields. ML-based VMs, on the other hand, provide a standardized framework and domain relevant features to train models for individual wells by learning from the historical data. Although, state-of-the-art ML algorithms such as neural networks [5], gradient boosting [6], support vector machine [7], and random forest [8], have proven effective in modeling complex nonlinear systems, the efficacy of these models for domain-specific application such as in oil and gas is dependent on feature engineering to capture the physics of the production process.

The methods proposed in the prior literature advanced the modeling accuracy and usefulness of using machine learning for virtual flow metering but has not considered the post-deployment model maintenance and performance monitoring to the best of our knowledge. Successful initial model training increases user acceptance rate however, post-deployment model degradation issues often hinder users' interest in adopting these applications. Without a proper MLOps workflow, the performance of these models in production is not guaranteed.

MLOps is increasingly important for developing large complex ML systems. The core idea of MLOps is to streamline the workflow of managing different components or phases in a ML application's lifecycle [9]. It could be applied to a small-scale application with a few models or an application with hundreds of models that run simultaneously. The complexity of model management increases as the number of models increases. For a large ML system, it is impractical to manually monitor all models' performance. It is also impractical to retrain all degraded models offline because pausing the prediction jobs could have a negative effect on the downstream modules. Therefore, an automatic monitoring and retraining strategy directed to production optimization for oil and gas is required to ensure a seamless operation.

The paper presents a framework and a virtual meter system developed for SAGD operation. SAGD is an in-situ thermal recovery technology where two horizontal wells are positioned near the bottom of the formation, and they are separated by a vertical distance. The top well is called the injector well, and the bottom well is called the producer well. Steam is pumped into the injector well and heat up the heavy oil in the formation. The produced liquid, such as oil, formation water, and concatenate, drain down to the producer well and get pumped up to the surface [10]. The representative SAGD production field in this paper consists of over 270 wells. There are 4 parameters production engineers are interested in: emulsion rate, produced gas rate, steam vapor rate, and flashed vapor rates. The emulsion rate is used to calculate the production KPI. The gas and vapor rates are the risk level indicator. On an average, each well goes through a production test every week. Between well tests, there is no production data available for the key indicators and oil rate listed above. Therefore, engineers and operators rely on past tests to estimate current flow rates and make decisions to adjust well production. A conservative approach is maintaining a constant set point between tests so that gas and vapor rates can be presumed steady. However, this strategy limits production. Virtual meters provide an alternate approach to estimate these rates between well-tests and without using expensive multiphase flow meters.

System Architecture

As an integral part of the production optimization application, VMs provide 24/7 predictions on the target flow rates. The VMs outputs are utilized for calculation of well production KPIs, real-time alarm, set points optimization, as well as control system automation. Figure 1 describes the relationship between each module in the software.

