University of
South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2024
Industrial IT and Automation

Evaluation of Machine Learning Algorithms for Flow Rate Estimation in Oil and Gas Industry

Neville Aloysius D'Souza

Faculty of Technology, Natural Sciences and Maritime Sciences
Campus Porsgrunn

University of
South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis 2024
Title: *Evaluation of Machine Learning Algorithms for Flow Rate Estimation in Oil and Gas Industry*
Pages: *75*
Keywords: *Machine Learning, Data Validation, Flow Rate Measurements, Errors.*

Student: *Neville Aloysius D'Souza*
Supervisor: *Gaurav Mirlekar, Carlos Pfeiffer*
External partner: *Equinor*

Summary:
Accurate measurement of flow rate of the multiphase flow of oil, gas and water from the oil wells, is an important part of the oil and gas industry. This enables the safe operation and proper optimization of the production. Therefore much research has been dedicated to improve the accuracy of measurements. Various methods like Virtual flow metering and Multi phase flow meters are used.

With the increasing availability of process data, machine learning algorithms have been applied to create models that are beneficial to the oil and gas industry. They can be used for various parameter estimations, predictive maintenance and so on. The application of these algorithms for flow rate estimation provides a more accurate representation of the oil and gas production process.

The goal of this thesis is to use the simulator data, to create machine learning models. These models are used to predict the flow rates of oil, gas and water from the wells. Two oil wells are evaluated here. Ten machine learning algorithms are evaluated. LSTM provides the best results with MAPE of 1.96% for Well 1 and 1.56% for Well 2. In addition, the effects of noise on the models are explored. Median filter with window size of three provides good noise reduction. Finally the uncertainty of the prediction are quantified using 95% confidence intervals in XGBoost models .

The University of South-Eastern Norway accepts no responsibility for the results and conclusions presented in this report.

Preface

This thesis is submitted for the degree of Master of Science in Industrial IT & Automation at University of South-Eastern Norway, Porsgrunn.

I am deeply grateful to my thesis supervisors, Gaurav Mirlekar and Carlos Pfeiffer for their support, guidance and supervision of this thesis work.

I also want to thank Roshan Sharma for providing the Oil/Gas production model simulator.

Porsgrunn, 15th May 2024

Neville Aloysius D'Souza

Contents

List of Figures

List of Tables

Nomenclature

AI	Artificial Intelligence
ANN	Artificial Neural Network
EnKF	Ensemble Kalman Filter
FL	Fuzzy Logic
ICA	Imperialist Competitive Algorithm
IoT	Internet of Things
KDD	Knowledge Discovery in Database
KNN	K Nearest Neighbors
LSTM	Long Short Term Memory
MAPE	Mean Absolute Percentage Error
MLP	Multilayer Perceptron
MPFM	Multiphase Flow Meter
MSE	Mean Squared Error
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
RNN	Recurrent Neural Network
SVM	Support Vector Machine
SVR	Support Vector Regression

THP	Tubing Head Pressure
VFM	Virtual Flow Metering

1 Introduction

A system for producing oil and gas is usually made up of several wells connected to a flowline that transports the generated fluid from the wellheads to an inlet separator of a processing facility. If the field is submerged, a riser connects the flowline to the inlet separator. Choke valves positioned at the wellheads regulate the flowrate of the produced fluid. The oil and gas from the wells usually is in the form of a multiphase fluid. Here the multiphase fluid can be three or two phase fluids. It can be oil/water mixture, oil/gas, oil/gas/water and so on.

Accurate flow rate of each phase is necessary to optimized the production. This also enables safe operation and efficient control. To this end the fluids are usually physically separated to obtain accurate flow rate of constituent phases. Multiphase flow meters are another technology that can be deployed to increase the accuracy of measurements. With these meters there is no need to physically separate the phases of the multiphase fluid [1].

1.1 Background

In response to the cost of MPFMs and the disadvantages of relying only on them for multiphase flow measurement, VFM methods were developed. Machine learning modelling is one part of VFM that aims to address this. Research is ongoing on how to increase the accuracy and robustness of these methods. With better models, the production optimization and safety of the oil and gas production from reservoirs and wells can be improved.

The use of data driven modelling (also called machine learning modelling) in the oil and gas industry has been increasing with the availability of and storage of process data. As early as 1993 Qin and Toral [2] have used neural networks to estimate the flow rates of multiphase flow. Since then, there has been considerable research to improve the application of machine learning models in the oil and gas industry. A more detailed explanation and literature survey on use of machine learning in oil and gas industry is described in Chapter 2.

1.2 Objective

The main objective of this thesis to explore the use of machine learning algorithms in oil and gas industry. To this end, the problem can be subdivided into:

- Literature review on use of machine learning algorithm in oil and gas industry.
- Data collection and preprocessing.
- Predictions of flow rates of oil, gas using machine learning.
- Evaluate the effect of measurement noise on machine learning algorithms performance.
- Quantify the uncertainty in the predictions.

1.3 System Sketch

Fig 1.1 shows the scope and work flow of the thesis.

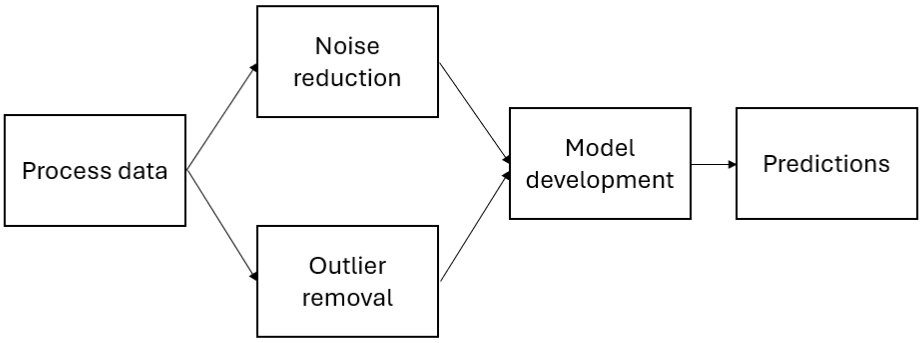


Figure 1.1: System sketch

1.4 Limitations

The limitations of the Thesis are:

- The data is based on simulator, which is based on a model.
- The models limitations are mentioned in Chapter 3.

- The sample size of 5762 is small for a machine learning problem.
- The work probably cannot be used in a production environment.

1.5 Report Structure

The report is structured as follows:

1. Chapter 1: Contains the Introduction to the thesis, the objectives, scope and limitations of the thesis.
2. Chapter 2: The literature review about the oil and gas production process is described here. The use of machine learning algorithms is explored.
3. Chapter 3: The mathematical model of the single oil well is described. Brief explanations of the machine learning algorithms that are used in the thesis is included.
4. Chapter 4: The results from the machine learning models are described
5. Chapter 5: The effects of measurement errors is explored.
6. Chapter 6: Uncertainty in the predictions is quantified.
7. Chapter 7: The results and discussions are described.
8. Chapter 8: The conclusion of the thesis is explained.

2 Literature Review

This chapter describes the oil and gas production process, and details the current literature on the use of machine learning algorithms in them for various purposes..

2.1 Oil and gas production

The production of oil and gas requires measurements of various process data. This process data is used to ensure a optimal production of oil and gas, and also ensures the safe operation of the production system. One of the most important variables that is necessary for this is the accurate measurement of oil, gas and liquid flow rates from the oil wells. Since there is multiphase flow from the oil wells, (here multiphase refers to the combination of different phases in a fluid, for oil wells it is oil mixed with gas/water or sand/mud) it is a challenge to obtain the individual flow rates of oil and gas. Traditional a separator is used as shown in Fig 2.1 to obtain an accurate flow rate of oil, gas and water. Here to measure the individual phases the multiphase mixture are separated physically with a separators. Phase flow meters are used to obtain accurate flow measurements.(add ref). This process requires a steady state flow from the given oil well. In addition to this, the other oil wells have to be shut down to avoid interference with the results. This is a costly and time consuming process.

2.1.1 Multiphase Flow metering

To solve this problem multiphase flow meters (MPFMs) can be deployed as an alternate to well testing. These are devices used to measure the individual flow rates of oil, gas, and water in a single pipeline. Multiphase flow meters provide several key advantages over traditional separation-based measurement systems. The advantages of using MPFMs are:

1. Continuous, real-time monitoring: MPFMs can provide instantaneous measurements of the individual phase flow rates, allowing for continuous monitoring of well performance without the need for periodic well testing. This enables faster decision-making and optimization of production

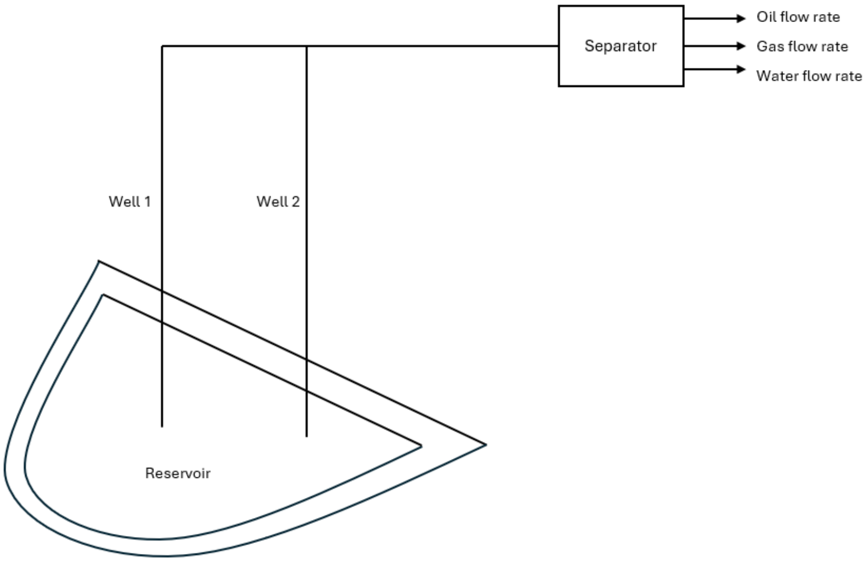


Figure 2.1: Example of sub-sea oil production

2. Reduced infrastructure: Multiphase flow meters eliminate the need for bulky and expensive test separators, reducing topside equipment and infrastructure, especially in offshore applications. This can lead to significant cost savings [3]
3. Improved reservoir management: Accurate knowledge of the individual phase flow rates allows for better reservoir characterization, production allocation, and optimization of field development. This is crucial as oil and gas fields mature and become more complex

There are different technologies that can be used for MPFM, they are briefly described here:

1. Tomography: Tomography-based MPFMs use a series of sensors to create a cross-sectional image of the flow, allowing the individual phase flow rates to be determined. This technology can handle a wide range of flow conditions but requires complex data processing [4].
2. Gamma Densitometry: Gamma densitometry MPFMs use radioactive sources to measure the density of the multiphase flow, which can then be used to calculate the individual phase flow rates. These meters tend to have high accuracy but require special handling of the radioactive materials [4][5].
3. Differential Pressure Meters: Differential pressure MPFMs measure the pressure drop across a restriction in the flow, such as a Venturi, to infer the individual phase flow rates. These are relatively simple and low-cost but can have limited accuracy, especially at high gas volume fractions [4][5].

4. Wet Gas: Wet gas MPFMs are designed to measure gas and liquid flow rates in gas-dominant flows, where the liquid content is low. They often use a combination of differential pressure and gamma densitometry techniques [6].
5. Ultrasonic Sensing: Ultrasonic MPFMs use high-frequency sound waves to measure the velocity and density of the multiphase flow, which can then be used to calculate the individual phase flow rates. This technology can be non-intrusive but may struggle with high gas volume fractions [7].
6. Coriolis: Coriolis MPFMs measure the Coriolis effect induced by the multiphase flow to determine the individual phase flow rates. They can provide high accuracy but may be limited in their ability to handle high gas volume fractions [8].

2.2 Virtual Flow Metering

While MPFMs has many advantages in measurement of multiphase flow, they are very expensive. Also, the accuracy of them can be degraded over time. In addition maintenance of these sensors are important to ensure good working conditions.

Oil and gas production systems will already have many sensors installed which monitor certain physical quantities. These can be used to develop a model which can be used to predict the flow rates. This process is called virtual flow metering.

For VFM the process data usually collected are:

- Bottomhole pressure and temperature.
- Wellhead pressure and temperature upstream of the choke.
- Wellhead pressure and temperature downstream of the choke.
- Choke opening.

VFM can be subdivided into First principles VFM and Data driven VFM. In most literature for oil and gas production, the use of machine learning is referred to as Data driven VFM. They can be further subdivided into steady state and dynamic models. For this thesis the focus is on steady state data driven VFM.

2.2.1 Data driven VFM

Data driven VFM (also called machine learning VFM) is the method where a model of the oil and gas production system is created using the available sensor data. Here in depth domain knowledge about the process is not necessary to create a model. A typical schematic for a sub-sea oil and gas production systems which used data driven VFM is shown in Fig 2.2. Broadly the steps involved are as follows:

1. Data collection
2. Data pre processing
3. Model development
4. Predictions of flow rates
5. Data reconciliation

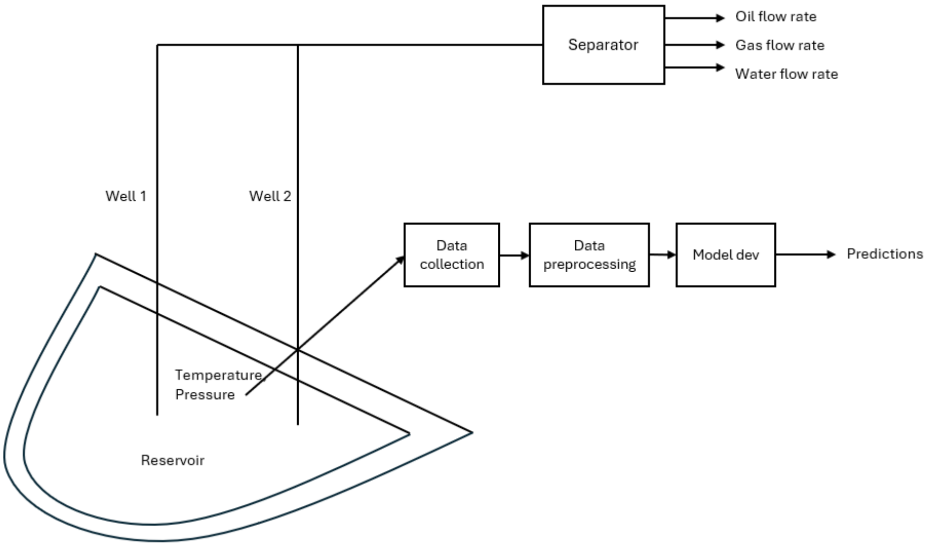


Figure 2.2: Data driven VFM

2.2.1.1 Data collection

The first step to creating a data driven model is the collection of relevant data. In Virtual Flow Metering systems, information is transmitted from wells and processing facilities and this includes sensor readings. This data may be wireless transmitted using IoT systems or through physical communication wires. It can will involve different communication protocols to ensure proper transmission of data . Historical data from the same or analogous fields may also be used as a calibrating data set for fine-tuning the model. Generally, the data collected tends to be unclean, contaminated, and may have missing values, outliers and redundant inputs.

