A Multi-Task Learning Model for Predicting the Number of Diesel Generators on Oil Drilling Rigs

Hessam Emami
Computer Science
Southern Methodist University
Dallas TX USA
hemami@smu.edu

Jinyu Du
Statistical Science
Southern Methodist University
Dallas TX USA
jinyud@smu.edu

ABSTRACT

This paper explores five machine learning models to predict the optimal number of required online generators in drilling operations. To begin with, we implement individual, general, and multi-task models that can predict the number of online generators for the current drilling state. In addition, we build a general recursive neural network (RNN) model and a multi-task RNN model that can predict the target for thirty minutes in the future. In our multi-task architecture, we consider each drilling state as a task. Training and test datasets have 60 days of historical data collected from multiple rigs in production. We use a dataset from one rig to train and test the performance of the first three models. The RNN models' training phase uses the same dataset with three sliding windows and the testing phase uses three unseen rig datasets for model testing and performance evaluation. Our results show that the RNN models can predict the targets accurately with recall and f1-score greater than 0.95 on average.

KEYWORDS

RNN, LSTM, multi-task models, time series, recall, f1 score

1. Introduction

The drilling rig is a structure that creates holes on the earth's surface that can be used to extract oil, gas, and water. Rigs consist of multiple components including hoisting, circulation, and rotating equipment that need a source of power to run. Rig engines provide the power, and their performance has been evolving since the 19th century when steam engines were the most common type, whereas later they were replaced by diesel and gas generators [1]. These generators are costly to operate, and they emit air pollutants such as CO and CO_2. Efficient methods of operation of diesel generators will help reduce the drilling cost and protect the environment. It is worth mentioning that there are performance enhancements in drilling power supply engines. For example, the Electric AC or DC drive produced by diesel engines is one of the engine types that are more reliable and flexible compared to mechanical engines [2]. However, the domain of our research is only limited to analytic approaches to the datasets available regardless of the type of the engines and generators.

The required power for the drilling process is not a constant value and has a dependency on the drilling states and other drilling parameters. Thus, different steps in the drilling process might require a different number of generators to produce the power. Oil and drilling companies log the number of online generators and calculate if the generators are underloaded or overloaded. They make recommendations for turning off a generator or turning on additional ones based on generator loading capacities. The process is not efficient because of multiple reasons. First, rig operators cannot respond to warnings promptly. Second, engines cannot go online immediately if extra power is required during the drilling operation. If the power is not available, a blackout may happen, which stops the entire operation until the engines become available. On the other hand, keeping the underloaded engines running is costly for the company, and it produces wastes that cause pollution. Therefore, shutting down the generators or starting additional ones without careful planning can impose risks on the drilling operation and result in the engineers hesitating to act. This issue motivates us to propose using machine learning models to accurately predict the number of required generators for the oil drilling process. Successful development of the prediction models will boost the engineers' confidence of changing the number of online generators.

We propose five different models to predict the number of required generators. Specifically, we build three models that can predict the number of online generators in the current state of the drilling. We also build a general recursive neural network (RNN) model with Long Short Term Memory (LSTM) layers. Then we extend the model and fit a multi-task RNN model with LSTM layers. The multi-variant and multi-task RNN model with LSTM layers can predict the number of required generators in thirty minutes.

2. Related Work

A simulation model for offshore rigs was created to estimate the fuel consumption and the running hours. This simulator exposes some common issues during tripping and suggested solutions based on the findings. The total power in Kw, stop time, and changes in the number of the engines are simulated during the pull out of the hole (POOH) operation [3]. Their observation exposed several issues. For example, small commissioning value of the stop time parameter cannot provide reliable performance during the tripping operations. Then the simulator was run again with the modified stop time to mitigate intermittent starting and stopping of the engines. The author created annual projected savings and a drawbacks table. They claim their method can reduce the total running hour by about 1000 hours and decrease CO2 emission by 79 tons.