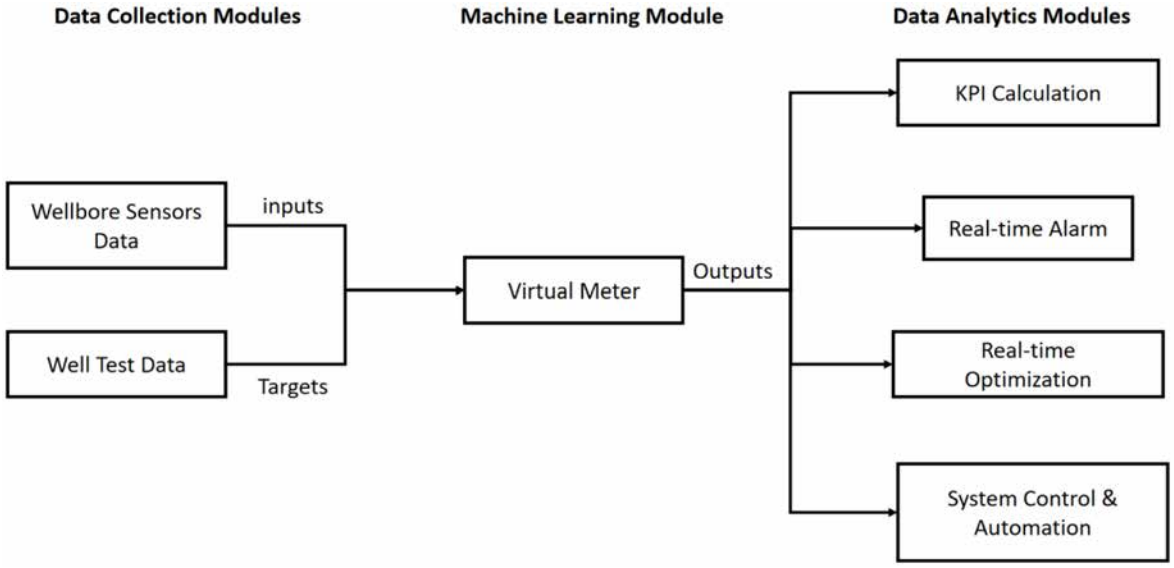

Figure 1—Overview of the application's modules

Data Collection Modules

Real-time sensor data are collected by the Pi server which is part of Operational Technology (OT) systems in an oil company. OPC Unifed Architecture (OPC-UA) server ingests the sensor data to OPC Publisher using OPC-UA Tunnel and routes to edgeHub, which persists the non-volatile data and translates the data to Avanced Message Queuing Protocol (AMQP). AMQP messages are received by Event Hub in the cloud, process the messages and stored in the Cassandra time-series database. To facilitate live values available for the ML predictions, Event Hub data is made available through the Live Values Server. The process is describedd in Figure 2.

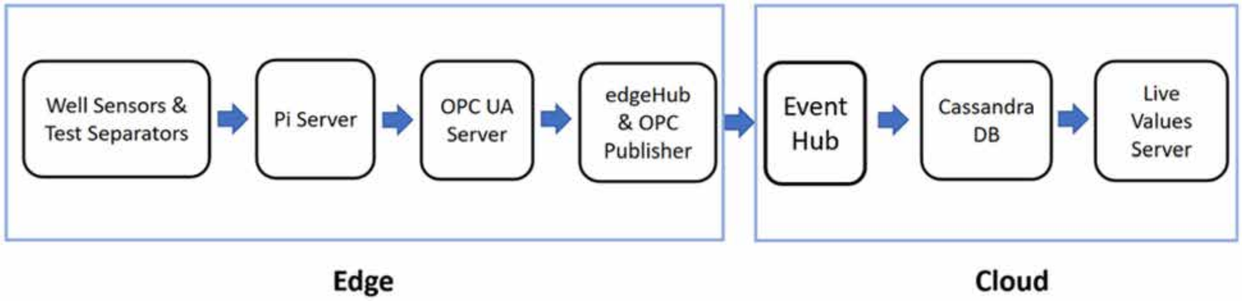

Figure 2—Data collection process

Wellbore sensors data and well test data serve as the input variables and target variables, respectively. After the data is collected and streamed to the cloud, they are staged for the next step of machine learning.

Machine learning Modules

The ML module serves as the bridge connecting the raw data collected from the field to the downstream analytics as shown in Figure 1. The details are explained in the methodology part.

Data Analytics Modules

KPI calculation is important for both the engineers and operators to get insight of the production level and report it to the management. This feature utilizes the live VM predictions to frequently update all metric calculations on a UI dashboard.

In practice, increasing steam injection and higher pump speed will result in increased production rate, while producing more gas and vapor. For operational safety, as well as regulation compliance, the risk-indicator parameters are monitored 24/7 by the alarm system. The application also provides notifications to the engineers when any of the parameters breaches a threshold.

We enable near real-time optimization by continuously estimating the production rate and monitoring the risk levels for each individual well. In the absence of a virtual meter, rates are either the production rates are presumed constant between tests or are the variability is unobserved. Therefore, any intervention or change in setpoints are avoided during the off-test periods.

Methodology

As mentioned in the System Architecture section, VM module connects the data collection modules to the analytics modules. It is important to design a robust domain relevant ML system with a proper MLOps workflow to streamline all components in the application.

Data Preprocessing

The raw data needs to be aggregated and processed into a suitable format for model training. Since the target flow rates are collected from weekly well tests, they are discontinued time series data shown in Figure 3. The reprocessing pipeline aggregates and combines all well test data to form a training dataset.