2.2.1.2 Data pre processing

Data filtering, where the removal of noise from raw data is performed is part of this step. There exists many filters that can be deployed to clean the raw data. In addition outlier detection, correcting missing values can be included. Preprocessing can also involve data transformation, which might yield new insights about the information the data contains. Feature engineering is the common term for this technique. Numerous strategies are employed in feature engineering, such as the linear and non-linear combination of raw data, feature selection techniques, and dimensionality reduction algorithms by Principal Component Analysis (PCA).

2.2.1.3 Model development

In order to create a model, an algorithm that can map input features to output (target) variables must be developed. The mapping process, also known as training or learning, involves the algorithm modifying the parameters so that it can precisely estimate the desired variables. Depending on the algorithm being used, the parameters must be changed. The weights that connect the neurons in a neural network, for example, are the parameters. In regression trees, on the other hand, the parameter may be the tree depth. Reduce the difference between the algorithm's predicted values and the actual (measured) values to minimise a cost function, which is how training is accomplished. Mean squared error (MSE) is usually used as a cost function to solve regression problems such as Virtual Flow Metering.

To make sure the trained model will function properly on data that it hasn't encountered during training, it needs to be validated and tested on additional datasets after training. Model generalisation is the capacity to provide precise predictions on novel data. Determining the model's precise hyper-parameters to get a good match with the data is another goal of validation. The model parameters known as hyper-parameters are those that are predetermined and not learned during training. For example, the number of layers, the number of nodes in the hidden layers, the regularisation parameters, etc. are examples of hyper-parameters in neural networks. The regularisation parameters are hyper-parameters that enable the final method to be less affected by noise and outliers. In order to prevent the algorithm from over fitting the data, Arnold et al [9] gives a thorough explanation of how hyperparameters affect the model's performance, along with more specific definitions.

The method used for validation for the regression problem is described in detail in Chapter 4. In brief, there are two methods used: 1)K-fold cross validation and 2)Early stopping. These are used to obtain the best hyper-parameters for each model.

2.2.1.4 Prediction of flow rates

Once the training and validation for the model is completed, the model is tested on unseen data. New predictions from this data are noted and the effectiveness of model can be determined. For oil and gas flow rate predictions the commonly used performance metric is the Mean absolute percentage error (MAPE). With this the performance across various algorithms can be compared. It is easy to interpret and can be used across different input data scales. MAPE can be found by:

$$\text{MAPE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{|y_i - \hat{y}_i|}{\max(\varepsilon, |y_i|)} \tag{2.1}$$

2.2.1.5 Data reconciliation

An optimization algorithm adjusts the model parameters, for instance, flow rates, choke discharge coefficient, gas and water fractions, and friction and heat transfer coefficients such that the model outputs match the validated measured data being constrained to process conditions, for instance, the material balances . In Virtual Flow Metering systems, the reconciliation algorithm is often written in the constrained least-squares form The reconciliation procedure in virtual flow metering systems is frequently expressed in the constrained least squares form.

2.3 Previous work on machine learning

Much research has already been done on the use of data driven models to predict flow rates of oil and gas and other parameters in the oil industry. The next sections details some of the findings of previous research on these.

2.3.1 ANN

Artificial neural networks are one of the most popular types of machine learning algorithms that has wide use. In the oil industry the feed forward neural network or Multilayer perceptrons(MLP) are widely used. These networks are supposed to resemble the human biological neuron and the connections between them.

The earliest research on use of neural network was performed in 1993 by Qiu and Toral [2], on the extraction of stochastic features from pressure signals, to find their relation to water-cut and liquid and gas flowrates by training back-propagation neural networks with calibration samples.

Al-Qutami et al, have developed a method of using neural networks to estimate phase flow rates. Using typical observations in oil and gas production wells, a soft sensor is used this work. Because common metering facilities are used, there is limited production monitoring, which is addressed by the designed system. It serves as a backup for multiphase flow metres, lowers operating and maintenance costs, and provides an affordable way to satisfy the demands of real-time monitoring. Feed-forward neural networks are used to create the soft sensor, and K-fold cross-validation and early stopping techniques are used to control generalisation and network complexity [10].

A VFM that can estimate the gas flow rate in multiphase flow production lines is developed using radial basis function network, as proposed in this study. The created VFM has exceptional performance and generalizability, as demonstrated by the testing results obtained from real well tests. The importance of measuring choke valves and bottom holes in order to make precise forecasts is also covered in this work. The suggested VFM model offers a potentially appealing and affordable way to satisfy the demands of real-time production monitoring while lowering operating and maintenance expenses [11].

Another paper by Al-Qutami et at, proposes an ensemble learning-based VFM system for fields with shared metering infrastructure and little data generation. The suggested approach produces a variety of neural network learners by adjusting the learning trajectory, NN architecture, and training data. To choose the ideal combining technique and the best subset of learners, adaptive simulated annealing optimisation is suggested. Using real well test data, the proposed approach was assessed and shown impressive performance, with average errors for liquid and gas flow rates of 2.4% and 4.7%, respectively. Using a cumulative deviation plot, which shows that predictions are within a maximum variation of ±15%, the accuracy of the created VFM was also examined [12].

Ahmadi et al, presents a novel approach to oil rate prediction, based on an actual MPFM situation.Artificial Neural Networks (ANN), Imperialist Competitive Algorithm, and fuzzy logic-based wells are presented. The network's input variables are line temperatures and pressures, while the output variable is the rate of oil flow. In this instance, a database was constructed using a 1600 data set comprising 50 wells in one of Iran's northern Persian Gulf oil fields. ICA-ANN is a dependable substitute that doesn't cause issues for people or the environment. Additionally, a comparison of the ICA-ANN model's performance against the ANN and fuzzy models has been made. The outcomes demonstrate the efficiency, dependability, and compatibility of the ICA-ANN mode [13].

AlAjmi et al, takes an engineering examination of the production surveillance system's integration of AI data-driven models to improve welltest data validation and lower production allocation uncertainty. Data-driven oil flow rate computational models were created using artificial neural networks, fuzzy logic, and functional networks for both critical and subcritical flow circumstances. 31 distinct wells 595 production rate tests were used to train and evaluate these AI models. As a function of choke size and operating conditions, the prediction findings demonstrated a significant correlation with real field data, offering a dependable tool or methodology for estimating oil flow rate [14].

Al-Jasmi et al, uses NNs to forecast liquid rate and water cut performance in a mature reservoir with a water cut of more than 20%. The available surface and downhole, real-time production, time-dependent, and completion design data were used to train the neural network. The time-dependent data are presented as time series that can be altered by users to create different scenarios through adjustments to well operations. In addition to offering a base-case forecast, this method simulates the outcomes of modifications to control factors like pump frequency and tubing head pressure (THP). Users can model production to predict and avoid negative well pump events by varying the pumping head and frequency [15].

Alimonti et al, proposes an alternate method for analysing producing wells using fuzzy logic (FL), knowledge discovery in databases (KDD), and MFM. KDD is the automated extraction of implicit knowledge from large-scale information sources using patterns. It is possible to process distributed, ad hoc field measurements (such as MFM and downhole measurements) using artificial intelligence (AI), data integration, data cleaning, data mining, and pattern analysis. After that, FL can handle the information in terms of production optimisation and flow assurance.The reservoir and production network can also be analysed using the same methods to create an integrated production-system analysis [16].

Berneti and Shahbazian, developed a novel approach based on the feed-forward artificial neural network (ANN) and Imperialist Competitive Algorithm (ICA) to estimate the oil flow rate of the wells. The suggested method combines the global searching capability of the imperialist competitive algorithm with the local searching capability of the gradient-based back-propagation (BP) technique. The Imperialist Competitive Algorithm is employed to determine the neural network's starting weights. Using a data set of 31 wells in one of Iran's northern Persian Gulf oil fields, the ICA-ANN is used to estimate the wells' oil flow rates. The effectiveness of the ICA-ANN is shown by the comparison of its performance with that of ANN [17].

Hasanvand and Berneti, created a new approach to well oil rate prediction using artificial neural networks, based on a real-world example including multiphase flow metres. The network's input variables are line temperatures and pressures, while the output variable is the rate of oil flow. In this instance, a 600 data set comprising 31 wells in an Iranian oil field near the northern Persian Gulf was used. The data was gathered for each well over the course of three months, from December 2002 to November 2010 [18].

García et al, present a method that uses information from sensors, well testing, and simulations to measure each well's oil production is based on a neural network and online correlation logic. The approach for data selection, sensor validation analysis, modelling, online implementation, and outcome quality control is described in the study. The primary advantage of this implementation has been the ability to detect production deviations above or below well potential promptly, as well as the ability to determine and modify the elements influencing these deviations [19].

Denney et al, demonstrated NeuralFlow's effectiveness in the field over a nine-month period without the need for recalibration. Additionally, there are instances of data-driven VFM systems being used in fields when they were designed with a particular field scenario in mind [20].

Omrani et al, have looked into using artificial neural networks, totally data-driven approach for virtual flow metering and real-time back-allocation in oil and gas production wells. Simulated and real-world data from multiple gas wells were used to evaluate the suggested methodology. Two different type of artificial neural networks (ANNs) were tested on simulated and field data to assess the accuracy of estimations for steady-state, transients and dynamics in productions due to cyclic operation (shut-ins and restart). The outcomes demonstrated that ANN could correctly predict the multiphase flow rates in both field and simulated data [21].

Olivares et al, created a new workflow using a set of predictive proxy models, that combines high-frequency and sporadic data with artificial intelligence methods like neural networks and nodal analysis to enable engineers to process and comprehend production behaviour from the vast amounts of data collected in accordance with the system under study or evaluation. In addition to improving the accuracy of hydrocarbon accounting from the pumping process to the marine terminal and implementing an early detection system for anomalies that is published on the Internet for sharing with the entire asset management, this workflow allows validation of field well test data, thereby reducing uncertainties in well production allocation [22].

Shaban and Tavoularis, developed a method using Principal Component Analysis (PCA) to preprocess the raw data, while independent component analysis was used to identify dependent features. Phase flow rates were obtained as the output of multi-layer back-propagation neural networks, which were fed the extracted characteristics as inputs. In order to estimate the flow rates of both phases in an air-water flow in a vertical pipe with a diameter of 32.5 mm and in the pressure range of 100 to 140 kPa, the current method was utilised to calibrate a differential pressure sensor. Direct flow rate measurements and the current method's predictions agreed fairly well [23].

2.3.2 LSTM

Recurrent neural network (RNN) have been more recently used in the oil industry, here the Long Short-Term Memory (LSTM), a type of model is popularly used. In this type of models, the long-term dependencies in time series data can be handled.

Andrianov showed that a recurrent neural network with Long Short-Term Memory (LSTM) may be used to forecast the rates for a series of future time instants in addition to reliably estimating the multiphase rates at the current time (i.e., functioning as a virtual flow metre). The outcomes of hydrodynamical modelling and LSTM forecasts compare favourably for a synthetic severe slugging event. LSTM results for a variable rate well test's synthetic noisy dataset demonstrate that the model can also accurately predict multiphase rates for a system with fluctuating flow patterns [24].

Loh et al, customised a LSTM model for predicting gas flow rates in mature gas wells, accounting for input parameter uncertainty. Furthermore, the Ensemble Kalman Filter (EnKF) is utilised to update the flow rate predictions based on fresh observations in order to improve the prediction's accuracy and robustness owing to changes in the system over time. The new method was evaluated using data from two mature gas production wells that have extremely dynamic production and salt deposition issues [25].

Sun et al, proposed a new method for modelling time-series-related issues (such production forecasting) utilising RNN-based sequence-to-sequence models was presented in this paper. For assets with or without reliable operation history data, the established data-driven strategy increases the efficiency and accuracy of the history matching and forecasting processes. Furthermore, open-source libraries were used to construct the case studies and methods in this article. These libraries might easily be integrated into proprietary or in-house software [26].

2.3.3 Other methods

Xu et al, used a Support Vector Machine technique, which performed better than a Neural Network approach, to estimate the flowrates based on the Venturi pressure difference measurements from the experiments [27].

Zang et al, contrasts a back-propagation neural network (ANN), a classification approach (random forest), and a very basic regression method (MLR). Step rate well testing from three different wells provide real-time data that is used to train all three systems. All three trained models are put through a blind test to compare how predictable each approach is. All methods gave good results with low errors [28].

Gerrard and Taylor, discussed how Shell employs data-driven VFM software called FieldWare Production Universe (FW PU) in its fields all over the world. The Smart Fields programme provided the inspiration for this software's development. Its goal was to optimise field production in Shell's fields by utilising smart machinery, technologies, and procedures [29][30].

Grimstad et al, used B-spline surrogate models to estimate flowrate. They employed Prosper's pressure drop, choke, and inflow performance models to get the data for the method, and they fitted the outcomes using a cubic spline interpolation tool [31].

Bikmukhametov and J¨aschke, used regression trees and the gradient boosting method as a VFM system to forecast oil flowrates in various field development scenarios. They examined the situations in which VFM is employed as a stand-alone system and as a backup for an MPFM. The data produced by the OLGA programme was used to train the algorithm. As demonstrated by the results, the algorithm has a fair chance of predicting multiphase flowrates even with relatively modest datasets from the MPFM measurements and well testing. To increase the accuracy of flowrate prediction, the technique can also be integrated with neural networks inside ensembles [32].

Bello et al, describes a novel method for creating a virtual flow metre for production wells using well configuration data, available time series field data, and hybrid intelligent modelling technology. Real-world field data is contrasted with the simulation results from the hybrid intelligent virtual flow rate metre. Future performance of currently operational wells is predicted using the proven model. To find their influence on the novel approachs predicted accuracy, different factors are tested [33].

Al-Qutami et al, suggests using a heterogeneous ensemble of regression trees and neural networks to create a VFM model that uses parameter perturbation and bootstrapping to create variability among learners. Simulated annealing optimisation is used to prune the ensemble in order to better guarantee accuracy and lower ensemble complexity. Eight production wells' worth of well-test data spanning five years are used to validate the suggested VFM model. The performance of the results is better than that of homogeneous ensemble approaches [34].

3 Modelling

This chapter describes the mathematical model of the gas lifted oil well system. It also includes a brief explanation of the machine learning algorithms used for creating models for flow rate predictions.

3.1 Description of the oil well

The modelling of the oil well is based on the work of Janatian et al. A single gas lifted oil well is shown in Figure 3.1. Through the gas lift choke valve, high-pressurized natural gas is continually injected into the wells annulus in this system, which is mostly utilised to extract lighter crude oils. The injected gas finds its way into tubing at some points located at proper depths and mixes with the multiphase fluid from the reservoir. As a result of this mixing, the density of the fluid in the tubing will be reduced, which means that the flowing pressure losses in the tubing reduce. Consequently, the reservoir pressure will be able to overcome the flowing resistance in the well and push the reservoir fluid to the surface. Each well has its own inflow characteristics [35].

