PLUNGER LIFT STAGES SEPARATION AND VIRTUAL FLOW METERING GENERATION THROUGH MACHINE-LEARNING

Gustavo Carvalho, Raphael Viggiano, Eduardo Pereyra, and Cem Sarica
The University of Tulsa

Mike Micozzi and Wrangler Pankrantz
Ovintiv

ABSTRACT

The plunger lift process can be divided into four distinct cycles: buildup, upstroke, liquid discharge, and after flow. One key parameter that can be measured for optimizing oil production is the total gas and liquid flow rate produced while the valve is open. Typically, the only known parameters are the controller's on and off times, so post-processing is required to identify the liquid discharge period and quantify the observed flow rate.

Human analysis is sufficient to identify when liquid discharge occurs, which is characterized by a sudden increase in gas flow rate. Analyzing a single well over a period of time is feasible; however, evaluating tens or hundreds of wells is impractical. This study proposes a machine-learning approach based on neural networks to automatically split plunger lift cycles and predict gas flow rate. The model employs a long short-term memory (LSTM) neural network, commonly used for time-series data, for regression or classification tasks. The model's input is a time window containing casing and tubing pressures, along with gas flow rate data; its output consists of probabilities corresponding to each plunger cycle for the classification, or gas flow rates for the regression. After the automatic cycle splitting, the cumulative gas flow rate during the liquid discharge period is quantified and recorded.

To train the classification model, field data must be acquired and manually labeled by a subject-matter expert. To automate this part, a graphical user interface (GUI) was developed to load well data and interactively select the corresponding plunger stage. The model was trained/tested on data from six wells, achieving less than 2% in the gas flow rate prediction, and more than 98% accuracy in predicting the stages.

This study presents an efficient, automated method to address a common challenge in production monitoring: quantifying well performance. Once trained, the proposed neural network can rapidly classify real-time data, enabling improved troubleshooting, production optimization, and performance tracking.

Keywords: Plunger-lift, machine-learning, virtual flow meter.

1 INTRODUCTION

Plunger lift is an artificial lift method widely used to enhance production in gassy wells. The plunger creates a physical barrier between the gas and the liquid, decreasing the amount of liquid that falls back and enhancing liquid removal from the well, thereby increasing accumulated oil production.

The conventional plunger lift process is characterized by intermittent opening and closing of the well by a control valve, known as on and off time, respectively. The intermittent process presents four distinct stages:

- Buildup: the well is shut-in, and the plunger falls down through gas and liquid while the casing pressure builds up due to the gas production and accumulation in the well.
- Upstroke: after accumulating enough gas and building enough pressure, the well is opened. The differential pressure causes the plunger to rise, carrying the accumulated liquid and unloading the well.
- Liquid discharge: after traveling the well, the plunger and liquid reach the lubricator, and the tubing pressure increases due to the arrival of a second gas portion that resided below the plunger.
- After-flow: oil is effectively produced to the flowline, carried by the remaining gas in the well. The well is closed, and the cycle is repeated.

The off time contains the buildup stage, while the on time contains the upstroke, liquid discharge, and after-flow. Figure 1 illustrates one plunger lift cycle with hypothetical tubing and casing pressures, and the gas flow rate characteristic curves.

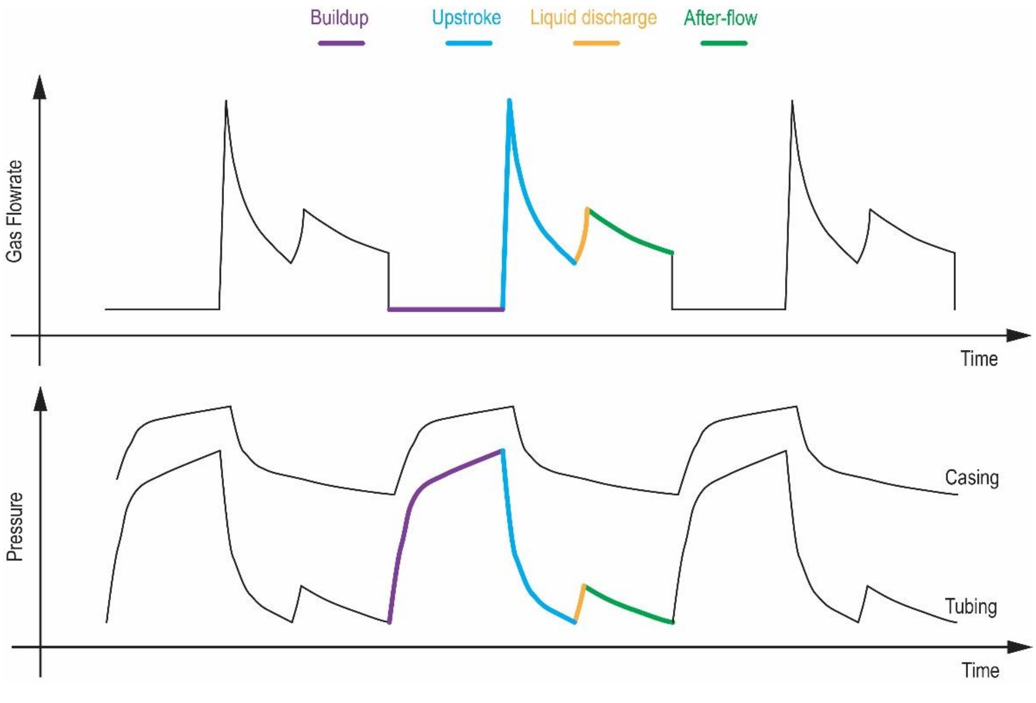

Figure 1 – Hypothetical conventional plunger lift cycle.