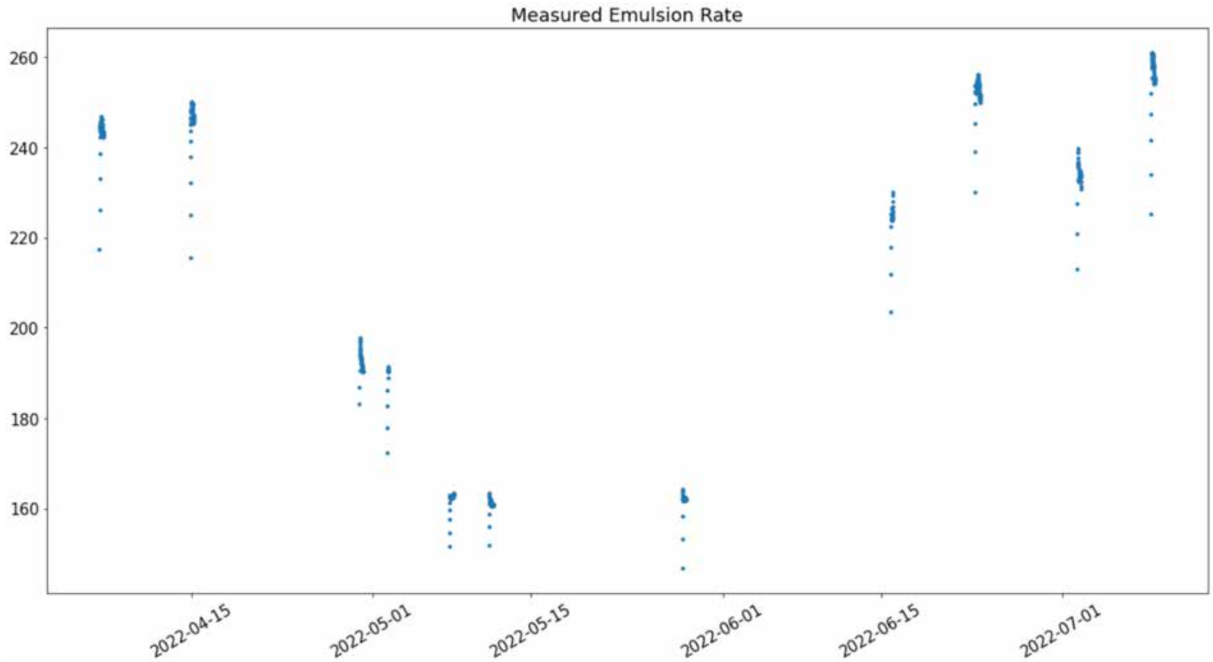

Figure 3—Emulsion rate used as a target training set

The input data, such as production pressure, temperature, and pump speed rate, are available from the wellbore sensors module. Pressure and temperature data have a domain shift problem when the production pipe is changed from a wellbore environment to a test separator environment. The reprocessing pipelines normalized the pressure and temperature data by calculating the delta between the well-test average and the wellbore average on both sides as shown in Figure 4. The normalized times series data is shown in Figure 5.