Time-series analytics algorithm can detect drilling fluid lost circulation incidents (LCIs) to prevent non-productive time (NPT) in drilling operations [5]. In the training phase, they use historical data obtained from sensors. In the testing phase, the deployed model is evaluated by unseen oil well data in the production. Several model architectures are employed, among which LSTM and CNN (convolutional neural network) show better F1 score performance. But the authors mainly focus on CNN because its other metrics are higher than those for LSTM. In this paper, we propose RNN models with LSTM layers and use similar techniques for data segmentation.

2.1 Our approach and problem formulation

Given the oil rig datasets, our goal is to accurately predict the number of generators required for operation oil rig operation. We built five different models. They are

1) independent classification models,

2) a general classification model,

3) multi-task classification models,

4) a general recursive neural network (RNN) model with LSTM layers, and

5) multi-task RNN models with LSTM layers.

Among the five models, models 1, 2, and 3 do not consider the time series components of the datasets and aim to predict the number of generators required for oil rig operation given other attributes. Models 4 and 5 incorporate the time series components of the datasets and aim to predict the number of generators needed in 30 minutes (t + 30 minutes) for any starting time t in 30-minute intervals. Because the target can only take on integer values, we round the regression output from models 4 and 5 to make them classification models. We demonstrate the multi-task model architecture in Figures 3 and 4.

2.2 Dataset description

Four datasets (RUUU, RTTT, RYYY, RZZZ) with relevant attributes were collected after consultation with oil and gas experts in the industry. Notice that the names for the dataset are pseudo-names with no real meaning to protect data privacy. A short description of the dataset attributes is provided in Table 1.

Table 1. A short description of the dataset attributes [4]

Tag	Description
Generator Power	The total percentage load
Generators Online	Total number of online generators
Bottom Depth	The depth of the well
Hookload	Total weight of the drill string
Standpipe Pressure	Mud pump pressure
MSE Flow	Mud pumps GPM
TDModeState	Drilling state (Drilling =1, Tripping = 2 Sliding =3)
Top Drive Speed	The speed of motor in RPM
Top Drive Torque	The Torque of the motor
Bit Depth	Bit Depth

Each dataset has the past 60 days' log information of a rig drilling operation. Data were collected every second. Because our goal is to predict the number of required generators in thirty minutes and considering multi-variant time series models have a better performance to determine the target one step ahead, we process the data and take rows at every thirty minutes interval.

The raw datasets have the 'number of online generators' column and the 'loading percentage of the generators' column. Using these two columns and the generator loading guideline in Table 2, we need to feature engineer a new column that is the ideal number of online generators in operation at any given time, which will be the target of our models.

Table 2. The generator loading guideline

Number of Online Engines	Generators are properly loaded	Generators are slightly under loaded be allowed	Generators are under loaded
1	Any load		
2	> 45.0%	40% - 45%	< 40%
3	> 63.7%	53.3 - 63.7%	< 53%
4	> 72.5%	60 - 72.5%	< 60%

2.3 Visualization of the target attribute

The data were collected in time sequence, and we aim to predict the target in (t + 30) minutes. Thus, it is important to visualize how the target changes with time and identify if there are any time-series patterns. We ordered the data by date and time, then plotted the target column (y, the total number of desired generators) against the index (x). We show two plots of target vs. sequence index for the training dataset named RUUU and one of the three test datasets named RZZZ in Figure 1 and Figure 2 respectively.