Identifying and categorizing each stage allows for the generation of a synthetic well response that can be later used to optimize the production of each individual well (Viggiano et al., 2024). The amount of liquid being produced can be directly related to the amount of gas produced. Cumulative gas production per cycle is an important variable to be measured and optimized (Akhiiartdinov et al., 2020). The main problem is how to quantify this quantity since there is no “smart” device to exactly measure the gas produced during the after-flow, especially if no sensor is available to detect the different stages. Therefore, one possible strategy is to split the plunger lift into cycles and analyze the after-flow stage individually.

These stages are relatively easy to identify and split for a single cycle; however, it becomes difficult to identify and split cycles for tens or hundreds of wells with real-time production data, especially if no sensor to measure plunger arrival time is available. If no auxiliary instrument is available, such as Ecometer, the raw pressure and flow rate signals must be post-processed to perform the cycle splitting. It is generally done using peak-detection algorithms, which are prone to errors, especially when the available data is noisy. To optimize this process and ensure its applicability across a broad range of operational scenarios, a robust computational algorithm is needed.

Machine learning and artificial intelligence-based models are widely used to improve the understanding of artificial lift methods. Successful implementations have been reported for virtual metering, failure prediction, and operational parameters optimization (Agwu et al., 2024). Artificial neural networks (ANNs) are powerful models that can learn the dynamics of real data during training. If the model is driven by its data, we call it a data-driven model, since no physical assumptions are considered in the process. ANNs are global function approximators that can be used to model specific processes. In supervised learning, data is generated first and then used to train and validate the neural network model.

Simple multilayer ANNs have been used to infer the gas flow rate from pressure data. Akhiiartdinov et al. (2020) used ANN to develop a model that estimated the production gas within 20% error, even without using any time-series treatment. However, they pointed out the difficulty of determining the optimal input parameters and architecture.

Recently, more specialized ANN architectures have been used; they combine multiple specialized layers to better process time-series data or perform feature extraction from the input data. Liu et al. (2025) and Liang et al. (2025) developed complexes ANN architecture for anomaly detection in plunger lift operations. The developed model could identify real-time fault conditions and guide troubleshooting responses.

This study presents a machine learning approach for two important analyses of plunger lift data: prediction and optimization. The developed model consists of an ANN with a specialized architecture for time-series processing and feature extraction, capable of performing regression and classification tasks with plunger lift data. The regression task is used to create a virtual flow meter that infers gas flow rate from pressure measurements, while the classification task is used to split plunger lift cycles and stages, preparing the data for further in-depth per-stage analysis.

This paper has the following structure: Section 2 presents the methodology for developing and training the ANN and for preparing the data. Section 3 presents the results of model training and its applications. Finally, Section 4 presents the conclusions of this work.

2 METHODOLOGY

The envisioned algorithm requires time-series measurements of gas-flow rate, and tubing, and casing pressures, as illustrated in Figure 1. Using these inputs, the algorithm identifies and classifies the stages within each plunger-lift cycle. Viggiano et al., (2024) used a peak-and-valley detection routine to delineate these stages; however, its performance is limited to datasets with low noise. When pressure or flow signals exhibit noise-induced spikes, the routine may incorrectly detect false peaks, leading to misclassification of stage initiation and termination. Consequently, a more robust and adaptive algorithm is needed. The new method must be able to learn from expert-labeled data and generalize reliably under noisy or evolving operating conditions.

Casing and tubing pressures are measurements available in most producing wells. Gas-flow-rate data are typically available when wells are routed to a test separator or, less frequently, when dedicated flow meters are installed. Importantly, the proposed algorithms in this study are not restricted to wells equipped with permanent gas-flow measurement devices. Gas-flow data obtained during well-test separator operations can be used for training, after which the trained model can infer production rates from only pressure measurements, effectively serving as a virtual flow meter. Furthermore, Viggiano et al., (2024) demonstrated that tubing-pressure data alone may be sufficient for estimating plunger-lift production when used as input to a properly trained model.

The next step is selecting a machine learning (ML) model for the proposed problem. To treat time-series data, a more sophisticated method is required; classical ML models (e.g., linear regression or random forest) are insufficient for learning temporal dependencies. Artificial neural networks are machine learning models that learn patterns from time-series data. Since ANNs can be present in distinct architectures, different models can be applicable to different kinds of problems. Various types of ANNs have been developed, and the selection of the architecture depends on the nature of the problem being addressed. ANNs can have specific types of architectures to handle specific data structures and learning tasks. The simplest architecture consists of interconnected neurons without recursion, known as a feedforward neural network. To model sequential data, recurrent neural networks (RNNs) were later introduced, enabling networks to incorporate information from previous time steps. However, traditional RNNs struggle to learn long-term dependencies due to issues such as vanishing and exploding gradients (Krichen & Mihoub, 2025). To solve this problem, long-short-term memory (LSTM) was proposed, enabling long-term information retention.

Long-short-term memory (LSTM) neural network can also be improved through the use of attention mechanisms (Vaswani et al., 2017), a concept widely adopted in modern deep-learning architectures, such as in large-language models. Attention mechanisms can significantly improve the LSTM performance by focusing on specific features, while the LSTM can concentrate on other data interactions (Krichen & Mihoub, 2025).

Based on this understanding and the power related to LSTM-Attention architectures, the ANN architecture presented in Figure 2 is proposed in this work for each task:

regression for gas flow rate prediction and classification for cycle splitting. In the regression task, time windows of pressure data from the tubing and casing are fed into the ANN to predict the gas flow rate at the last time point. In the classification task, time windows of pressure or gas flow rate are fed into the ANN to generate outputs that are later converted to probabilities of plunger cycles at the time position in the middle of the time window.

As we are dealing with a time-series problem, we must define a time window length. This length must be defined a priori, and this will define how much information (timed data points) is used as input for the model.