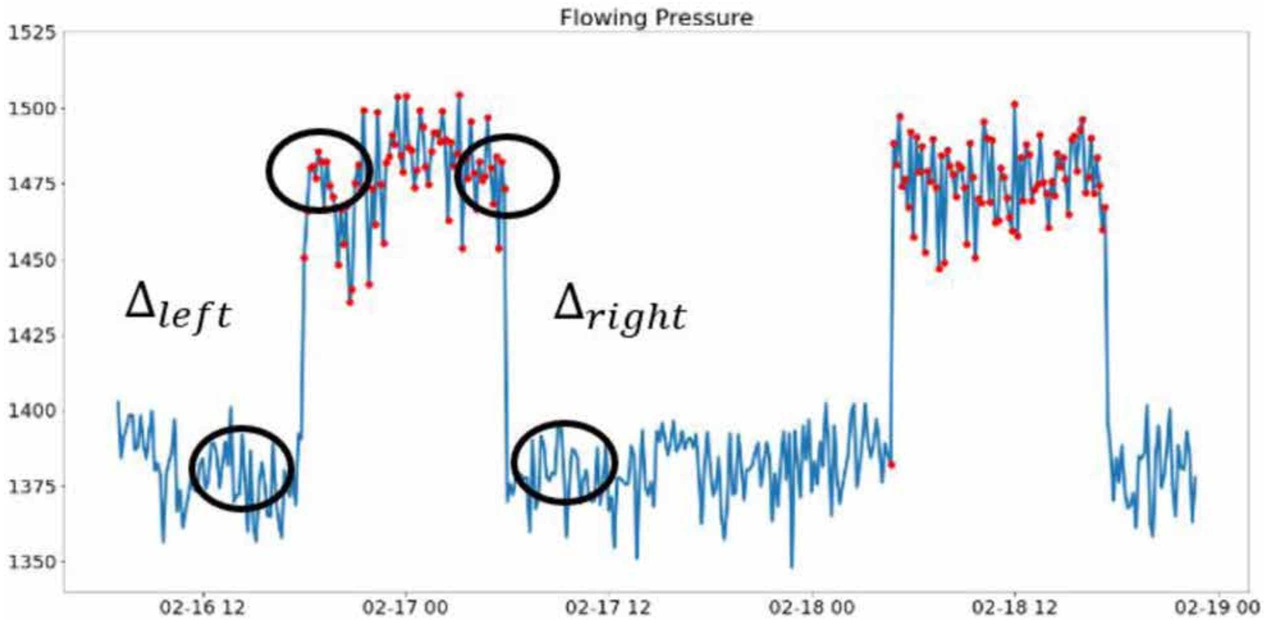

Figure 4—Pressure data before normalization

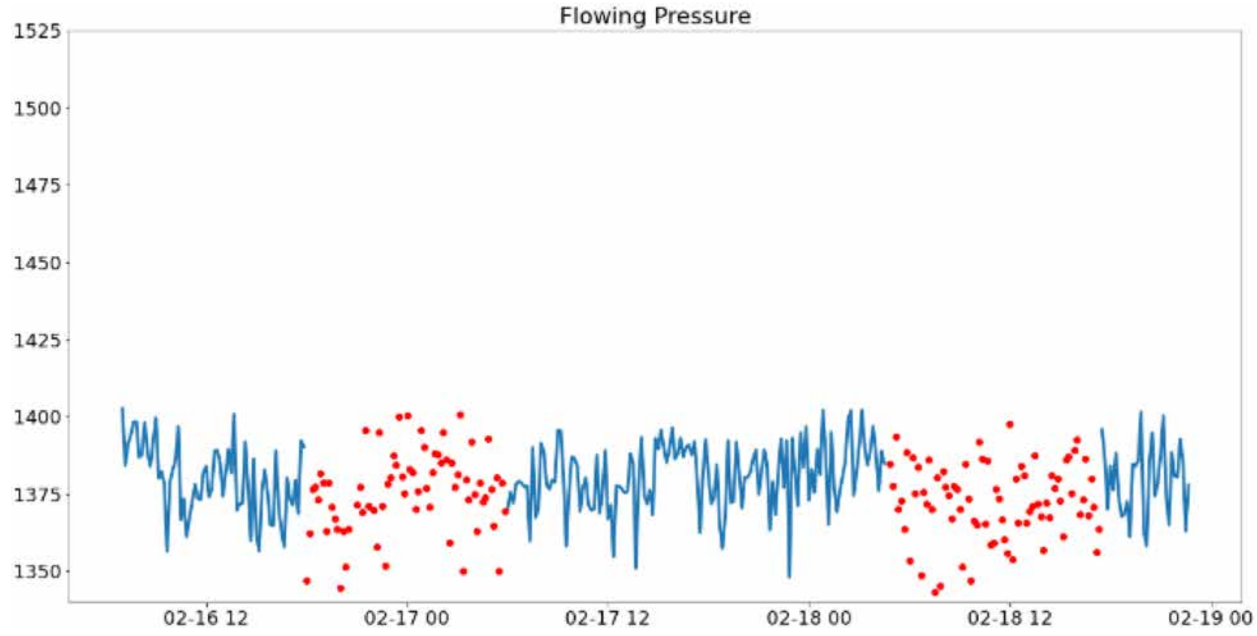

Figure 5—Pressure data after normalization

Features Selection and Modeling

Pearson's coefficient is used for feature selection. O&G domain knowledge is taken into consideration to create engineered features. Each VM type has its own feature set, and it is applied to all the wells. The VM models are based on XGBoost Regression. XGBoost scales with large training data size by parallelization. Parallelization happens during the construction of individual trees [11]. In addition, loading dataset as a DMatrix significantly increases the training speed [12]. In terms of computation, optimal resource utilization is made possible by the MapReduce algorithm that distributes training jobs over a cluster of servers. With these setups, the ML pipeline can handle hundreds of model training jobs simultaneously. Most VM models are trained with 3-year historical data. Young wells that are recently onboard need to wait at least 6 months to start training. For hyper-parameter optimization, sklearn's gridsearch method is used to search for the best hyperparameter set for each VM model.