In addition there are some assumptions made to simplify the modelling process as described in the paper by Janatian et al [36].

- Pressure of the reservoir is constant.
- Density of liquid is constant and not a function of pressure and temperature.
- Loss of pressure heads due to friction in the pipes has been neglected.
- Temperature of gas and oil is constant at all points in the pipelines.
- All phases of multiphase fluid in the tubing are evenly distributed (no slugging).
- Flashing does not occur in any section of the oil well.

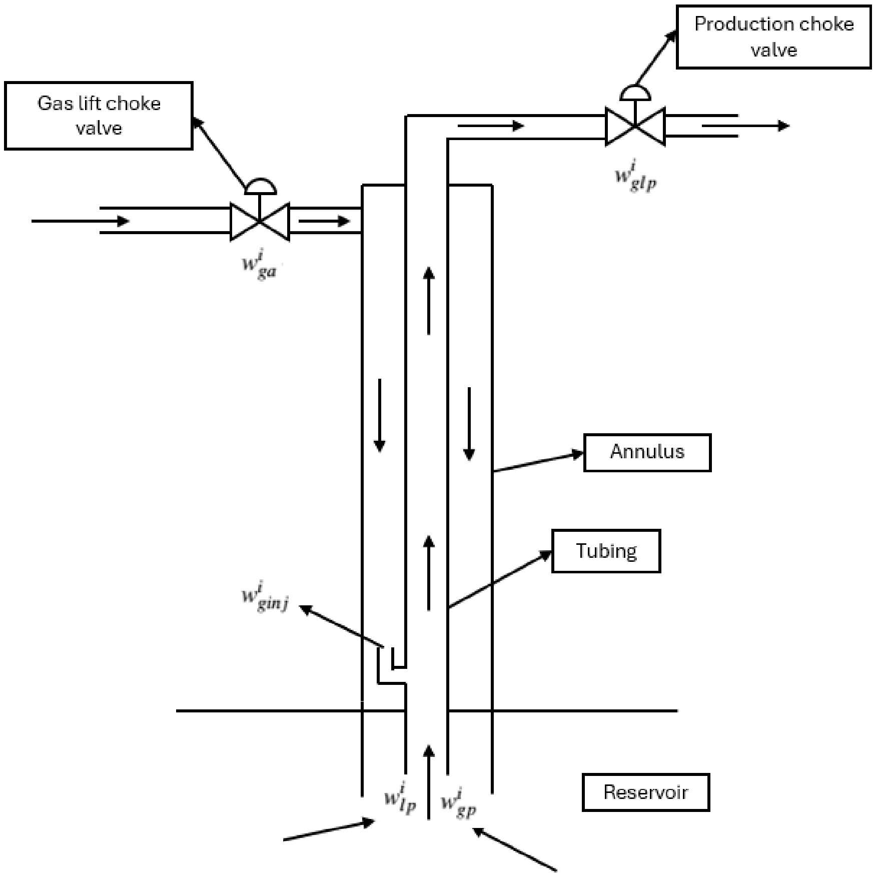


Figure 3.1: Simplified single oil well

3.1.1 Mathematical model of gas lifted oil well

From the work of Janatian et al [36], the mass balance differential equations governing the mass flows is given by: (i superscript is for the well number)

$$\dot{m}^i_{ga} = w^i_{ga} - w^i_{ginj} \tag{3.1}$$

$$\dot{m}^i_{gt} = w^i_{ginj} + w^i_{gr} + w^i_{gp} \tag{3.2}$$

$$\dot{m}^i_{lt} = w^i_{lr} - w^i_{lp} \tag{3.3}$$

Where,
m^i_{ga} - mass of lift gas in annulus,
m^i_{gt} - mass of gas in the tubing above the injection point,
m^i_{lt} - mass of liquid in the tubing above the injection point,
w^i_{ga} - mass flow flow rate of injected lift gas into ith well from the gas lift choke valve,

w^i_{ginj} - mass flow rate of gas injection from the annulus into the tubing,
w^i_{gp} - mass flow rate of gas phase through production choke valve,
w^i_{lp} - mass flow rate of liquid phase through production choke valve,
w^i_{gr} - mass flow rate of gas from reservoir into well,
w^i_{lr} - mass flow rate of liquid from reservoir into well.

The flow equations are as follows:

$$\begin{aligned}
w^i_{ging} &= K^i Y^i_2 \sqrt{\rho^i_{ga} max(P^i_{ainj} - P^i_{tinj}, 0)} && (3.4)\\
w^i_{gp} &= \frac{m^i_{gt}}{m^i_{gt} + m^i_{lt}} w^i_{glp} && (3.5)\\
w^i_{lp} &= \frac{m^i_{lt}}{m^i_{gt} + m^i_{lt}} w^i_{glp} && (3.6)\\
w^i_{lp} &= PI^i max(P_r - P^i_{wf}) && (3.7)\\
w^i_{gp} &= GOR^i w^i_{lp} && (3.8)\\
w^i_{glp} &= C_v(u^i_2) Y^i_3 \sqrt{\rho^i_m max(P^i_{wh} - P_m, 0)} && (3.9)\\
w^i_{op} &= \frac{\rho_o}{\rho_w}(1 - WC^i) w^i_{lp} && (3.10)
\end{aligned}$$

Where,
K^i - gas injection valve constant,
Y^i_2 - gas expandability factor for the gas that passes through the gas injection valve,
ρ^i_{ga} - average density of gas in the annulus,
P^i_{ainj} - pressure upstream of the gas injection valve in the annulus,
P^i_{tinj} - pressure downstream of the gas injection valve in the tubing,
w^i_{glp} - total mass flow rate of all phases from the production choke valve,
PI^i - productivity index,
P_r - reservoir pressure,
P^i_{wf} - bottomhole pressure,
GOR - gas to oil ratio,
$C_v(u^i_2)$ - production choke valve characteristics as it is opening,
Y^i_3 - gas expandability factor for the gas that passes through the production choke valve,
ρ^i_m - density of multiphase mixture in tubing above injection point,
P^i_{wh} - wellhead pressure,
P_m - gathering manifold pressure,
w^i_{op} - oil compartment of the liquid produced from production choke valve w^i_{lp},
ρ_o - density of oil,
ρ_w - density of water,
WC^i - water cut.

The pressure equations are given by:

$$P_a^i = \frac{Zm_{ga}^i RT_a^i}{MA_a^i L_{a\_tl}^i} \tag{3.11}$$

$$P_{ainj}^i = P_a^i + \frac{m_{ga}^i}{A_a^i L_{a\_tl}^i} gL_{a\_vl}^i \tag{3.12}$$

$$P_{tinj}^i = \frac{Zm_{gt}^i RT_t^i}{MV_G^i} + \frac{\rho_m^i gL_{t\_vl}^i}{2} \tag{3.13}$$

$$P_{wh}^i = \frac{Zm_{gt}^i RT_t^i}{MV_G^i} - \frac{\rho_m^i gL_{t\_vl}^i}{2} \tag{3.14}$$

$$P_{wf}^i = P_{tinj}^i + \rho_l^i gL_{r\_vl}^i \tag{3.15}$$

Where,
P_a^i - pressure of gas in annulus downstream of the gas lift choke valve,
Z - gas compressibility factor,
R - universal gas constant,
T_a^i - temperature in tubing,
M - molar mass,
A_a^i - annulus cross-section area,
$L_{a\_tl}^i$ - total length of annulus,
P_{ainj}^i - pressure upstream of the gas injection valve in the annulus,
g - acceleration due to gravity,
$L_{a\_vl}^i$ - vertical length of annulus,
P_{tinj}^i - pressure downstream of the gas injection valve in the tubing,
T_t^i - temperature in tubing,
ρ_m^i - density of multiphase mixture in tubing above injection point,
$L_{t\_vl}^i$ - vertical length of tubing above injection point,
V_G^i - volume of the gas in the tubing above the gas injection point,
ρ_l^i - average density of the liquid phase,
$L_{r\_vl}^i$ - vertical length of tubing below injection point.

The densities and remaining equations are:

$$\rho^i_{ga} = \frac{M(P^i_a + P^i_{ainj})}{2ZRT^i_a} \tag{3.16}$$

$$\rho^i_l = \rho_w WC^i + \rho_o(1 - WC^i) \tag{3.17}$$

$$\rho^i_m = \frac{m^i_{gt} + m^i_{lt}}{A^i_t L^i_{t\_vl}} \tag{3.18}$$

$$Y^i_2 = 1 - \alpha_Y \frac{P^i_{ainj} - P^i_{tinj}}{max(P^i_{ainj}, P^m_{ainj}in)} \tag{3.19}$$

$$Y^i_3 = 1 - \alpha_Y \frac{P^i_{wh} - P_m}{max(P^i_{wh}, P^m_{wh}in)} \tag{3.20}$$

$$V^i_G = A^i_t L^i_{t\_vl} - \frac{m^i_{lt}}{\rho^i_l} \tag{3.21}$$

Where,
ρ^i_{ga} - average density of gas in the annulus,
A^i_t - annulus cross-section area,
Y^i_2 - gas expandability factor through the gas injection valve.

Table 3.1.1 shows the Well 1 and Well 2 parameters, using these and the equations described an open loop simulation of the systems is created in Matlab. With this simulation the data for creating machine learning models can be created.

Table 3.1: Parameters for Well 1 and Well 2

Parameter	Well 1	Well 2	Unit
K	68.43	67.82	$[\frac{\sqrt{\frac{kgm^2}{bar}}}{hr}]$
PI(1e+4)	2.51	1.63	$[\frac{kg/hr}{bar}]$
GOR	0.05	0.07	-
WC	0.20	0.10	-
L_{a_tl}/L_{t_tl}	2758	2559	$[m]$
L_{a_vl}/L_{t_vl}	2271	2344	$[m]$
A_a	0.0174	0.0174	$[m^2]$
A_t	0.0194	0.0194	$[m^2]$
L_{r_vl}	114	67	$[m]$

3.2 Machine learning algorithms

There are many machine learning algorithms that have been developed for regression tasks. The algorithms that are used in this thesis for predicting the flow rates are briefly described here [37].

3.2.1 Multivariate Linear Regression

Linear Regression is the simplest machine learning algorithm. It makes a prediction by simply computing a weighted sum of the input features, plus a constant called the bias term, as shown in equation 3.22

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \tag{3.22}$$

Where,
$\hat{y}$ - predicted value,
n - number of features,
x_i - ith feature value,
θ_j - jth model parameter,
θ_0 - bias term.
This can be modified to output multiple $\hat{y}$ values. Multivariate linear regression is a statistical technique that models the linear relationship between multiple independent variables and a single dependent variable. It extends simple linear regression by allowing for the inclusion of more than one predictor variable. The goal is to find the linear equation that best predicts the dependent variable based on the independent variables [38].

3.2.2 k-Nearest Neighbors Regression

he k-nearest neighbors (kNN) algorithm is a non-parametric, supervised learning method used for classification and regression tasks. It works by identifying the k closest training examples to a given data point and assigning a class or value based on the majority vote or average of those neighbors [39][40]. The key steps in KNN are:

1. Determine the distance metric to measure proximity between data points, such as Euclidean or Manhattan distance.
2. Select the value of k, which represents the number of nearest neighbors to consider.
3. For a new data point, identify the k closest training examples and assign the class or value based on those neighbors.

kNN is a versatile algorithm that can handle both numerical and categorical data without making assumptions about the underlying data distribution. It is commonly used in applications like recommendation systems, pattern recognition, and anomaly detection. The choice of k is important, as lower values can lead to overfitting while higher values may cause underfitting [41].

3.2.3 Support Vector Regression

Support Vector Regression (SVR) is a nonparametric technique that uses kernel functions to estimate a function from a set of training data. The goal is to find a function f(x) that deviates from the target values y by no more than $\boldsymbol{\varepsilon}$, while being as flat as possible. This is achieved by solving a convex optimization problem that minimizes the norm of w, subject to the constraint that the regression errors are within $\boldsymbol{\varepsilon}$ [42].

SVR can handle high-dimensional data and nonlinear relationships by implicitly mapping the input data into a higher-dimensional feature space using kernel functions. Unlike other regression models that try to minimize the error between the real and predicted values, SVR tries to fit the best line within a threshold value (distance between hyperplane and boundary line). The data points on either side of the hyperplane that are closest to the hyperplane are called Support Vectors, which are used to predict the output [43].

SVR has several advantages, such as being robust to outliers, having excellent generalization capability, and easy implementation. However, it is not suitable for large datasets, and its performance may degrade when the number of features exceeds the number of training samples [44].

3.2.4 Decision Tree Regression

A decision tree algorithm is a supervised machine learning technique used for both classification and regression tasks. It constructs a tree-like model of decisions based on the data's attributes. The process starts at the root node and splits the data into subsets using the most significant attribute based on selection criteria like information gain or Gini impurity [45].

Each internal node of the tree represents a ”test” on an attribute, each branch represents the outcome of that test, and each leaf node represents a class label or a continuous outcome. The paths from root to leaf represent classification rules or regression paths. Decision trees handle both numerical and categorical data and are intuitive, as they mimic human decision-making processes. They are particularly useful in scenarios where relationships between parameters are non-linear or complex [46].

However, decision trees can suffer from overfitting, especially with very complex trees. Techniques such as pruning are used to remove parts of the tree that do not provide additional power in order to reduce overfitting and improve the model's generalizability. Decision trees are foundational elements in more complex algorithms like Random Forests and boosting methods, enhancing their stability and accuracy [47].

3.2.5 Gradient Boosting Regression

Gradient Boosting Regression is a powerful machine learning algorithm that combines multiple weak models to form a strong learner. It is particularly effective for regression problems where the goal is to predict continuous values. The algorithm works by iteratively training decision trees on the residuals of previous predictions, which are the differences between the actual and predicted values. Each tree is trained to minimize the error of the previous tree, and the learning rate determines the contribution of each tree to the final prediction [48].

The process begins with an initial guess, typically the mean of the target variable. Then, at each iteration, a new tree is trained to predict the residuals from the previous tree. The residuals are the differences between the actual and predicted values. The new tree is added to the previous trees, and the process is repeated until a stopping criterion is reached, such as a maximum number of trees or a minimum improvement in the model's performance [49].

The final prediction is the sum of the predictions from all the trees, weighted by their learning rates. This approach allows the algorithm to capture complex relationships between the input variables and the target variable, making it highly effective for regression problems [50].

3.2.6 XGBoost Regression

XGBoost is a powerful algorithm for building supervised regression models. It was developed by Chen and Guestrin [51]. It is an implementation of gradient boosting that is designed to be highly efficient and scalable. The algorithm is particularly effective for regression problems where the goal is to predict continuous or real values. XGBoost is based on the concept of ensemble learning, where multiple base learners are trained and combined to produce a single prediction.

The core components of XGBoost for regression include the objective function, base learners, and regularization. The objective function is responsible for defining the loss function and the regularization term. The base learners are the individual models that are trained and combined to produce the final prediction. Regularization is used to prevent overfitting by penalizing complex models [52].