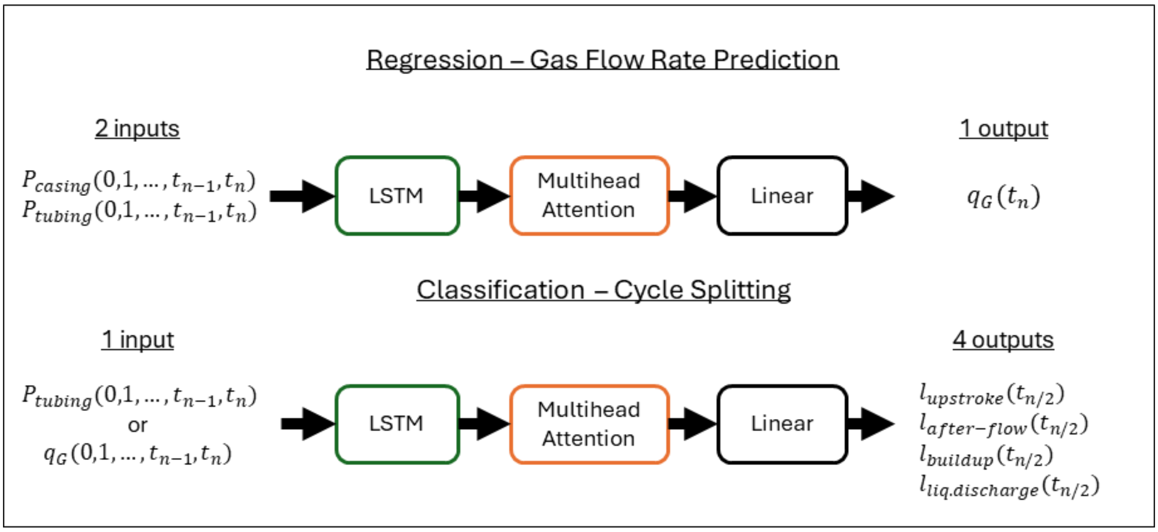

Figure 2 – ANN architecture for the regression and classification tasks.

Each layer specifies the following hyperparameters: the LSTM layer contains 2 layers with 50 neurons each, the attention layer contains 1 attention head with dropout set to 0, and the linear layer contains 1 layer with 50 neurons. Those values are defined by a trial-and-error procedure.

After defining the architecture, we must define the training and validation procedures. The process of training an ANN is not a deterministic problem; parameter initialization is random, and every training cycle is different from the previous one.

Data preparation is a necessary step to handle missing data and outliers, ensuring the ANN is trained on meaningful information. Also, the datasets are normalized using the mean and standard deviation of the training data (Z-score normalization).

The dataset is then divided into train and test sets. The training dataset is used to optimize the ANN parameters, while the test dataset is used to assess overfitting. The training loop repeats up to 100 times. Inside each loop, the training data is split into batches (32 for regression and 64 for classification) and passed through the model. If the training/validation error does not improve after 20 iterations, early stopping is triggered. At the end, the model with the lowest error is saved. The optimizer used in this study is the Adam model with a learning rate of 1.0×10^{-3}.

The ANN is developed using the PyTorch library in Python, enabling easy implementation and modification of the proposed architecture. The general training/evaluation procedure is depicted in Figure **3**.

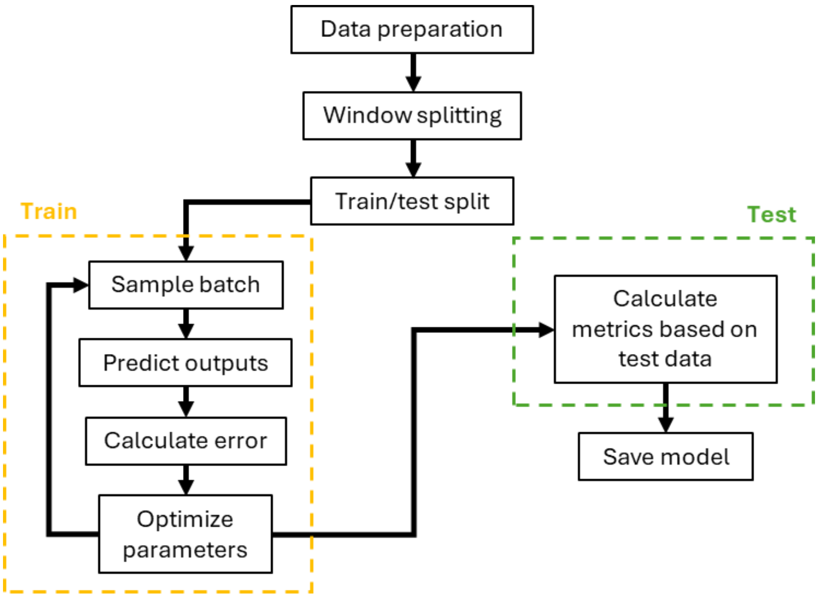

Figure 3 – General training and evaluation procedure.

In simple terms, the same neural network can work for both regression and classification tasks. The main difference between the two tasks is minor architectural corrections, such as the last-layer modification and the training loop, which requires different data for training. In the following sections, specific details for each task are provided.

2.1 Regression

For the regression task, the neural network maps the sequence inputs in the present to the expected gas flow rate. Pressure readings are the input, while the gas flow rate is the output. The data processing is depicted in Figure **4**, where each pressure sensor's raw signal is converted to a window with an adjustable size $n_{lookback}$, and the last gas flow rate is annotated for each respective extracted window. The model's input is the pressure signal sequence.