MLOps

The VM module is designed with two major goals: large-scale implementation and automation of online learning. For initial training, models can be built in a Jupyter notebook environment where data scientists have the flexibility to experiment different methods and adjust model parameters accordingly. Each VM type adopts one template. Each template provides a underlining model framework, feature set, and metric rules. With templates, scaling model to the entire field is efficient and quick. After a new well is onboard, the ML pipeline uses the template to load the feature set and target variables to the training engine and start the job. Once models are deployed to production, they are more difficult to tune than when initially trained because models cannot be taken offline arbitrarily. Therefore, to achieve a large-scale implementation, besides a robust data processing pipeline and a distributed computation framework, the system needs a mechanism to continually evaluate and update models to maintain their performance.

Online machine learning is a popular method for model management in a production environment. In comparison, offline learning is suitable for static systems with no or little data drift. Those systems require little maintenance after model deployment, and model updates can be done by schedule. For large dynamic systems like the SAGD process, it is impractical to adopt offline learning because it is time-consuming to re-tune all models manually. It is also insufficient to rely solely on a fixed schedule to update model. Model degradation could happen frequently due to senor malfunction, equipment recalibration, and data corruption, etc. Waiting for the next scheduled update could potentially impact production due to poor prediction accuracy. Therefore, health monitoring is needed to detect degraded models and re-tune them on demand.

Besides automating online learning, the system should implement a policy to take a degraded model offline for troubleshooting when online retraining no longer improves its accuracy. For example, when a well is installed with a new pump, it is likely that the VM accuracy will deteriorate. Such an event should trigger an immediate response to take the model offline and inform users of the change.

The above-mentioned model management issues can be addressed by a streamlined MLOps workflow. MLOps is increasingly important for large ML applications due to their inherent complexity of models as well as wide scale implementation. MLOps includes monitoring model health, model retuning, offline maintenance, as well as emergency response. The paper proposed the following MLOps workflow shown in Figure 6.