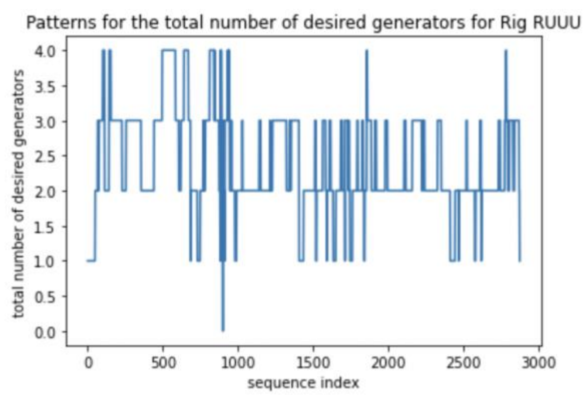

Figure 1. Patterns for the target column for Rig RUUU

2.4 Problem formulation for the multi-task RNN model

Formally, let $M = \{M_1 = 1, M_2 = 2, M_3 = 3\}$ be the set of three tasks that correspond to the three different mode states in the dataset. Given a series of samples $X^{M_i} = \{X_1^{M_i}, X_2^{M_i}, \cdots\}$ as input, which are extracted from the oil rig data using sliding windows, the multi-task RNN model generates $\hat{y}^*_{M_i}$ as the prediction output for task $M_i, i = \{1, 2, 3\}$.

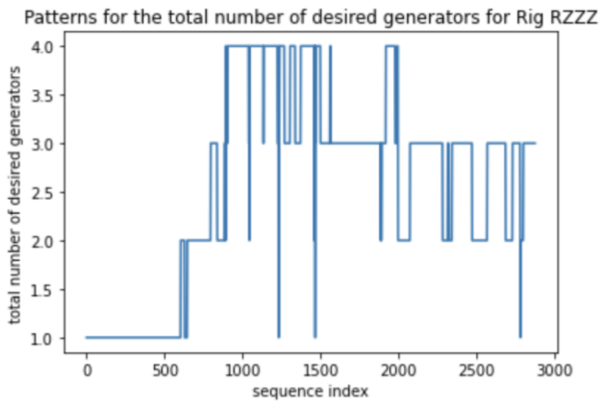

Figure 2. Patterns for the target column for Rig RZZZ

2.5 Data pre-processing

Datasets were imported and rows with missing values and columns with overlapping information were dropped. Moreover, we feature engineered a new column as the target variable. All rows were ordered based on the timestamp of data collection. Then we extracted rows at 30-minute intervals for model building.

2.6 Multi-task model architecture

Multi-tasking learning using a neural network works well for datasets that have a small amount of data for some tasks but a large amount of data for all tasks collectively. For our oil rigs dataset, we build two multi-task models with and without incorporation of the time series nature of the datasets. The multi-task model (Model 3) does not consider the time component. It has two untrainable dense layers and one trainable dense layer with softmax activation. The mode states ($M_i = 1, 2, 3$) are treated as separate tasks. Inputs to this model are the regular rows and columns as those from the processed dataset. Outputs from this model are probabilities, which are processed to class labels. Details of the multi-task model architecture are shown in Figure 3.

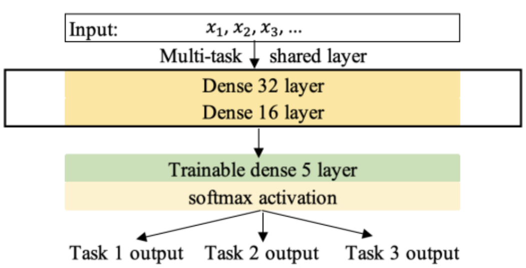

Figure 3. An overview of the multi-task model architecture

The multi-task RNN model (Model 5) considers the time component. It has two untrainable LSTM layers and one trainable dense layer with linear activation. Inputs to this model are subsamples obtained by slicing the processed dataset using sliding windows of a certain size. Outputs from the trainable dense layer are rounded to integers. The mode states ($M_i = 1, 2, 3$) are treated as separate tasks. Details of the multi-task RNN model architecture are shown in Figure 4.

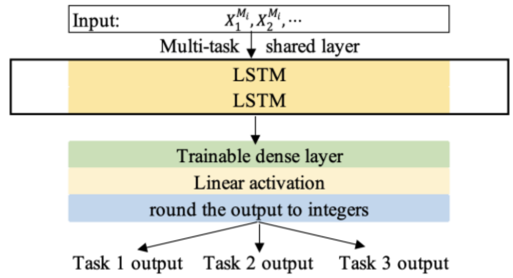

Figure 4. An overview of the multi-task RNN model architecture

2.7 Model training algorithm

The training algorithm for the multi-task model (Model 3) is shown below.