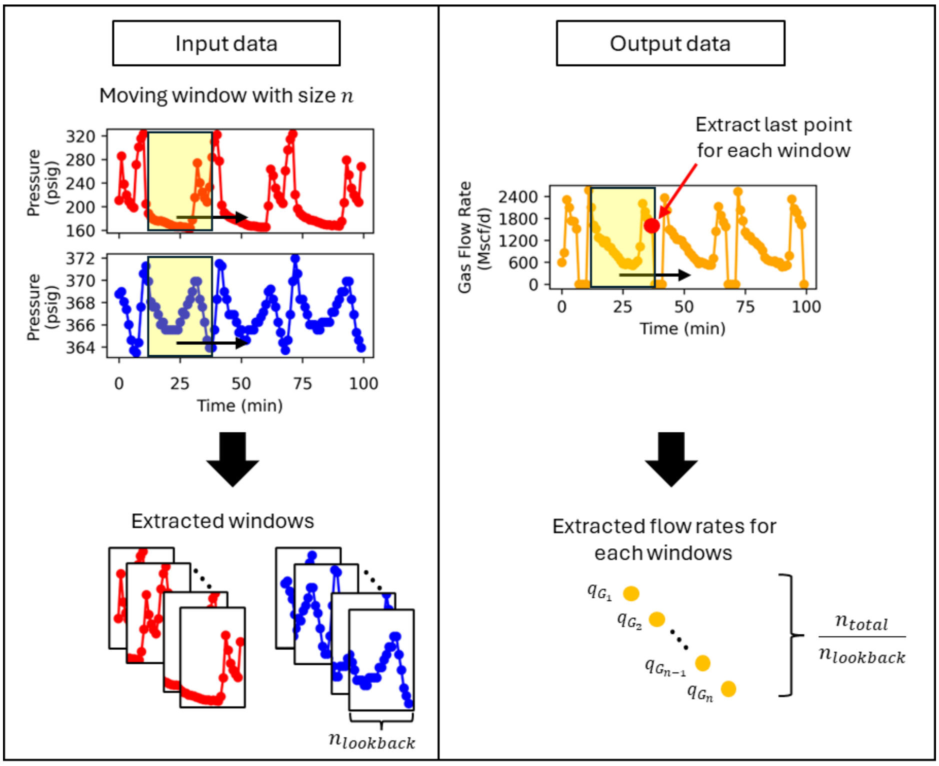

Figure 4 – Regression model data preparation.

The regression model output is later transformed into a mean squared error (MSE) to be minimized by the optimizer.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (1)$$

where y_i is the expected value, $\hat{y}_i$ is the estimated value by the ANN, and N is the total number of samples.

To evaluate each model's performance, different metrics must be defined. To assess the regression task performance, the following metrics are used: mean absolute error (MAE), the root mean squared error (RMSE), and the coefficient of determination (R^2):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i) \quad (2)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (3)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{4}$$

$\bar{y}$ is the average of the expected values.

2.2 Classification

For the classification task, the neural network maps the input to a specific pre-defined category (plunger stages). Pressure and gas flow rate readings are fed into the model to predict the stage of the data point in the middle of the sequence. Since this task involves offline data analysis, we use a bidirectional LSTM architecture that considers both forward and backward directions; in other words, it leverages past and future context to extract features from the data.

The training data is not readily available; it must be generated, often requiring a subject-matter specialist to label real data with known labels. The ANN model requires substantial data to learn different patterns; ideally, the more data available, the more powerful our model can be. Classifying a large amount of data requires significant time and resources. To optimize the generation of labeled data, a graphical user interface (GUI) was created to classify plunger lift stages based on pressure or gas flow rate data.

Figure 5 shows the GUI developed to guide manual labeling of the data, allowing unlabeled raw signals to be manually classified for training data generation. The GUI enables loading sensor data, selecting points with the mouse cursor, labeling data by stage, and saving the data.

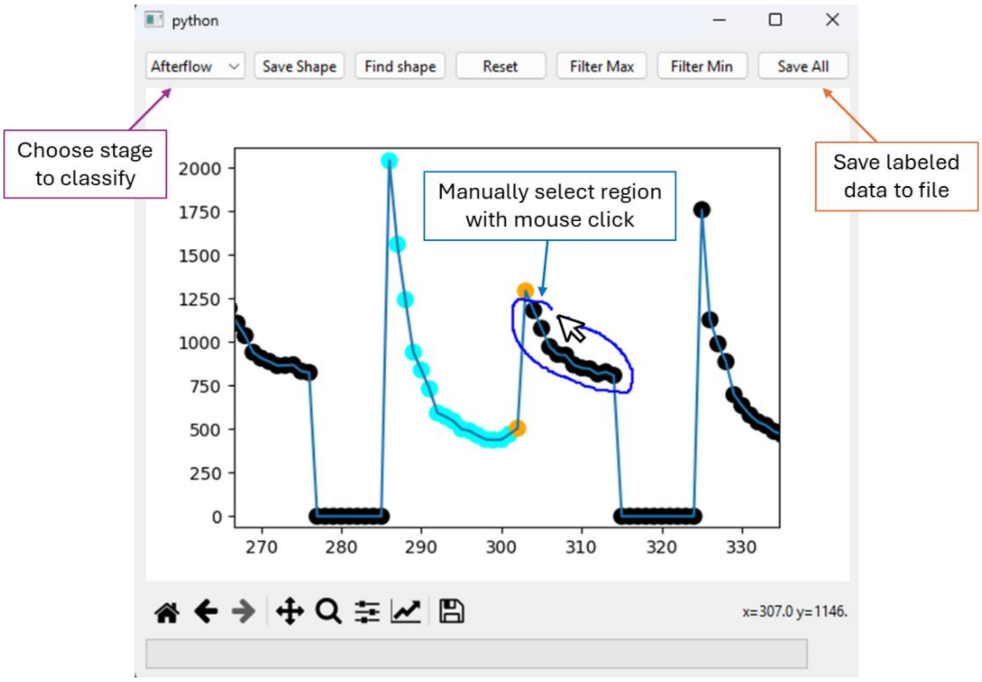

Figure 5 – User interface for labeling the plunger data in different stages

This labeling process is required only during the model-training phase. Because conventional plunger-lift cycles exhibit consistent behavioral patterns across wells, a well-constructed labeled dataset is expected to capture the vast majority of operating scenarios encountered in the field. Nevertheless, the dataset can be continually expanded, and the ANN can be retrained or fine-tuned as new data becomes available. This adaptability is one of the key advantages of the machine-learning approach: the model can improve over time, incorporating additional cases and evolving operating conditions to enhance prediction robustness and overall performance.