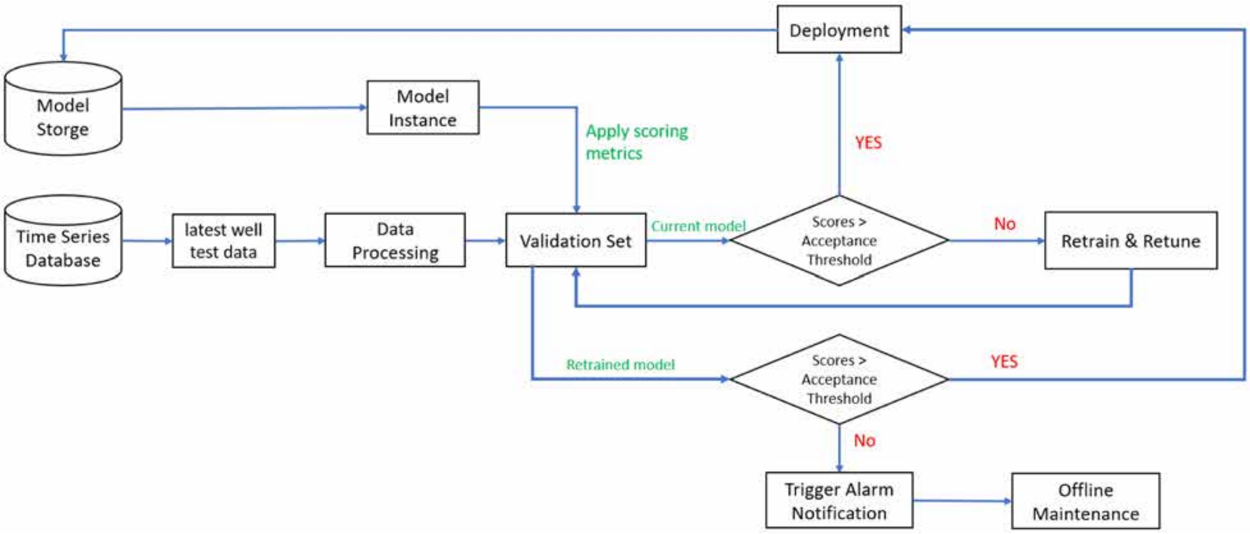

Figure 6—MLOps workflow

This workflow starts for each model after the initial training. Model validation is triggered after every well test finished. One complete cycle includes well-test identification, metric evaluation, retraining, and deployment. In every validation cycle, a model is extracted from the model file storage. The latest well test is used to calculate the model score. If the score is above the minimum acceptance threshold, it will be deployed back to the field for prediction. If the score is below the acceptance threshold, it will enter the first phase of retraining process. The first phase is an automatic re-tuning by a pre-configured job. It invokes the grid search process to recalculate the best hyperparameters based on the latest training set. Once the job finishes, it returns the newly trained model to the test set. If the model passes the acceptance threshold, it gets deployed to the field. However, if the model still fails, it triggers the alarm notification. A maintenance ticket is created, and the response team takes the model offline for troubleshooting.

There are several advantages of this MLOps workflow:

I. It utilizes, to the largest extent, the automation of online learning. The degraded models get updated immediately when new well tests are available. No human effort is involved in this process.
II. Due to the complexity of the large ML system, the notification system allows quick response and fixing on the bad-actor models. It reduced the downtime caused by under-performed models impacting the production environment.
III. The validation process records the model scores for all well tests. It generates a health report for ad-hoc analysis. If a model is performing above acceptance level but degrading. The Ops team could intercept the model and troubleshoot while keeping the model online for production.

Results and Use Cases

Production Monitoring

Before the VMs are implemented, the only way to know the emulsion rate is through well tests. Since the well tests are intermittent, any action such as speeding up a pump will not be reflected by data until the next test starts. Engineers and operators must rely on their experiences to estimate the emulsion volume. It was common to hold a pump speed constant so that emulsion rate can be assumed constant between well tests. However, this is not necessarily true because the emulsion rate also gets affected by other factors such as pressure fluctuation, wellbore fluid level, and steam injection rates.

Well tripping caused by pump failure sometimes happens between well tests. Once a well is tripped, production goes down to zero. If a well is about to go into a test, engineers could wait until then to restart the pump. But if the well is far from the next test, engineers have to restart without having the visibility of the flow rate. This is challenging for production planning.

This scenario is demonstrated in Figure 7. With VMs, engineers have full visibility on the emulsion rate 24/7. The graph shows a well trip scenario where engineers could restart the pump and receive feedback immediately before a well test.

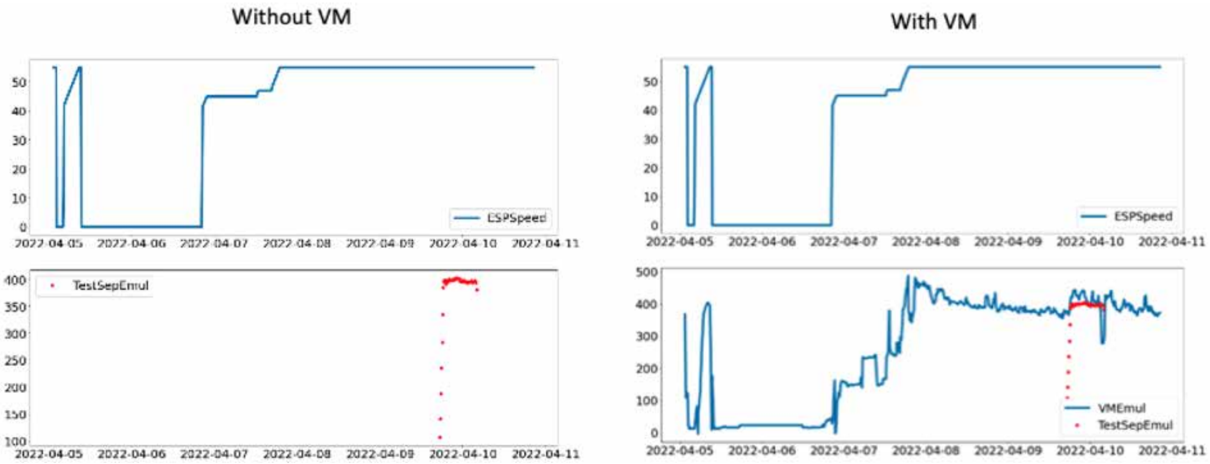

Figure 7—Emulsion VMs used for production monitoring

Risk Mitigation

In practice, speeding up a pump could produce more emulsion, but it also produces more gas and vapor. It is important to maintain a good balance between having a good production rate and minimal risk level. Figure 8 shows that without VMs, the two well tests did not trigger any alert. But with VMs, there are a few places between the well test breached the threshold and alert triggered. The VMs helped mitigate the risk by providing real-time feedback to the field operators. In this scenario, it is recommended to slow down the pump to lower the gas rate.