XGBoost uses a unique approach to building regression trees. Each tree starts with a single leaf and all residuals go into that leaf. The algorithm then calculates a similarity score for this leaf based on the residuals. The similarity score is used to determine how to split the data into two groups. This process is repeated recursively until a stopping criterion is reached. XGBoost is widely used in various applications due to its high accuracy and efficiency. It is particularly effective for large datasets and can be easily integrated with other tools and packages such as scikit-learn and Apache Spark [53].

3.2.7 PC Regression

Principal component regression (PCR) is a regression analysis technique that combines principal component analysis (PCA) and linear regression. The key idea behind PCR is to first perform PCA on the predictor variables to obtain a set of uncorrelated principal components, and then use these principal components as the new predictors in a linear regression model, instead of the original variables [54].

The main advantages of PCR are that it can help address issues like multicollinearity and high dimensionality in the predictor variables. By using a subset of the principal components, PCR can reduce the number of predictors in the regression model, which can improve the model's interpretability and generalization performance. However, PCR does not perform feature selection, as each principal component is a linear combination of all the original predictors [55].

While PCR can be a useful technique, it has some limitations. It relies on the assumption that the directions of maximum variance in the predictor variables are also the most predictive of the response variable, which is not always the case. Additionally, PCR can result in information loss, as it discards some of the principal components during the regression step.

3.2.8 PLS Regression

PLS regression is a powerful statistical technique that is particularly useful for analyzing high-dimensional data with many predictor variables.The key idea behind PLS regression is to find a set of latent components (linear combinations of the original predictors) that maximize the covariance between the predictors and the response variable. Unlike traditional linear regression, PLS does not require the predictors to be orthogonal or the number of predictors to be less than the number of observations [56].

PLS regression works by iteratively extracting latent components that explain as much of the covariance between the predictors and response as possible. The resulting PLS model provides both dimension reduction and regression coefficients, allowing for accurate prediction of the response variable from the original high-dimensional predictors [57].

PLS regression has several advantages over other regression methods, including its ability to handle multicollinearity, its robustness to noise, and its suitability for datasets with more predictors than observations. As a result, PLS is a widely used technique in fields such as chemometrics, bioinformatics, and marketing research.

3.2.9 MLP Neural network

A Multilayer Perceptron (MLP) is a type of artificial neural network that consists of multiple layers of interconnected nodes, or neurons. Unlike a single-layer perceptron, which can only learn linearly separable patterns, an MLP can learn more complex, non-linear relationships in data [58].

The key components of an MLP are the input layer, one or more hidden layers, and an output layer. The input layer receives the data, which is then passed through the hidden layers, where the network learns to represent the data in a more abstract way. Each hidden layer applies a non-linear activation function to the weighted sum of its inputs, allowing the network to learn complex patterns. The final output layer produces the predicted result.MLPs are trained using a supervised learning algorithm, typically back propagation, which adjusts the weights of the connections between neurons to minimize the error between the predicted and actual outputs. This iterative process allows the MLP to learn the underlying structure of the data and make accurate predictions on new, unseen data [59].

3.2.10 LSTM

LSTMs (Long Short-Term Memory) are a type of recurrent neural network designed to address the vanishing gradient problem in traditional RNNs. The key feature of LSTMs is their memory cell, which can selectively retain or discard information as it flows through the network [60].

LSTMs have three gates that control the flow of information: the input gate, forget gate, and output gate. The input gate decides what new information from the current input and previous output should be added to the memory cell. The forget gate determines what information from the previous memory cell should be retained or forgotten. The output gate controls what information from the current memory cell and input should be used to produce the output. This gating mechanism allows LSTMs to learn long-term dependencies in sequential data, making them well-suited for tasks like language modeling, machine translation, speech recognition, and time series forecasting. LSTMs have been widely adopted and have significantly advanced the state-of-the-art in many sequence-to-sequence learning problems [61].

4 Predictions

This chapter describes the implementation of the machine learning algorithms to create models. These models are used to predict the flow rates of oil, gas and water.

4.1 Setup

The data to be used for creating machine learning models are generated using Matlab. The equations described in Chapter 4 are used to create an open loop simulator in Matlab. The gas injection is varied from 10% to 100% in each well. Ten datasets are obtained for each well. To simplify the modelling process, the mean of the datasets are used. In total one model is created for each well. A sample time of 1 minute is used to generate the simulation data.

4.2 Data splitting

For machine learning it is very important to split the data into train and test sets. This ensures the model is not over fitted and the performance of the model can be evaluated on the test set. Here the data is split into 70% train and 30% test data.

4.3 Data visualization

The test data is not used for any analysis or visualization, this is to reduce any human bias from contaminating the results. The co-relations between the features can also be found. This calculated using the standard correlation coefficient (Pearson's r) between pairs of variables.

4.3.1 Well 1

The input variables from the dataset for well 1 and well 2 are W_{ga}, the mass flow rate of injected lift gas into ith well from the gas lift choke valve, P_{wh}, the wellhead pressure and P_{wf}, bottomhole pressure.

The target variables are W_{op}, mass flow rate of liquid phase through production choke valve, W_{lp}, the mass flow rate of liquid phase through production choke valve and W_{gp}, the mass flow rate of gas phase through production choke valve.

The correlation graphs between the output variables and the input features for well 1 is shown in Fig 4.2. It can be observed from the last column that P_{wf} has positive correlation with all the three target variables. A perfect positive correlation would be a 45 degree straight line, but here it can be seen that the non linearities are also present.P_{wf} more non linearity with the target variables, it have as a slightly negative correlation with them.(W_{ga} shows a more discrete effect on the target variables. This is excepted as mass flow injected into the wells cause the change in oil, gas and liquid production).

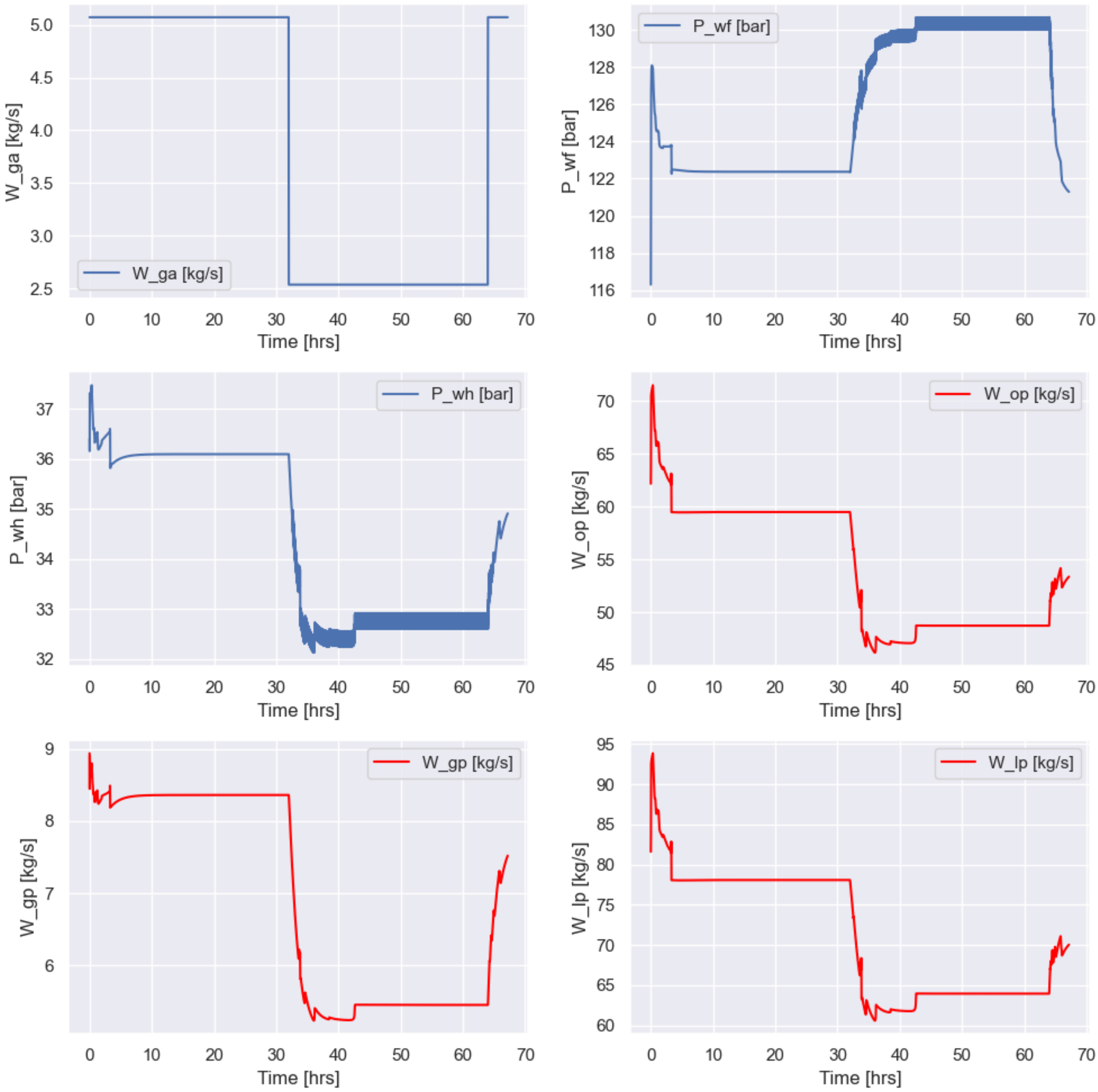


Figure 4.1: Plots of input features in red, output variables in blue (Well 1)

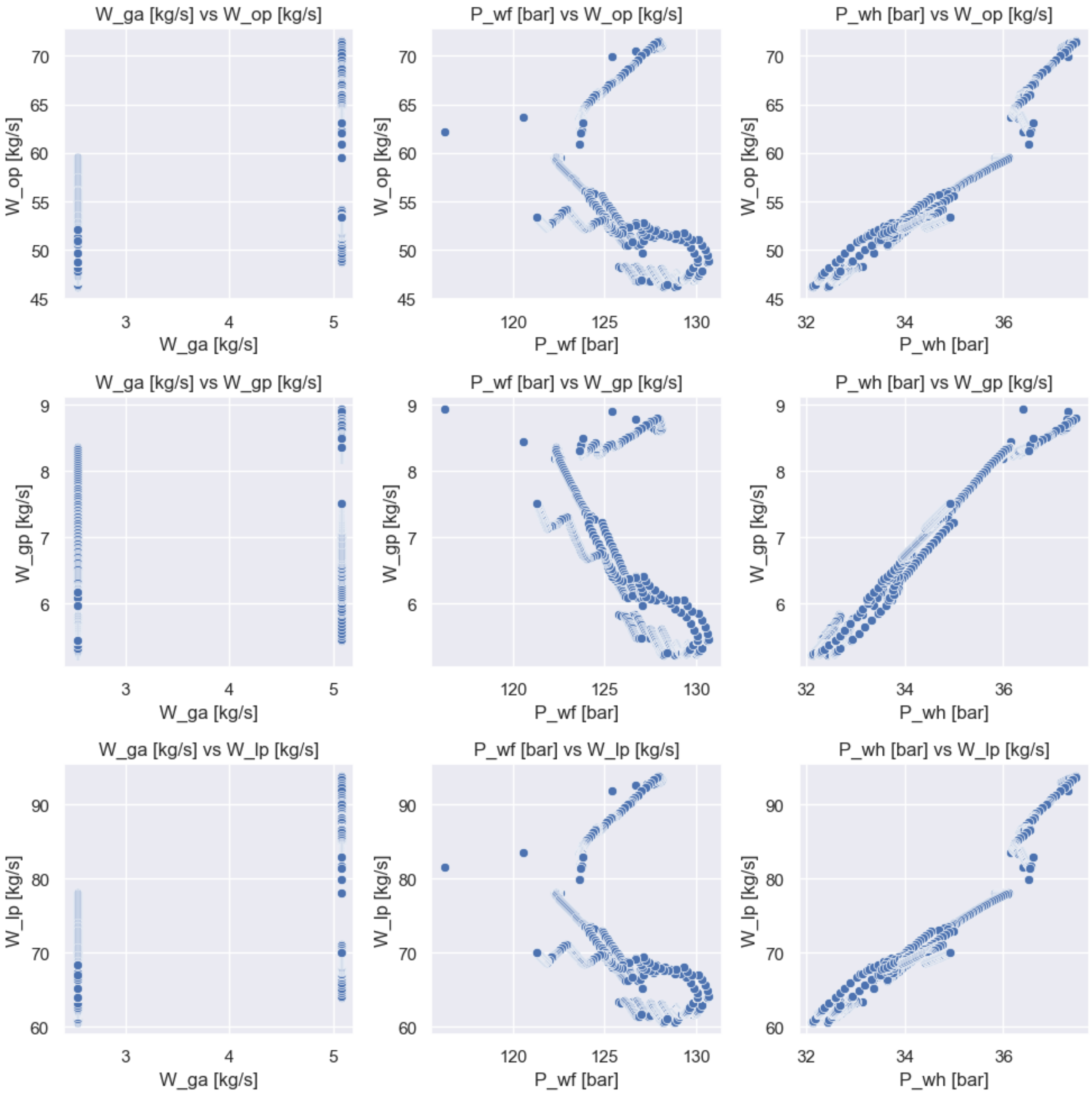


Figure 4.2: Plots of correlation graphs (Well 1)

4.3.2 Well 2

The input and target variables for well 2 are the same as described for well 1. The correlation graphs between the output variables and the input features for well 1 is shown in Fig 4.4. Here the observations are similar to the Well 1.