Algorithm 1. Training algorithm for the multi-task model
Input: Training samples for the three tasks
y^*, the true target values
Output: $\hat{y}^*$, the predicted target values
1 Build and train a general model (Model 2) with three dense layers,
2 Using weights from the first two dense layers of the general model as the shared base,
3 For each subtask m in M = {1, 2, 3}
4 train individual models with trainable dense layers and softmax activation
5 Return $\hat{y}^*$, the predicted target values

The training algorithm for the multi-task RNN model (Model 5) is shown below.

Algorithm 2. Training algorithm for the multi-task RNN model
Input: Sequence of training samples for the three forecast tasks
y^*, the true target values
Output: $\hat{y}^*$, the predicted target values
1 Build and train a general RNN model (Model 4) with two LSTM layers, dropout, and a dense layer with linear activation
2 Using weights from LSTM layers of the general model as the shared base,
3 For each subtask m in M = {1, 2, 3}
4 train individual models with trainable dense layers with linear activation
5 round the output from regression to intergers
6 Return $\hat{y}^*$, the predicted target values

2.8 Training

The model training process involves different loss functions for regression (mse) and classification (categorical_crossentropy) models. Regression models are transformed into classification models by rounding the output into integers as categorical labels.

3. Experiments and results

We evaluate the performance of models differently. Details of the difference in model evaluation can be found in subsection 3.1.

3.1 Experimental setup

Models 1, 2 and 3 are neural networks with fully connected dense layers and they are trained and tested (train size: test size = 6:4) using dataset from an oil rig named RUUU. Models 4 and 5 are

RNN models with LSTM layers and they are trained using a dataset from an oil rig named RUUU and tested using three other datasets from oil rigs named RTTT, RYYY and RZZZ. We use sliding windows to train and test models 4 and 5. The sliding windows move across the processed datasets and segment the time series data into groups of data according to the size of the sliding windows (30, 60, and 90) and different tasks/mode states M_i.

3.2 Model evaluation metrics

The target has imbalanced categories with some categories having fewer counts than others, thus recall and f1 scores are appropriate measures for the model performance. Recall, precision, and f1 score are defined in Equations (1), (2), and (3) respectively [6]. The number of categories of the target is 3. TP_i is the number of true positives. TN_i is the number of true negatives. FP_i is the number of false positives. FN_i is the number of false negatives, and $i = \{1, 2, 3\}$.

$$Recall = \frac{\sum_{i=1}^{3} \frac{TP_i}{TP_i + FN_i}}{3} \tag{1}$$

$$Precision = \frac{\sum_{i=1}^{3} \frac{TP_i}{TP_i + FP_i}}{3} \tag{2}$$

$$f1\ score = \frac{2 * Precision * Reall}{Precision + Reall} \tag{3}$$

3.3 Model evaluation results

Models 1, 2, and 3 are trained and tested differently from models 4 and 5. Thus we will summarize their performance metrics separately in Tables 3 and 4. Among models 1, 2, and 3, the multi-task model (model 3) performs better than the general model (model 2). The individual models (model 1) have the worst performance on test data. Results are summarized in Table 3.

Table 3. Models 1, 2 and 3 performance metrics

Models	Metrics	
	Recall	F1-score
Individual models	0.39	0.38
General model	0.66	0.59
Multi-task model	0.73	0.65

Between models 4 and 5, both the RNN + LSTM General model (model 4) and the Multi-task RNN + LSTM model (model 5) perform very well on three distinct test datasets. Compared with models 1, 2, and 3, the improved performance of models 4 and 5 may be due to the inclusion of the time components of the datasets into the models and the LSTM layers, which can preserve the time structure of data. Results are summarized in Table 4.

Table 4. Models 4 and 5 performance metrics (sliding window size is 60)

Models	Test Dataset	Metrics	
		Recall	F1-score
RNN + LSTM General model	RTTT	0.98	0.98
	RYYY	0.99	0.99
	RZZZ	0.81	0.76
Multi-task RNN + LSTM model	RTTT	0.96	0.96
	RYYY	0.99	0.99
	RZZZ	0.96	0.96

3.4 Sliding window size comparison

We want to compare the model performance under different size of the sliding windows, which is the number of rows in time series data segments for RNN model training and testing. We choose to use a sliding window with sizes 30, 60, and 90. Results are summarized in Table 5.