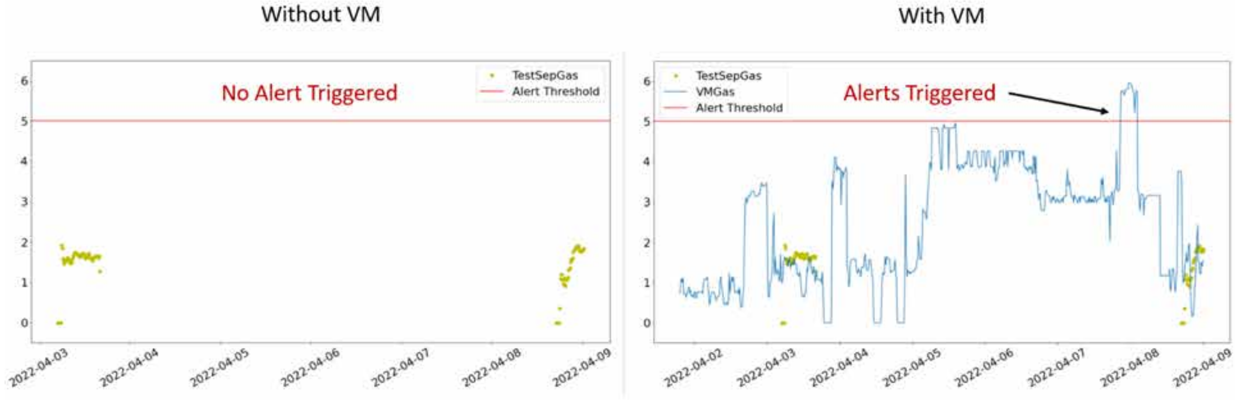

Figure 8—Produced gas VMs used for risk mitigation

Production Optimization

To optimize production, a reasonable set point need to be set based on different production scenarios. The goal of the PO is not only to predict flow rates accurately but also recommend set points for engineers control the wells. Figure 9 shows the optimizer recommended different actions based on the VM predictions. When the well was low on gas and vapor, a speed-up action was recommended to increase production rate. Once the risk levels were detected over the limit, a slow-down decision was made to decrease the production. The optimizer kept adjusting the set point to maintain the risks at their limits while maximizing the oil output rate.