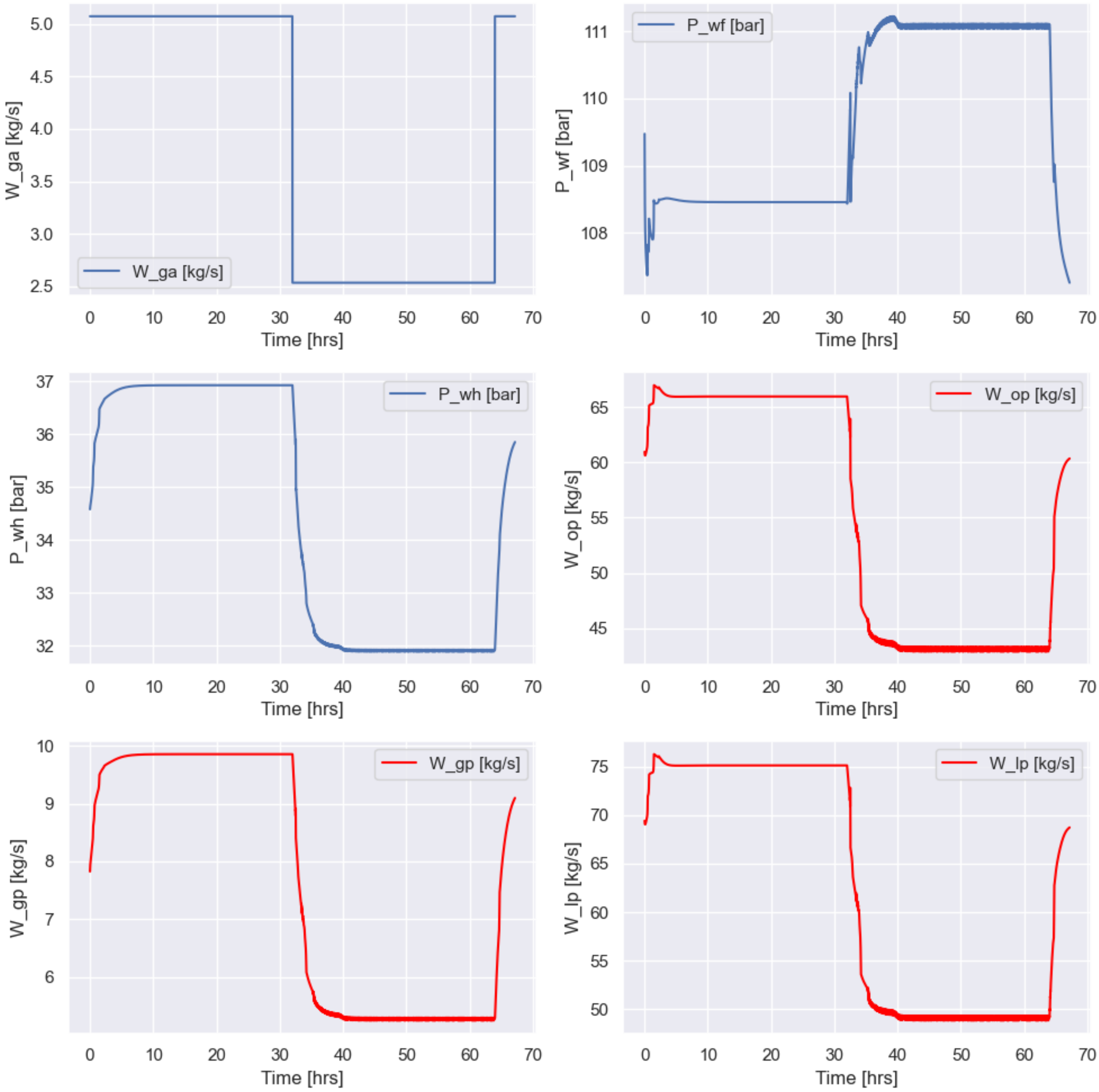


Figure 4.3: Plots of input features in red, output variables in blue (Well 2)

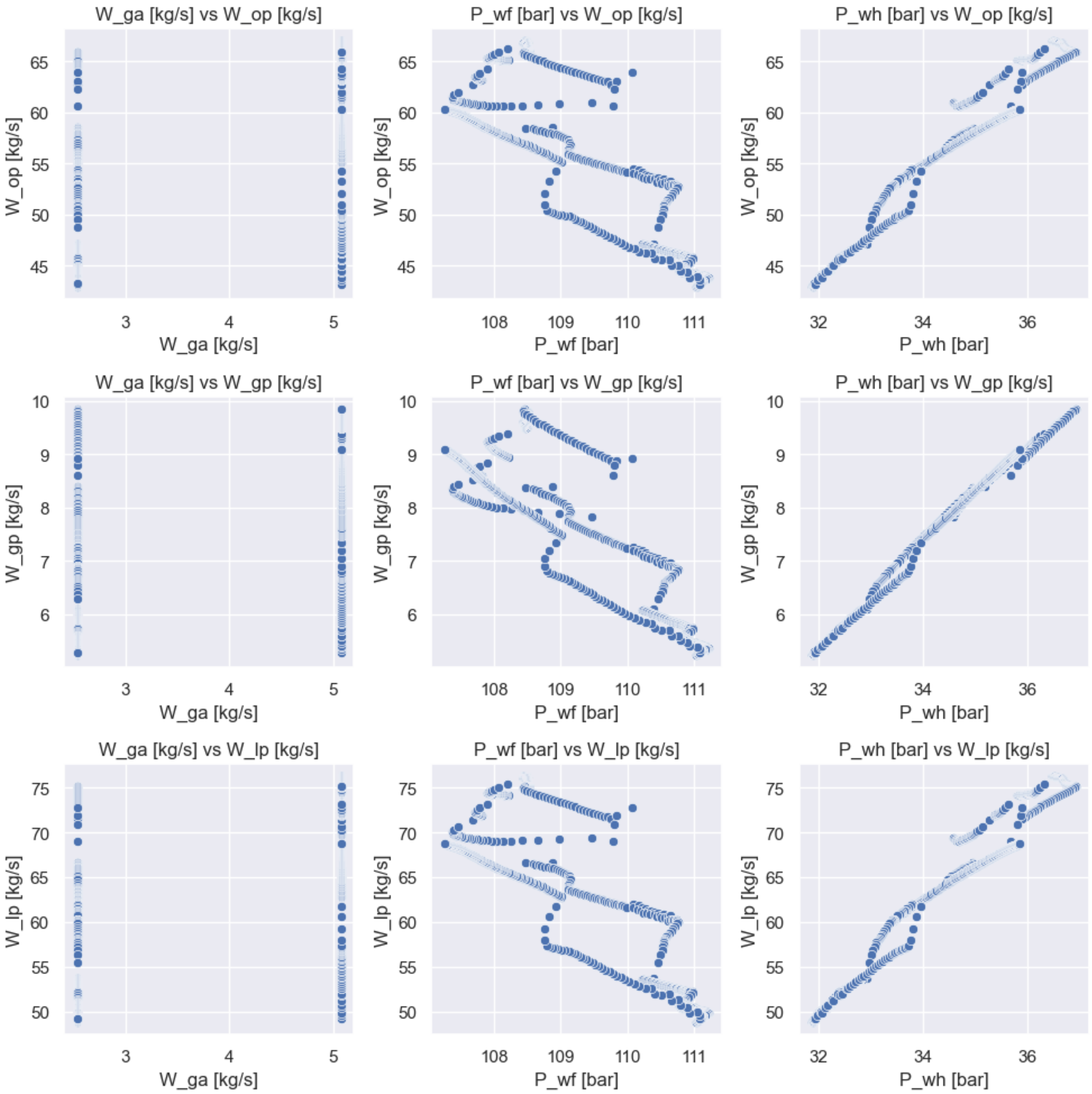


Figure 4.4: Plots of correlation graphs (Well 2)

4.4 Nested Validation

To tune the models, nested k fold cross validation is deployed. Since the data is a time dependant, the future data points should not be used to train the model. With the Time Series cross-validator of ”scikit-learn” it is easier to split the train data into train and validation sets. Two splits are chosen here, since the dataset is not large. To prevent overfitting early stopping is implemented. A visual representation of this splitting is shown in Fig 4.5.Feature scaling through standardization, is preformed on the train dataset. It involves rescaling each feature such that it has a standard deviation of 1 and a mean of 0.

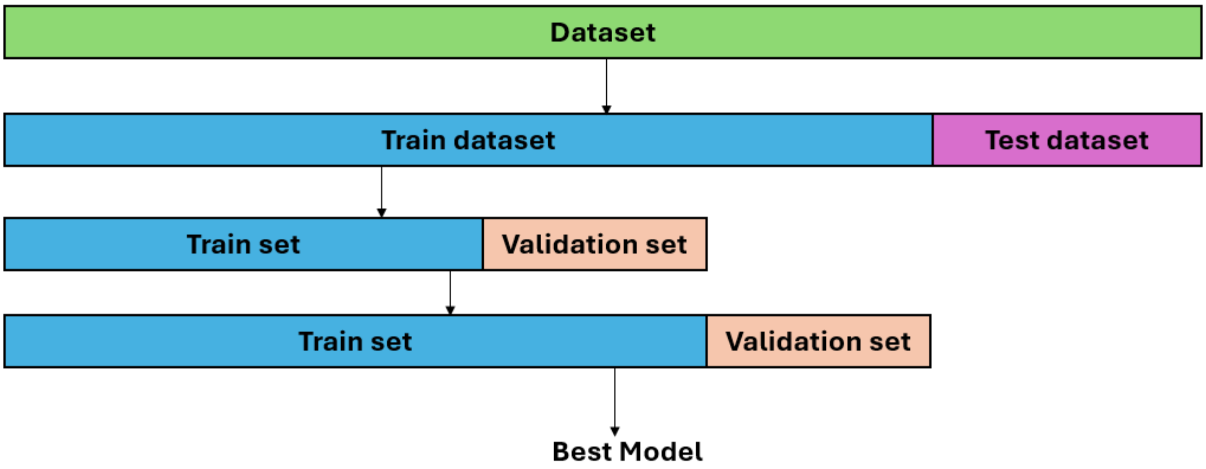


Figure 4.5: Nested k fold validation

4.5 LSTM Regression

Here a LSTM Regression model is trained and validated. The model is then used on the test set to obtain the MAPE metric. The predictions of the model on both sets are shown in Figures 4.6 and 4.7

- MAPE is 1.92% for well 1.
- MAPE is 2.11% for well 2.

The early stopping is determined by plotting the Training vs Validation loss as shown in Fig 4.8. This ensures the model is not overfitted on the training data. It can be easily implemented in Tensorflow. For the future algorithms a similar loss curves are obtained.

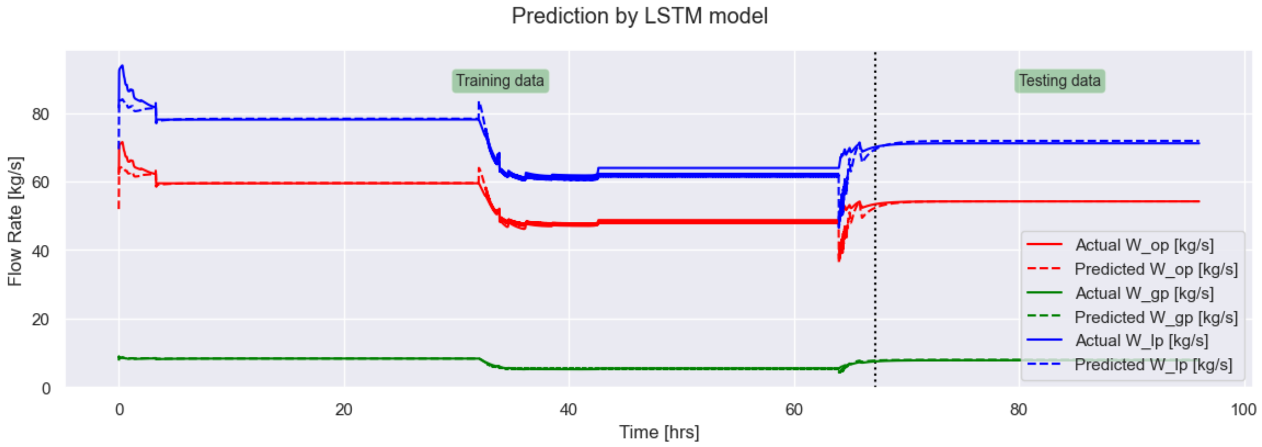


Figure 4.6: LSTM Regression model outputs on test set (Well 1)

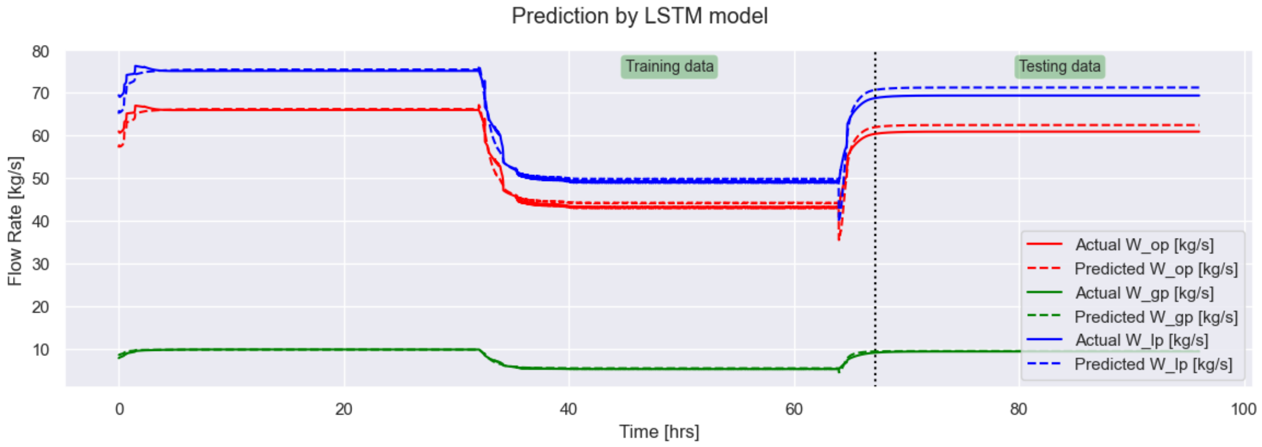


Figure 4.7: LSTM Regression model outputs on test set (Well 2)

For well 1, 32 memory cells were used. For well 2, 40 memory cells were used. The ’adam’ optimizer with loss function of mean squared error is used for training both models. A linear activation unit is used in the output layer

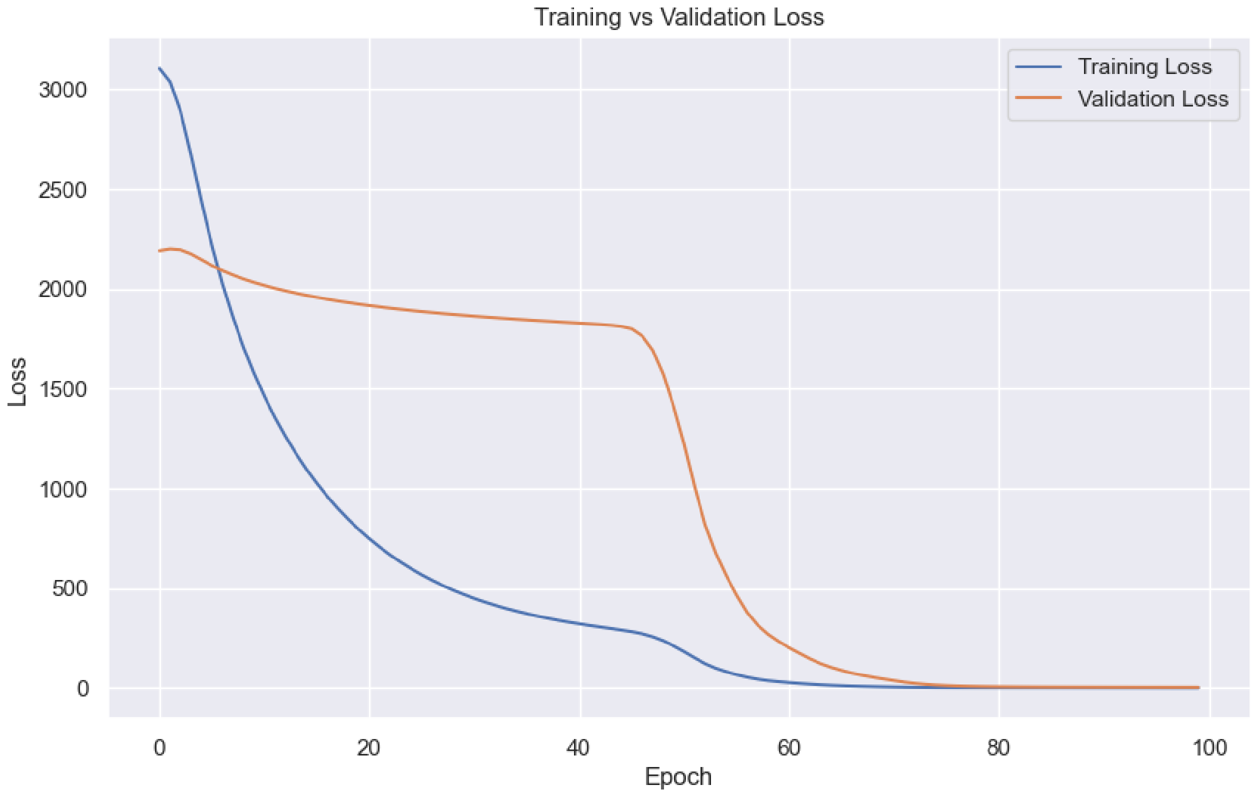


Figure 4.8: Training vs Validation loss

4.6 Multivariate Linear Regression

Here a Multivariate Linear Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions. GridSearchCV from Sklearn is used to find the best parameters for the models. The best parameters here are using the Intercept and overwriting the X parameter.