The data processing is depicted in Figure **6**. First, the data is labeled using the GUI previously discussed. Then, each dataset is split into windows with a length equal to approximately half of the average cycle time ($n_{well,lookback}$), and the stage of the middle point is stored for the classification task. One problem that may arise is the need to treat different window lengths; this issue is addressed by padding sequences shorter than the window size or cropping sequences longer than the window size.

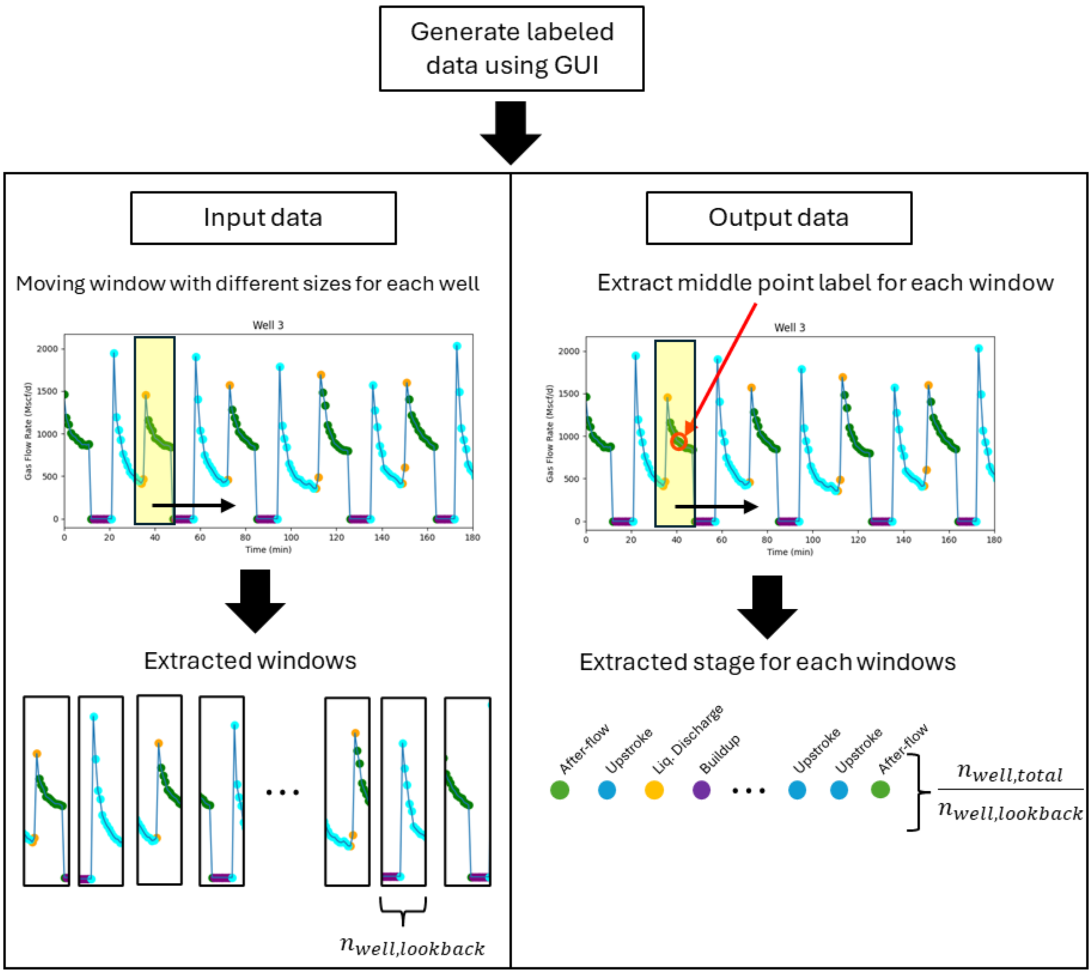

Figure 6 – Classification model training and evaluation.

The classification model output (known as logits) is later transformed into an error using a cross-entropy function:

$$Cross - entropy = -\frac{1}{N}\sum_{i=1}^{N}\frac{\exp(\hat{y}_i)}{\sum_{c=1}^{C}\exp(y_{i,c})} \quad (5)$$

where C is the number of classes/labels.

For the classification task, the metrics should change because now we are predicting labels, not a continuum variable. It is common to report values for precision, recall, and F_1-score, defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (8)$$

where TP stands for true positive, FP false positive, and FN false negative.

Precision indicates the proportion of the model's predicted values that are actually positive. The recall (or true positive rate) is the fraction of relevant instances retrieved. The F_1 score is the harmonic mean between precision and recall, and gives a balanced performance.

3 RESULTS

3.1 Regression: Gas flow rate prediction

Gas flow prediction is evaluated using data from a real well, hereafter referred to as Well 6. The amount of data used for the training, validation, and test are displayed in Table **1**. Each data point was collected with a 1-minute sampling rate.