We see that for the RNN + LSTM general model, window sizes do not influence the average performance metrics much, though the metrics seem to decrease slightly as the window sizes get larger for dataset RZZZ. For the multi-task RNN + LSTM model, its overall performance is as good as that of the RNN + LSTM general model. Moreover, window size too small or too large might negatively affect the multi-task RNN model performance.

Table 5. Performance evaluation with different window size

Models	Test Dataset	Window size	Metrics	
			Recall	F1-score
RNN + LSTM General model	RTTT	30	0.98	0.98
		60	0.98	0.98
		90	0.98	0.98
	RYYY	30	0.99	0.99
		60	0.99	0.99
		90	0.99	0.99
	RZZZ	30	0.98	0.98
		60	0.81	0.76
		90	0.97	0.97
Multi-task RNN + LSTM model	RTTT	30	0.96	0.96
		60	0.96	0.96
		90	0.96	0.96
	RYYY	30	0.99	0.99
		60	0.99	0.99
		90	0.99	0.99
	RZZZ	30	0.97	0.97
		60	0.96	0.96
		90	0.96	0.96

4. Limitations

In industry settings, generators have different types with different specifications. The studied rigs use almost the same types of generators; hence these models could have some challenges predicting the target for other generator types that they have not seen before. If historical data is available for other types of generators, new models can be trained when current models' performance is unsatisfactory. The dependency of the models on drilling plans is an obstacle to using our models in production. Historical data specified the drilling state in the past. So, the models can predict the target quite easily. However, it may be nontrivial to determine the drilling state during the operation for the next thirty minutes.

Our models involve many hyper-parameters such as sliding window size, layers of the model architecture, loss function, and training epochs. Due to the time constrain for the project, we only studied the effect of sliding window size. It will be interesting to investigate the effects of other hyper-parameters as well.

5. Conclusion

In this paper, we successfully develop an RNN + LSTM general model and a multi-task RNN + LSTM model. They can accurately predict the total number of desired generators required for oil rig operation. We also build and compare three models which do not include the time components of the dataset. The model performance metrics recall, and f1-score show that the general and multi-task RNN models with LSTM layers are superior to the other three models under consideration. Sliding window size does not seem to impact the RNN model performance much.

ACKNOWLEDGMENTS

The authors would like to thank Helmerich & Payne, Inc, an American petroleum contract drilling company for providing the rig datasets and assistance in defining the problem. Also, we appreciate the valuable teaching and guidance of Dr. Eric Larson.

REFERENCES

[1] Bommer, P., & Baker, R. (2008). A Primer of Oilwell Drilling. University of Texas at Austin.

[2] Electric & Mechanical Drilling Rig-all drive modes explained. Sovonex. (n.d.). Retrieved May 10, 2022, from http://www.sovonex.com/drilling-equipment/api-land-drilling-rigs/electric-mechanical-drilling-rigs/#electric-drilling-rigs

[3] Čemeljić, Hrvoje, (2020).Modelling and simulation of the operation of diesel generators on offshore drilling rigs by using the Python programming language

[4] Drilling rigs, engines, & Fuels - EPCM Holdings. (n.d.). Retrieved May 10, 2022, from https://epcmholdings.com/rigs_engines_fuels/

[5] Aljubran, M., Ramasamy, J., Albassam, M., & Magana-Mora, A. (2021). Deep learning and time-series analysis for the early detection of lost circulation incidents during drilling operations. IEEE Access, 9, 76833–76846. https://doi.org/10.1109/access.2021.3082557

[6] Yang, Hong, Shanshan Gong, Yaqing Liu, Zhengkui Lin, and Yi Qu. 2020. "A Multi-task Learning Model for Daily Activity Forecast in Smart Home" Sensor 20, no. 7: 1933. https://doi.org/10.3390/s20071933