- MAPE is 2.14% for well 1.
- MAPE is 7.57% for well 2.

It can seen that the prediction for the W^i_{gp}, the mass flow rate of gas the model output is almost perfect in both wells. Well 1 has slightly more offset for oil and liquid flow rate compared to Well 2.

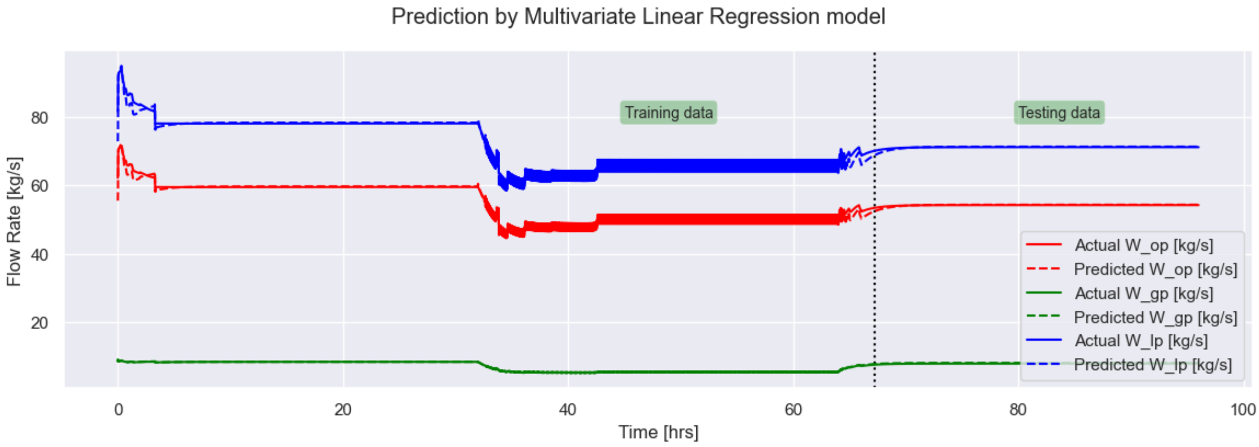


Figure 4.9: Multivariate Linear Regression outputs on test set (Well 1)

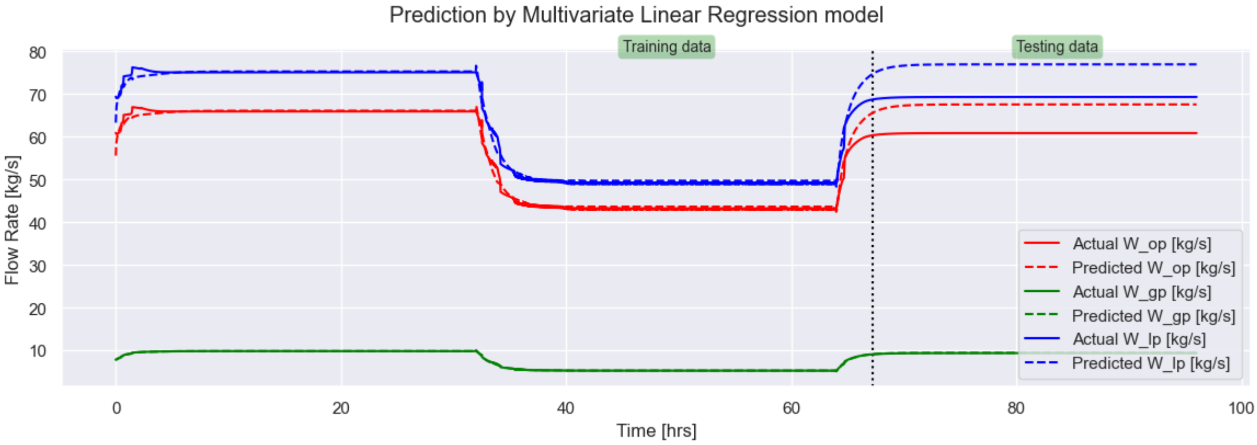


Figure 4.10: Multivariate Linear Regression outputs on test set (Well 2)

4.7 kNN Regression

Here a kNN Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions. With GridSerachCV Euclidean distance and 8 number of neighbours produces the best model for well 1. But for well 2 the Manhattan distance is found to give best model.

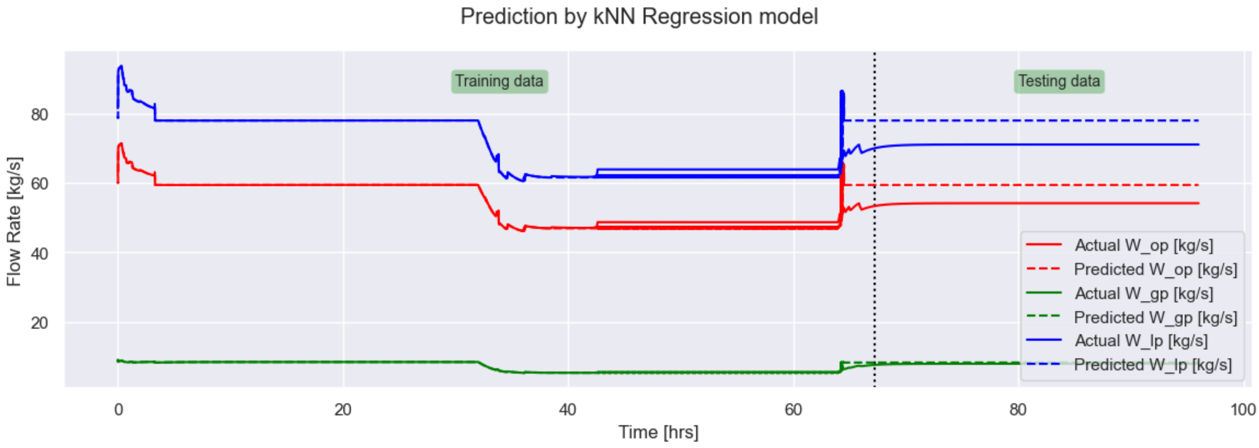


Figure 4.11: kNN Regression model outputs on test set (Well 1)

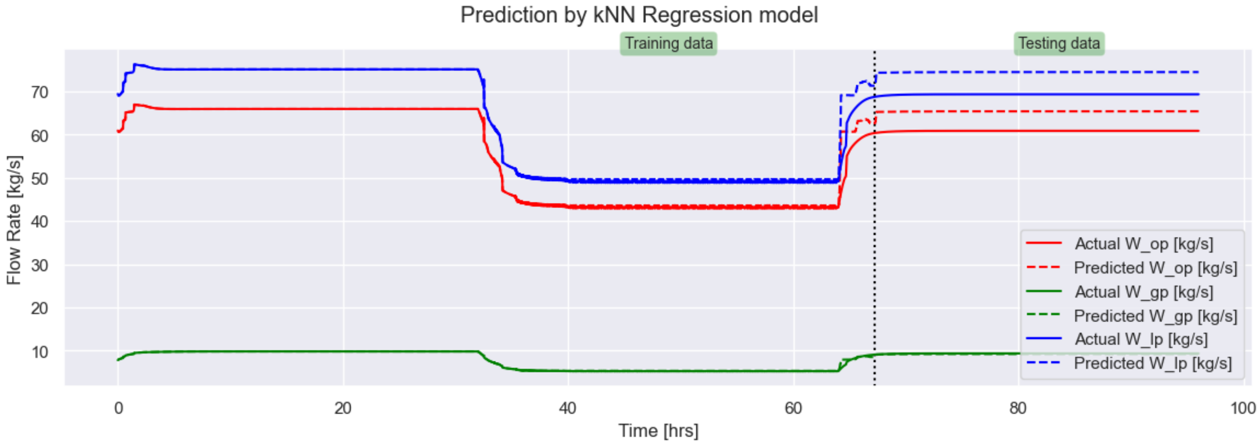


Figure 4.12: kNN Regression model outputs on test set (Well 2)

- MAPE is 8.05% for well 1.
- MAPE is 5.41% for well 2.

4.8 Support Vector Regression

Here a Support Vector Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions. From GridSearchCV kernel='rbf', C=1.0, gamma=0.1 are the best parameters for well 1. kernel='rbf', C=10.0, gamma=0.01 are best parameters for well 2.

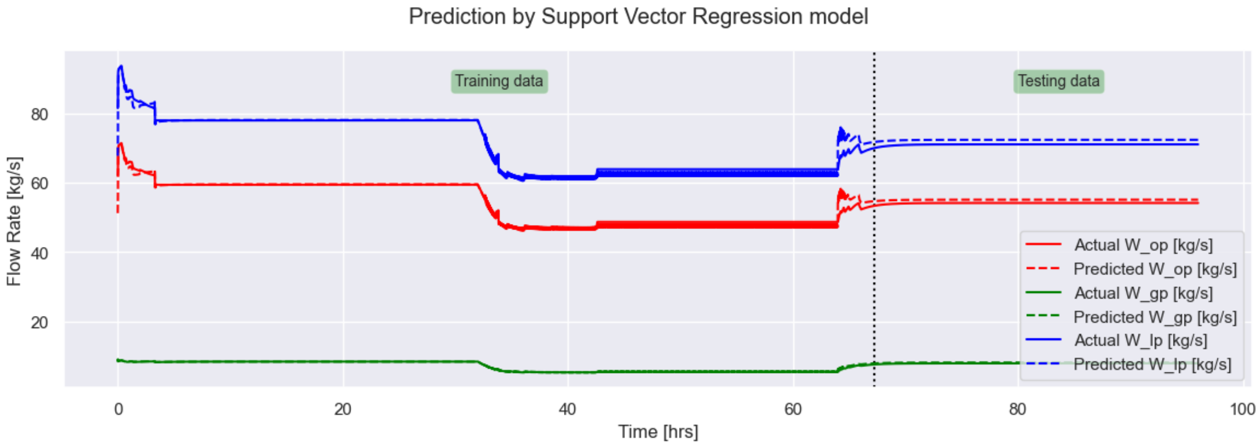


Figure 4.13: Support Vector Regression model outputs on test set (Well 1)

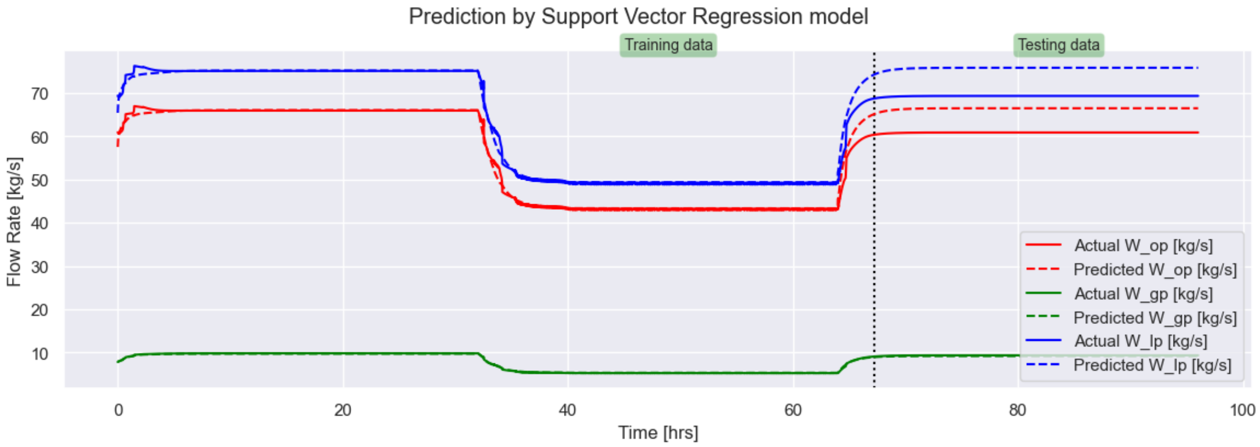


Figure 4.14: Support Vector Regression model outputs on test set (Well 2)

- MAPE is 5.05% for well 1.
- MAPE is 4.31% for well 2.

4.9 Decision Tree Regression

Here a Decision Tree Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions. From GridSearchCV , the max depth is one, max features is sqrt, min samples leaf is one , min samples split is two for well 1. For well 2, the max depth is ten, max features is log2, min samples leaf is one, min samples split is ten. This shows that each well has to be tuned individually to obtain best models.