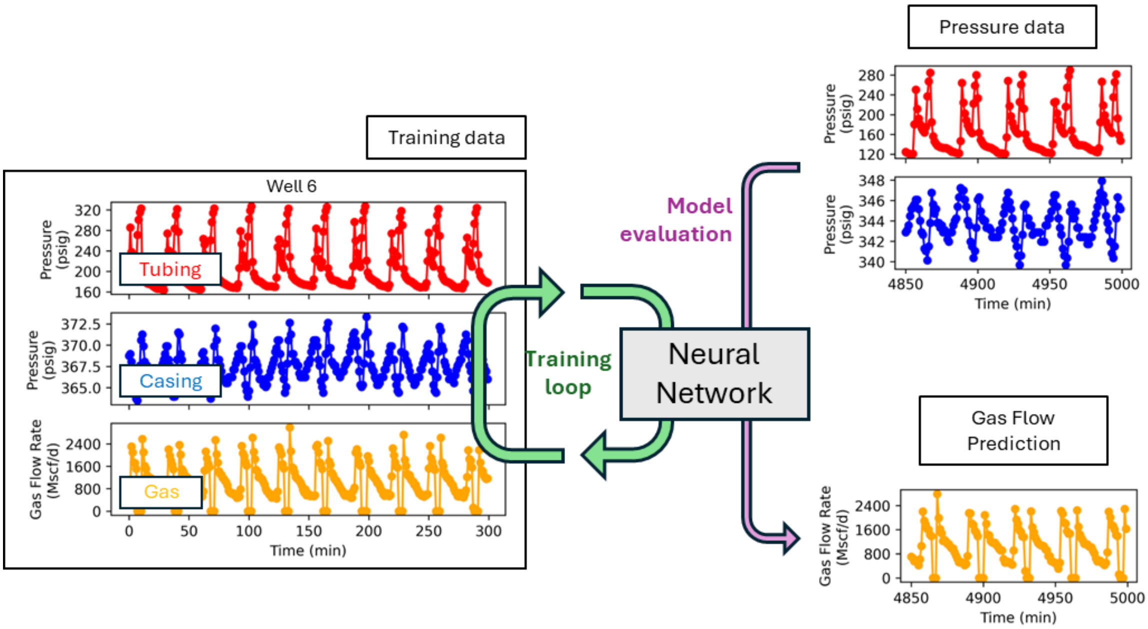

Figure 7 – Training and evaluation for the regression task

Table 1 – Data used for the gas flow rate prediction

	Amount of data points	Actual period of time (h)
Train	3500	58.3
Validation	750	12.5
Test	750	12.5

Figure **8** displays the average training results for the base ANN architecture. The training is carried out for 80 epochs, with each epoch consisting of several passes through the training batches to update the neural network parameters.

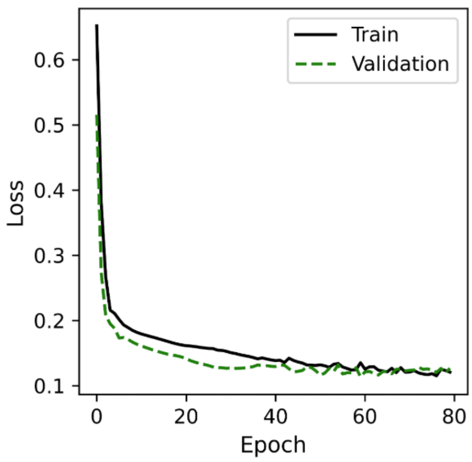

Figure 8 – Average training results for the base architecture with attention heads

The data indicates that approximately 58 h of production data is used to train the model. The gas flow rate measurement from the plunger-lift is nonlinear. Using linear regression to predict a more complex time-series data is highly prone to errors. Table **2** presents the scores obtained for the evaluation of the base case of the ANN architecture proposed in the last section. The Linear Regression and Random Forest models are shown here only to demonstrate their inability to handle time-series data.

Figure **9** displays a comparison of the expected values with the predicted values from different ML models. The first graph shows the predicted gas flow rate over time, while the second graph compares the expected and predicted values. Several predictions lie at zero expected gas flow rate, indicating the difficulty for all models to accurately predict the abrupt changes in gas flow when the valve is opened or closed.

Table **2** displays the values of each metric analyzed in the regression task. The regression task estimates the instantaneous rate of gas production, but the final variable of interest is the average amount of gas produced over the interval, given by:

$$q_{G,cumulative} = \frac{1}{\Delta t}\int_0^t q_G(t)dt \tag{9}$$

To obtain this quantity, the gas flow rate curves are numerically integrated and divided by the amount of time under analysis. This approach gives us the time-averaged gas production. In terms of the average gas production the difference between the true values and the LSTM+Attention architecture is less than 2%.

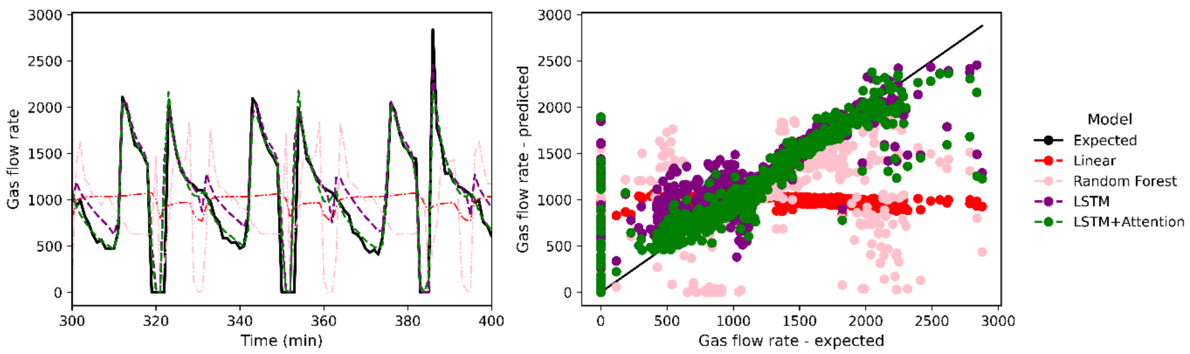

Figure 9 – Comparison of expected values and prediction from different ML models. (a) Model performance and (b) correlation plot.