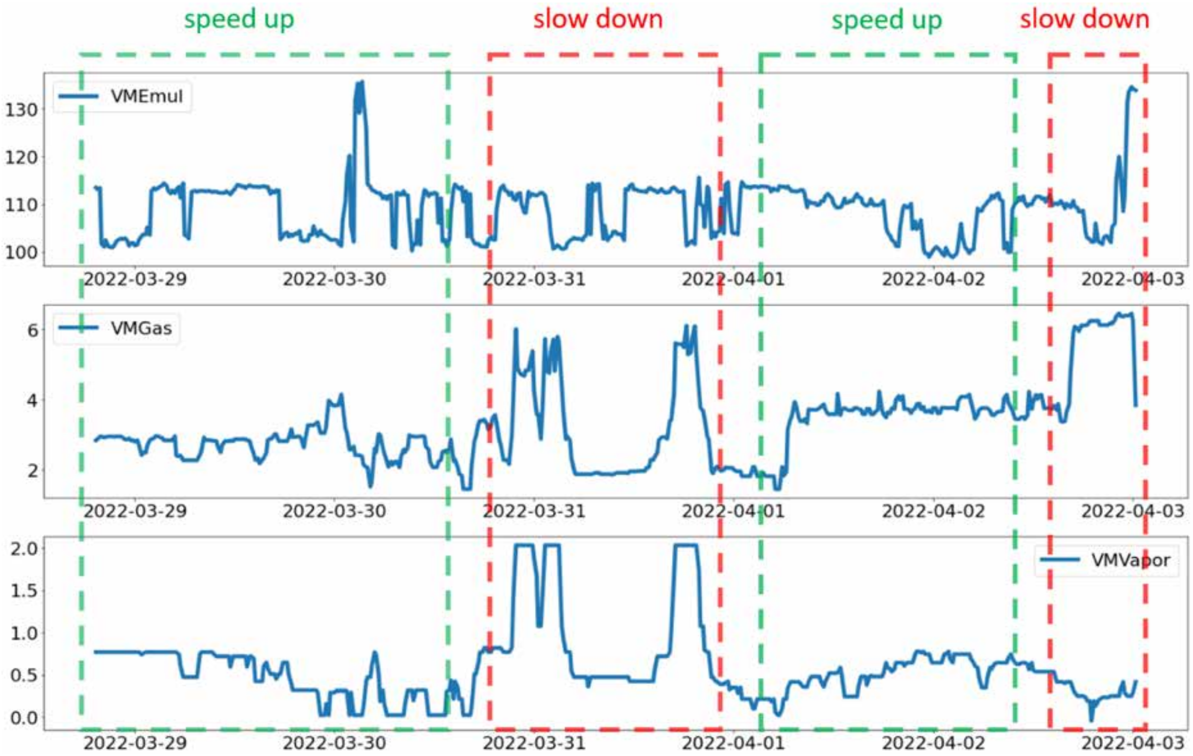

Figure 9—Set points recommendation for production optimization

Conclusion

We have proposed and demonstrated a successful application of VMs for SAGD production optimization. In the absence of VMs, well tests are used to measure flow rates such as emulsion, gas, and vapor. Due to their intermittent nature, conservative approaches were often adopted to maintain steady flow rates. These strategies allow engineers to better manage risks but also limited production. VMs provide full visibility on a well's operational status by predicting the targeted flow rate accurately. With real-time predictions, engineers can make informed decisions and plan production more proactively. The VM method can be extended to other conventional and unconventional oil and gas production.

Nomenclature

AMQP	Advanced Message Queueing Protocol
DevOps	Development operations
IoT	Internet of Things
ML	Machine Learning
MLOps	Machine learning operations
OPC-UA	OPC Unified Architecture
OT	Operational Technology
SAGD	Steam Assisted Gravity Drainage
VMs	Virtual Meters

Reference

1. Abbasi, Tayaba, King Hann Lim, and Ke San Yam. "Predictive maintenance of oil and gas equipment using recurrent neural network." *Iop conference series: Materials science and engineering*. Vol. **495**. No. 1. IOP Publishing, 2019.

2. Qiao, Yudeng, et al. "Application of PSO LS-SVM forecasting model in oil and gas production forecast." *2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE, 2017.
3. Garcia, Alejandro, et al. "An implementation of on-line well virtual metering of oil production." *SPE Intelligent Energy Conference and Exhibition*. OnePetro, 2010.
4. Ursini, Fabrizio, et al. "The benefits of virtual meter applications on production monitoring and reservoir management." *SPE Reservoir Characterisation and Simulation Conference and Exhibition*. OnePetro, 2019.
5. El-Abbasy, Mohammed S., et al. "Artificial neural network models for predicting condition of offshore oil and gas pipelines." *Automation in Construction* **45** (2014): 50-65.
6. Bikmukhametov, Timur, and Johannes Jäschke. "Oil production monitoring using gradient boosting machine learning algorithm." *Ifac-Papersonline* **52.1** (2019): 514-519.
7. Anifowose, Fatai, AbdAzeem Ewenla, and Safiriyu Eludiora. "Prediction of oil and gas reservoir properties using support vector machines." *IPTC 2012: International Petroleum Technology Conference*. European Association of Geoscientists & Engineers, 2012.
8. Liao, Lulu, et al. "Data mining: a novel strategy for production forecast in tight hydrocarbon resource in Canada by random forest analysis." *International Petroleum Technology Conference*. OnePetro, 2020.
9. Zhou, Yue, Yue Yu, and Bo Ding. "Towards mlops: A case study of ml pipeline platform." *2020 International conference on artificial intelligence and computer engineering (ICAICE)*. IEEE, 2020.
10. Nasr, Tawfik N., et al. "Novel expanding solvent-SAGD process ES-SAGD." *Journal of Canadian Petroleum Technology* **42.01** (2003).
11. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016.
12. Chen, Tianqi, et al. "Xgboost: extreme gradient boosting." R package version 0.4-2 **1.4** (2015): 1-4.