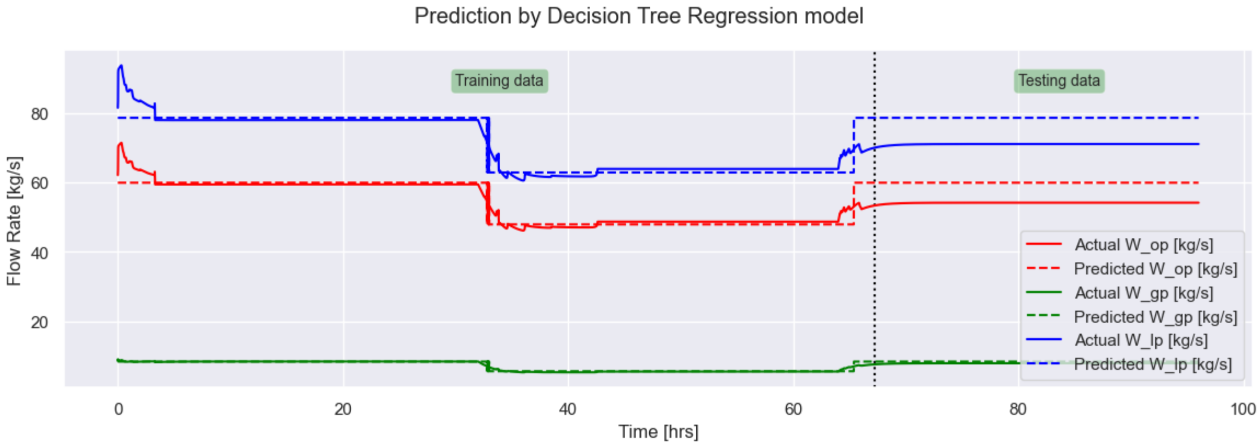


Figure 4.15: Decision Tree Regression model outputs on test set (Well 1)

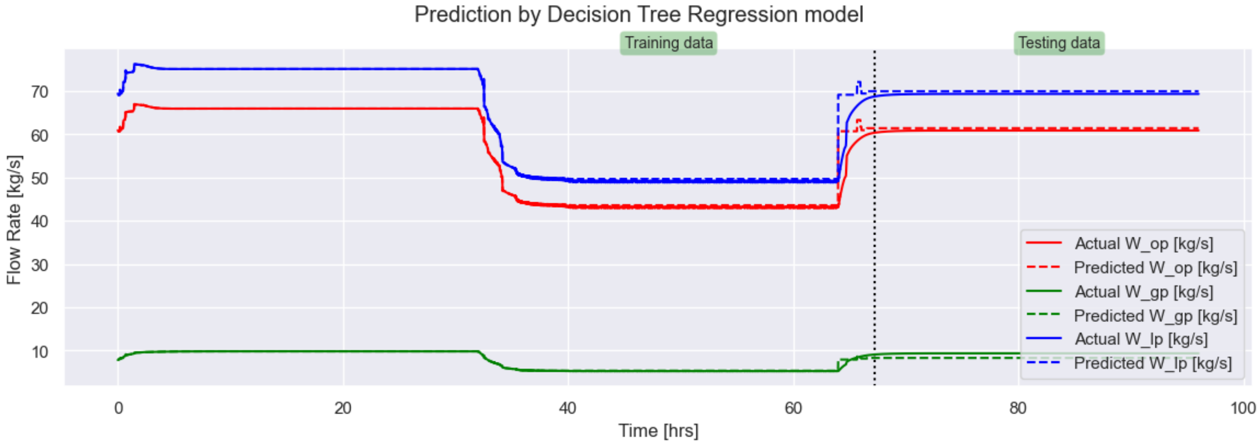


Figure 4.16: Decision Tree Regression model outputs on test set (Well 2)

- MAPE is 9.26% for well 1.
- MAPE is 5.31% for well 2.

4.10 Gradient Boosting Regression

Here a Gradient Boosting Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions. For well 1, learning rate=0.01, max depth=3, estimators=100. For well 2, learning rate=0.05, max depth=3, estimators=100 are the best hyper parameters.

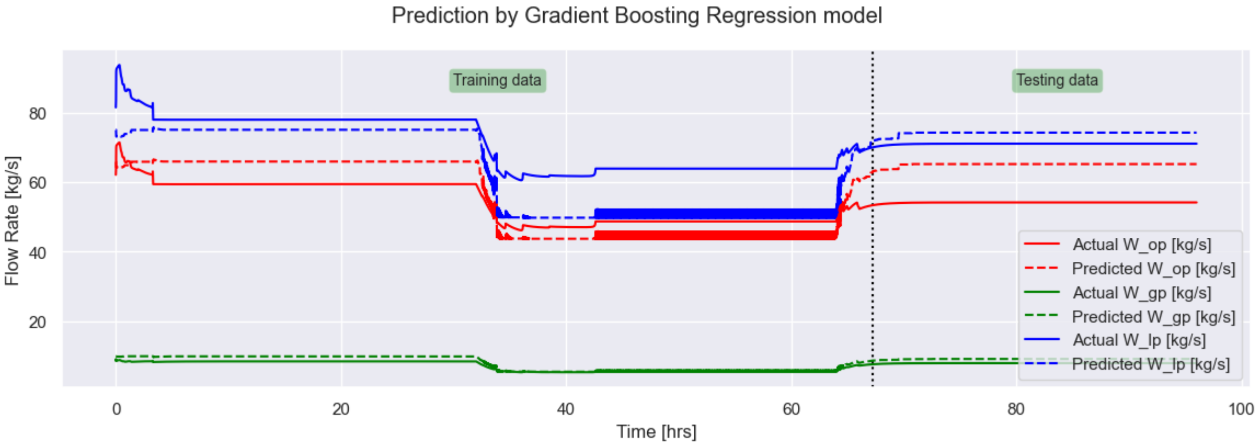


Figure 4.17: Gradient Boosting Regression model outputs on test set (Well 1)

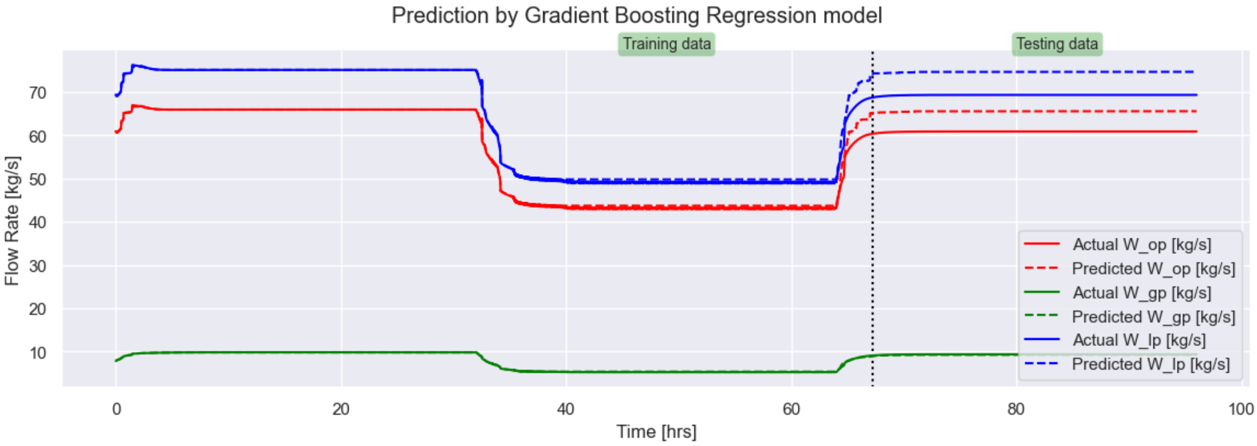


Figure 4.18: Gradient Boosting Regression model outputs on test set (Well 2)

- MAPE is 4.95% for well 1.
- MAPE is 5.55% for well 2.

In this algorithm the importance of the features on the model can also be found. P_{wh}, the wellhead pressure hows the maximum effect.

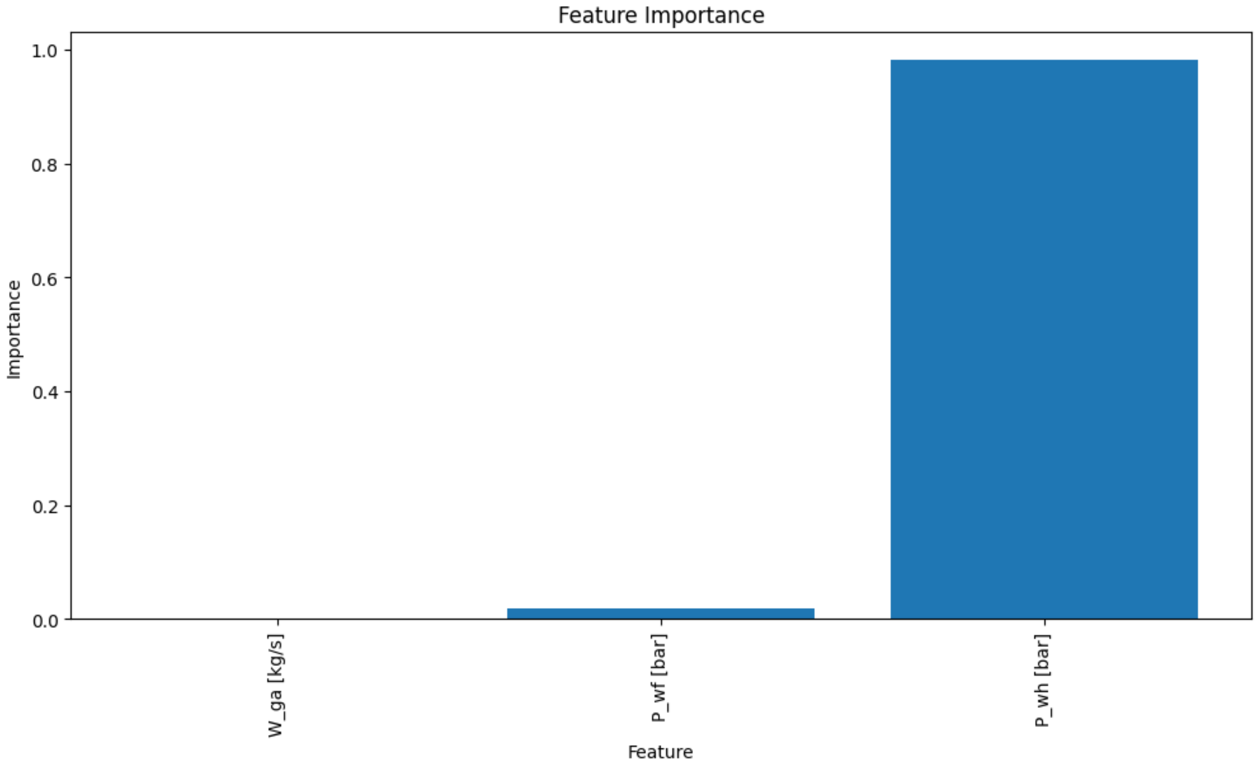


Figure 4.19: Feature importance from Gradient Boosting model

4.11 XGBoost Regression

XGBoost is a faster and more advanced version of Gradient Boosting algorithm. Here a XGBoost Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions. For well 1, learning rate=0.01, max depth=7, estimators=300. For well 2, learning rate=0.1, max depth=3, estimators=100 are the best hyper parameters.

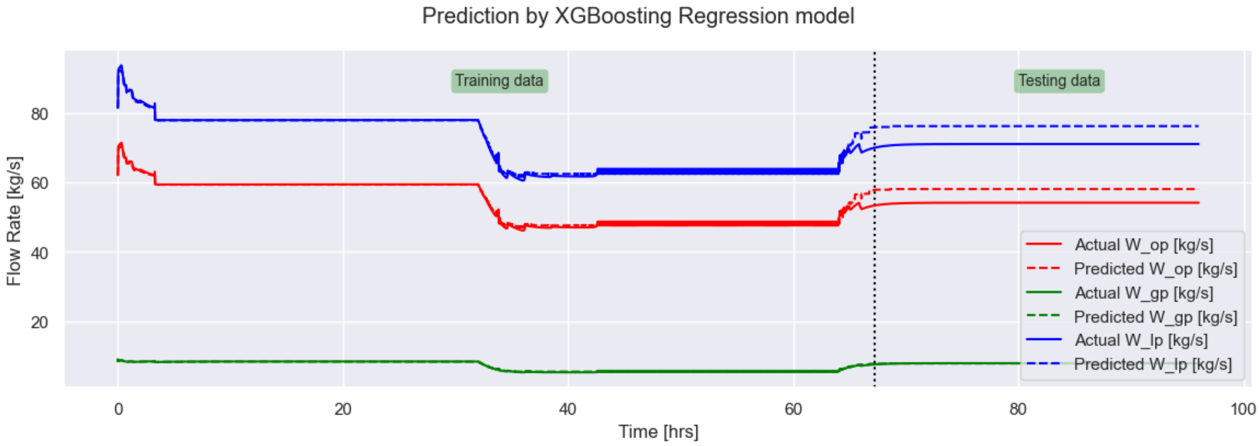


Figure 4.20: XGBoost Regression model outputs on test set (Well 1)

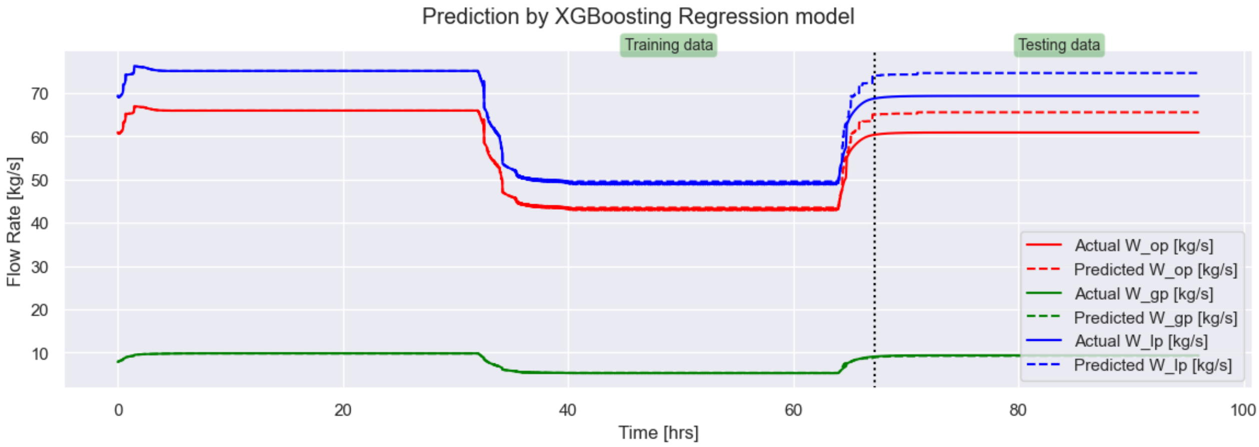


Figure 4.21: XGBoost Regression model outputs on test set (Well 2)

- MAPE is 4.23% for well 1.
- MAPE is 5.56% for well 2.

4.12 Principal component Regression

Here a Principal component model is trained and validated. The model is then used on the test set to obtain the flow rate predictions. One PC is used in both models.