Table 2 – Regression models scores

Model	MAE	$RMSE$	R^2	Average gas production MAPE (%)
Linear Regression	517	632	0.01	7.8
Random Forest	292	507	0.36	13.3
Neural Network (without Attention Head)	209	323	0.74	12.0
Neural Network (with Attention Head)	130	251	0.84	1.3

3.2 Classification: Stage splitting

Stage splitting is performed using the previously discussed classification architecture, where the output is converted to a probability of belonging to each stage. Data from five different wells are used to train the ANN, then the model is tested on a well that was not used during training, as indicated in Figure **10**.

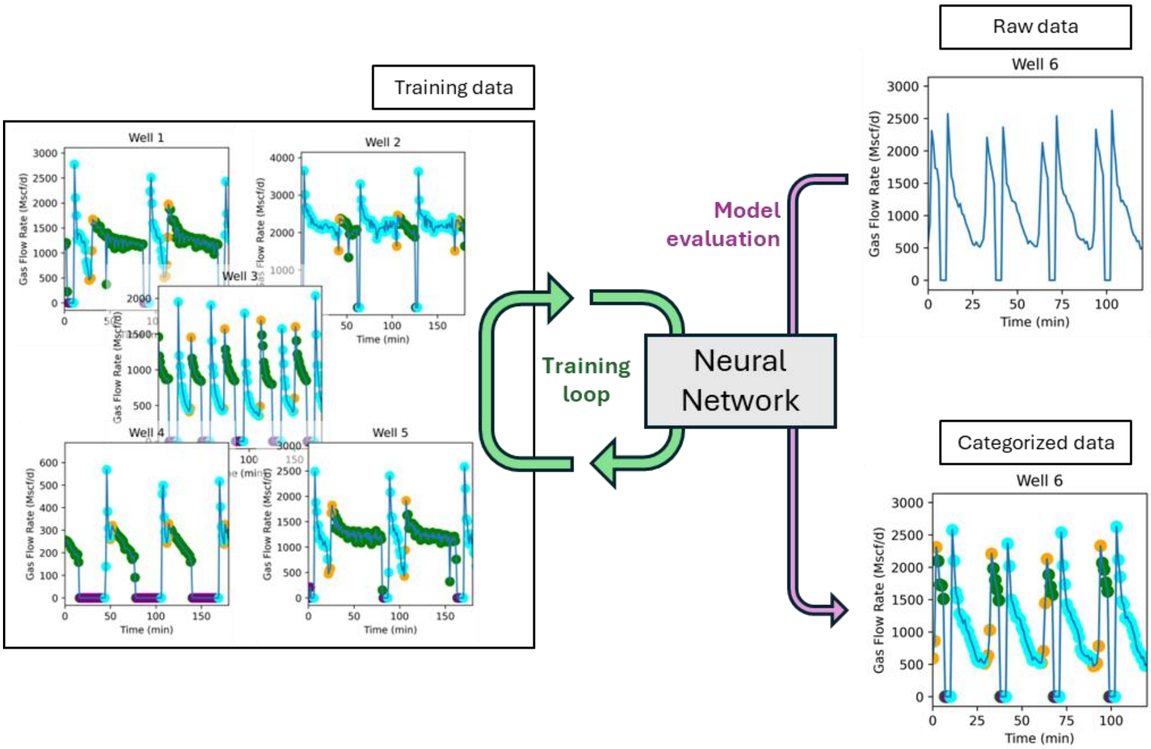

Figure 10 – Training and evaluation for the classification task

These plunger lift stages data are highly unbalanced; they were randomly selected from real-time production data. Table **3** presents the distribution of data points for each stage in each well, along with the average full-cycle time. It is observed that the wells exhibit distinct characteristics and variable cycle times.

Table 3 – Proportion of each class in the training and test data

	Proportion of each stage (%)					
Well	Upstroke	After-flow	Buildup	Liq. Discharge	Total	Avg. Cycle Time (min)
1	20	68	6	6	750	80
2	65	29	2	4	750	60
3	40	32	21	7	750	35
4	10	42	44	4	600	60
5	21	68	6	5	1730	80
6	62	18	6	14	600	35

After the training is performed, the performance assessment is done using the test data.

Figure 11 illustrates the confusion matrix for the classification task. The matrix presents the exact (true) values in the rows and predicted values in the columns. Ideally, all the counts should be in the matrix's diagonal. The color intensity indicates the amount of data classified in each category.

Table **4** presents the performance of the ANN in the prediction of each stage in terms of precision, recall, and F_1 score. The ANN performs exceptionally well, presenting a decreased recall only for the liquid discharge stage prediction. Misclassification of liquid discharge occurs primarily at the transition between the upstroke and after flow stages. This training is performed without any loss weighting, so the ANN learns the data dependence even with an unbalanced dataset.

Table 4 – Classification task performance

Label	$Precision$	$Recall$	F_1	Number of true points
Upstroke	0.96	0.99	0.98	367
After-flow	0.98	0.99	0.99	102
Buildup	0.97	1.00	0.99	34
Liquid Discharge	1.00	0.80	0.89	80

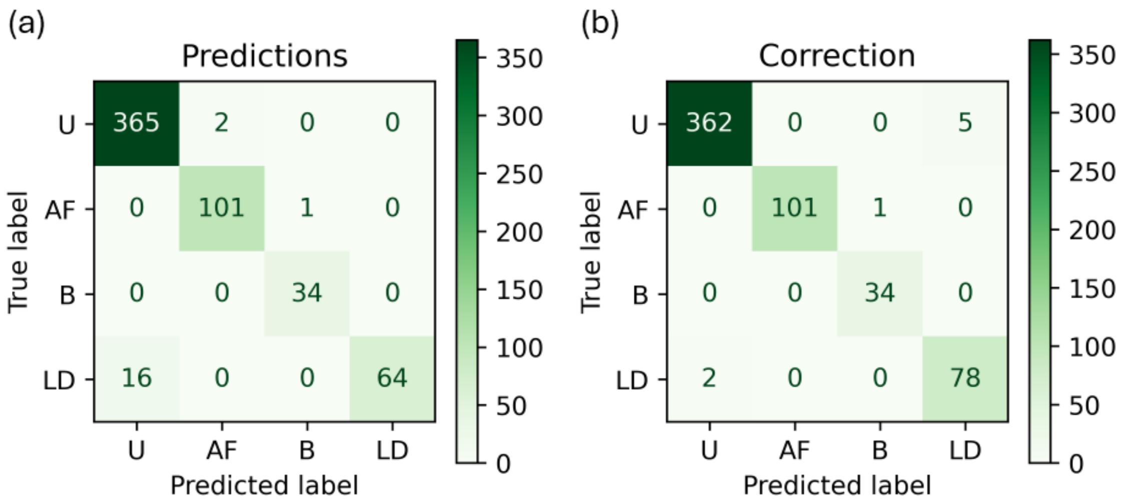

Figure 11 – Confusion matrix for the plunger stage splitting task (a) directly from the ANN and (b) after manual corrections in the liquid discharge. U – upstroke, AF – after-flow, B – build, and LD – liquid discharge.

In this work, we labeled the data assuming that each point belongs to a specific stage, whereas in fact some points represent transitions between different regions. To ensure a smooth transition between two stages, we could adopt a fuzzy-logic approach, where points in the transition region belong to both stages.

This methodology can be further enhanced by providing additional pre-defined classification labels. If a subject matter expert analyzes the operations of a well during abnormal production and generates labels for unexpected scenarios, the classification task can also predict specific events and identify abnormal conditions.

4 CONCLUSIONS

This study presents a machine-learning approach to analyzing plunger-lift data. The developed model can be used to estimate gas flow rate from pressure signals and split conventional plunger-lift signals. Both regression and classification tasks are done with minimal model changes. The regression task allows for the estimation of operational variables that lack field sensors; pressure data, such as tubing and casing pressures, are used to estimate gas flow rates for the optimization process and well production allocation. The classification task allows identification of each plunger lift cycle, enabling analysis of each cycle stage and quantifying local properties, which can be used to troubleshoot how those properties vary over time.

The tool developed in this work, once trained, is a potential candidate for real-time processing, since it is computationally cheap to execute. It means it can be deployed in production systems to generate real-time insights.

Although the model achieves good results, it relies on the data used during training. ANN models can also be developed using a physics-based approach, in which physical assumptions and constraints are incorporated. This approach could potentially extend the range of applications, e.g., PAGL, and generalize this solution.

Acknowledgments

The authors express their sincere appreciation to Tulsa University Horizontal Artificial Lift Projects (TUHWALP) for the opportunity to conduct this study and for providing the resources necessary to complete the research. They also gratefully acknowledge the contributions of the graduate research assistants, staff, technicians, industry partners, and student helpers involved in TUHWALP and Tulsa University Fluid Flow Projects (TUFFP) for their support and collaboration, which were invaluable to the success of this work.

Nomenclature

Symbols

C	Total number of classes (labels)
F_1	F1 score
$n_{lookback}$	Window size
$n_{well,lookback}$	Window size per well
n_{total}	Total number of data points

$n_{well,total}$	Total number of data points per well
N	Total number of samples
R^2	Coefficient of determination
$\bar{y}$	Expected (true) average value
y_i	Expected (true) value
$\hat{y}_i$	Estimated value
F_1	F1 score

Abreviations

ANN	Artificial neural network
FP	False positive
FN	False negative
GUI	Graphical user interface
LSTM	Long-short term memory
ML	Machine-learning
MAE	Mean absolute error
MSE	Mean squared error
MAPE	Mean average percentage error
RNN	Recurrent neural network
RMSE	Root mean squared error
TP	True positive

References

Agwu, O. E., Alatefi, S., Azim, R. A., & Alkouh, A. (2024). Applications of artificial intelligence algorithms in artificial lift systems: A critical review. *Flow Measurement and Instrumentation*, *97*. https://doi.org/10.1016/j.flowmeasinst.2024.102613

Akhiiartdinov, A., Pereyra, E., Sarica, C., & Severino, J. (2020). Data analytics application for conventional plunger lift modeling and optimization. *Society of Petroleum Engineers - SPE Artificial Lift Conference and Exhibition - Americas 2020, ALCE 2020*. https://doi.org/10.2118/201144-ms

Liang, X., Xing, Z., Gong, H., Wang, G., Lu, X., & Han, G. (2025). Intelligent fault diagnosis method for plunger lift systems integrating knowledge graph constraints. *Energy Reports*, *14*. https://doi.org/10.1016/j.egyr.2025.11.110

Viggiano, R., Micozzi, M., Chaudhry, F., Pereyra, E., & Sarica, C. (2024). A Novel Plunger Data Analysis Methodology. *Society of Petroleum Engineers - SPE Artificial Lift Conference and Exhibition - Americas, ALCE 2024*. https://doi.org/10.2118/219567-MS