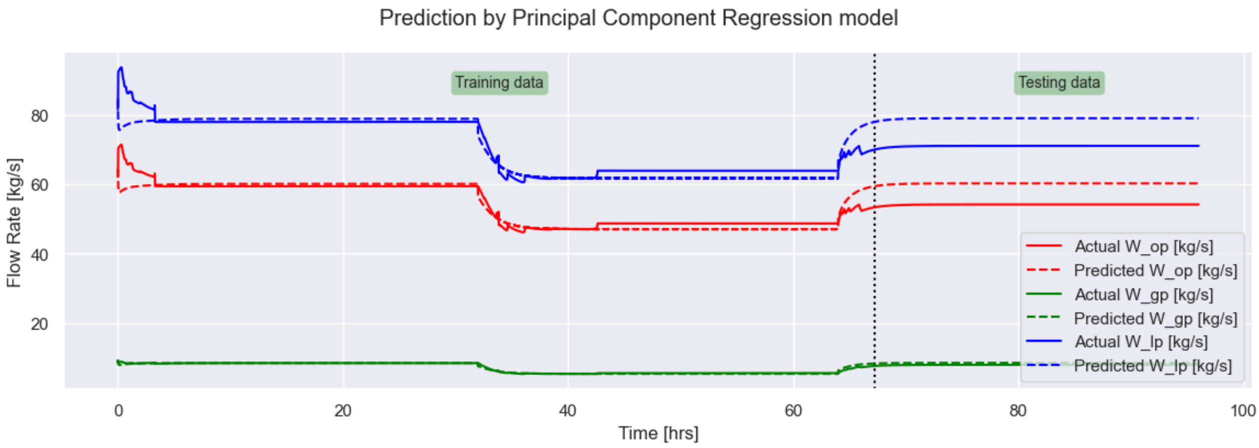


Figure 4.22: Principal Component Regression model outputs on test set (Well 1)

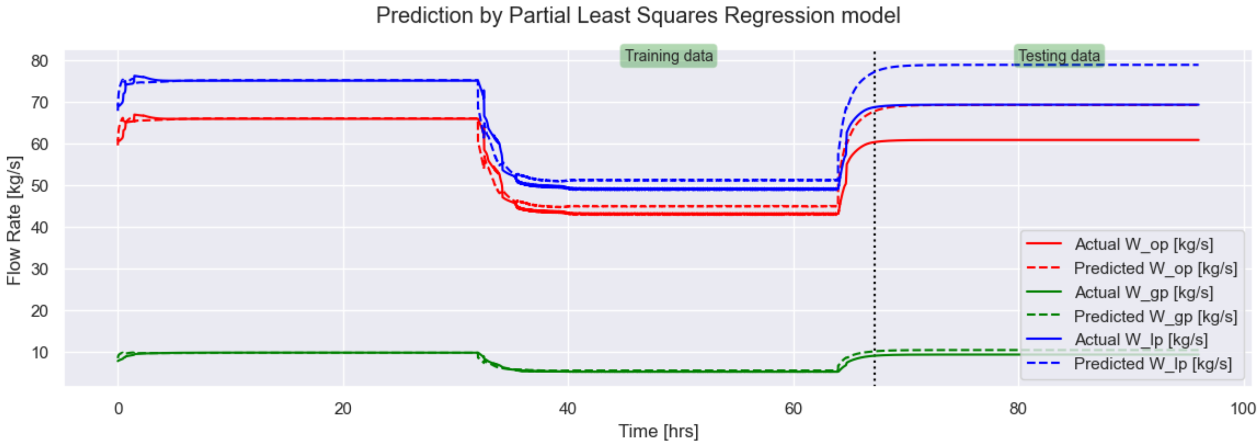


Figure 4.23: Principal Component Regression model outputs on test set (Well 2)

- MAPE is 9.52% for well 1.
- MAPE is 16.69% for well 2.

4.13 Partial Least Squares Regression

Here a Partial least squares Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions.

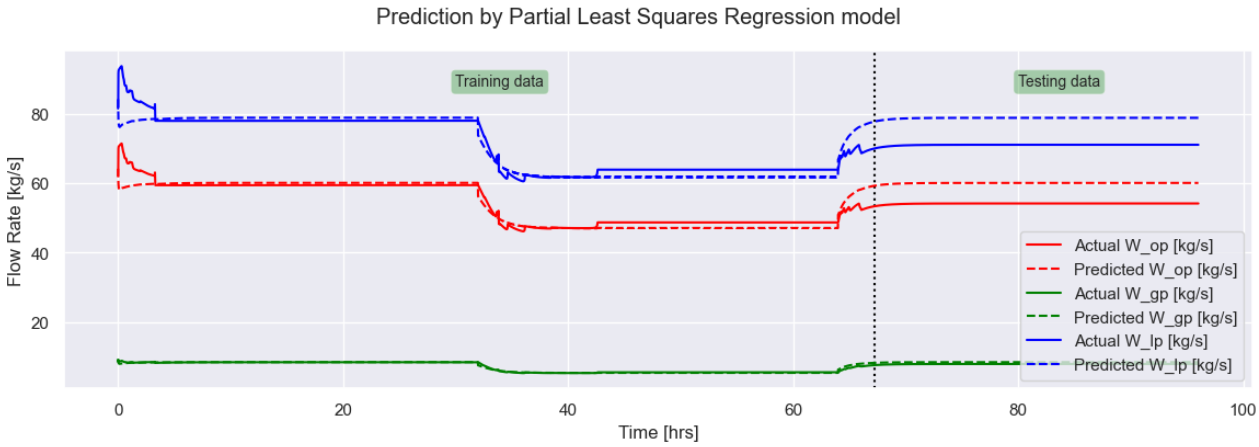


Figure 4.24: Partial least squares Regression model outputs on test set (Well 1)

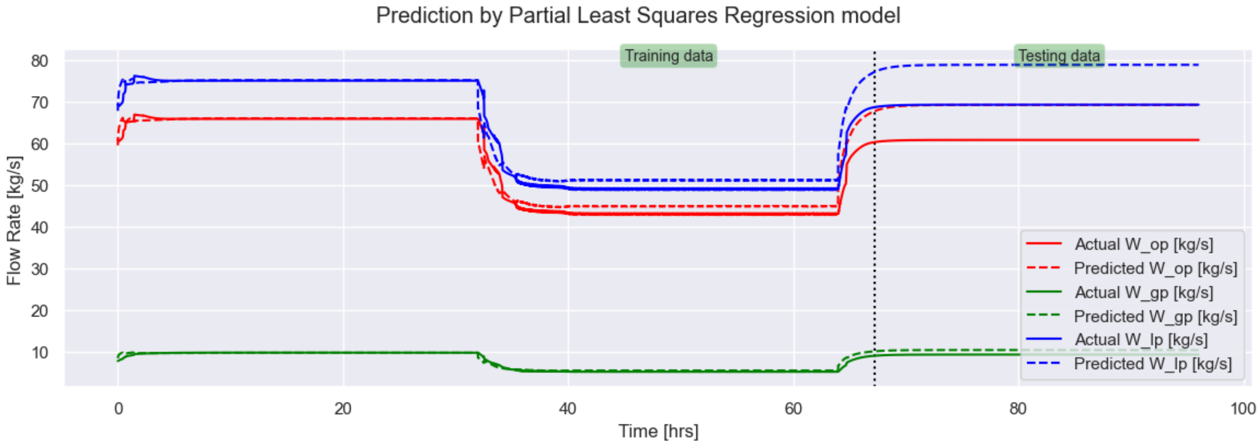


Figure 4.25: Partial least squares Regression model outputs on test set (Well 2)

- MAPE is 2.14% for well 1.
- MAPE is 7.57% for well 2.

4.14 MLP Neural Network Regression

Here a MLP neural network Regression model is trained and validated. The model is then used on the test set to obtain the flow rate predictions.

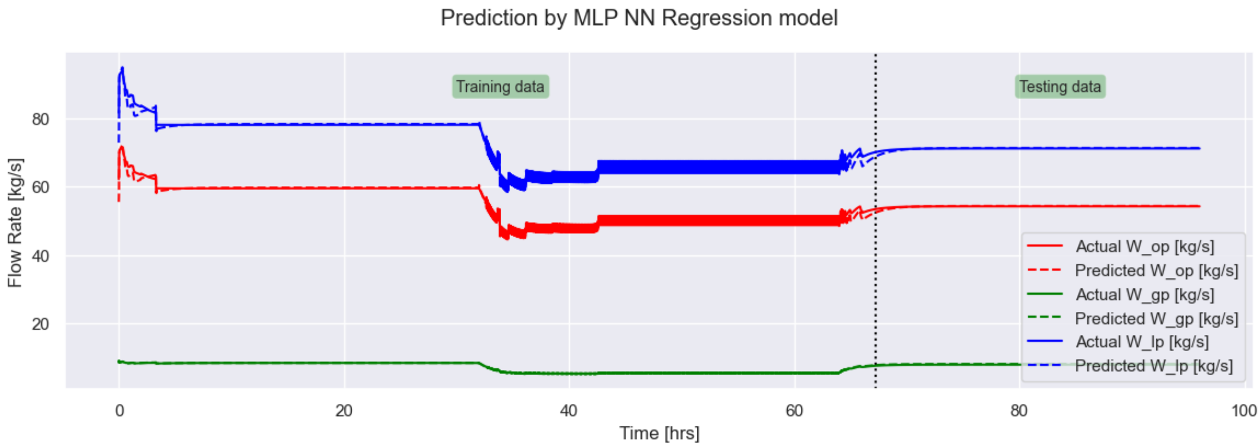


Figure 4.26: MLP neural network Regression model outputs on test set (Well 1)

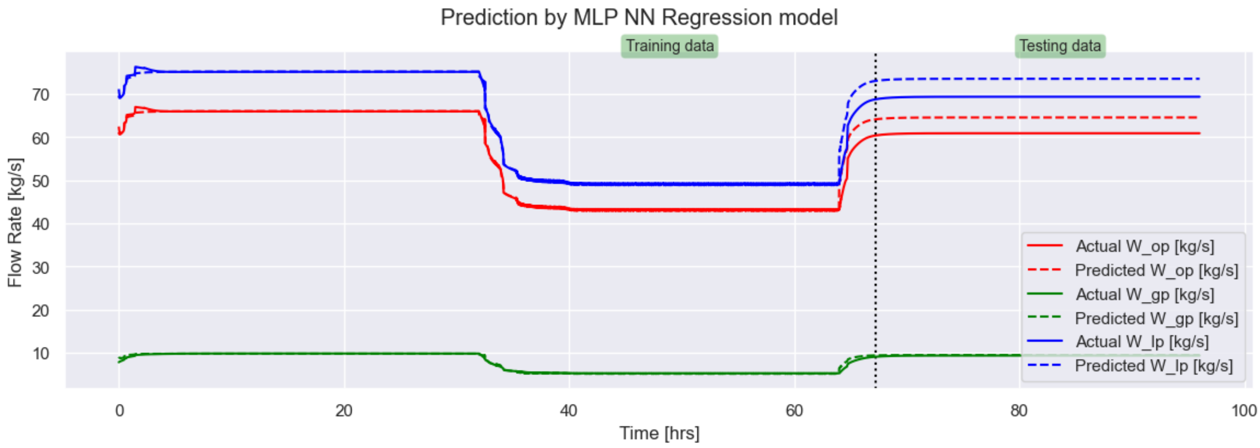


Figure 4.27: MLP neural network Regression model outputs on test set (Well 2)

- MAPE is 2.43% for well 1.
- MAPE is 5.49% for well 2.

5 Measurement Errors

This chapter describes various types of measurement errors which occur in the collection of process data. Methods to filter out and correct the errors are also explored. Outlier detection is also explored.

5.1 Types of errors

The three main types of measurement errors in process data are:

- Systematic errors
- Random errors
- Gross errors

Systematic errors, also known as determinate errors, are consistent and predictable deviations from the true value in measurements. These errors occur due to flaws in the measurement process, such as faulty equipment, incorrect calibration, or procedural mistakes. Systematic errors are consistent and always in the same direction. This consistency makes them particularly challenging to detect and correct, as they can significantly skew the results of an experiment or study. Examples of systematic errors include offset errors, where the instrument does not accurately return to zero, and scale factor errors, where measurements are consistently too high or too low by a certain percentage. Identifying and correcting these errors requires careful analysis of the measurement process and the use of control samples or standards to assess the accuracy of the measurements.

Random errors are a type of measurement error that affect measurements in unpredictable ways, meaning the measurements are equally likely to be higher or lower than the true values. This type of error is often referred to as ”noise” because it blurs the true value or the ”signal” of what’s being measured. Random errors are almost always present in research, even in highly controlled settings,their impact can be reduced using various methods. To reduce random errors, sample size can be increased, as large samples have less random error than small samples. This is because the errors in different directions cancel each other out more efficiently when you have more data points. Collecting data from a large sample increases precision and statistical power.

Gross errors in the measurement of process data refer to significant mistakes or oversights that occur during the measurement process, leading to a substantial deviation from the true value. These errors are primarily attributed to human factors, such as lack of experience, improper handling of instruments, poor judgment, and equipment failure. They can also be influenced by human factors like fatigue or stress, which can affect a user's ability to operate the measuring instrument accurately.

5.2 Effect of error on models

The effects of systematic and gross errors requires in-depth statistical analysis and will not be analysed here. The main focus here is to check the effect of random errors, in the form of noise. The effect of random errors will be tested on three machine learning models. XGBoost, MLP NN and LSTM.

5.2.1 Impulse noise

Impulse noise introduces sudden jumps or falls in the data values, simulating real-world data with occasional spikes at random locations. First a noise sample of 3% is created. The values in the sample are uniformly distributed between 20% of the minimum value of the column and 30% of the maximum value of the column. This ensures that the noise added is relative to the range of the data in the column. The noise is randomly distributed across the column and added to the 3 input features. The 3 algorithms are trained and tested. Here the figures are shown of only Well 1, since the effects are very similarly observed in Well 2.

The MAPE for well 1 is reduced to 5.98% and 4.67% for Well 2. For MLP NN the MAPE

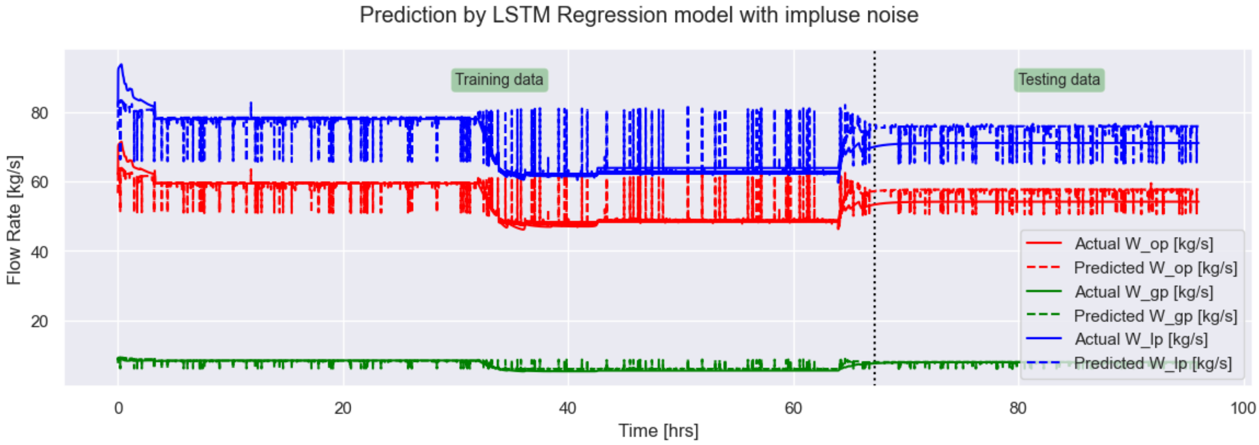


Figure 5.1: Effect of Impulse noise on LSTM model

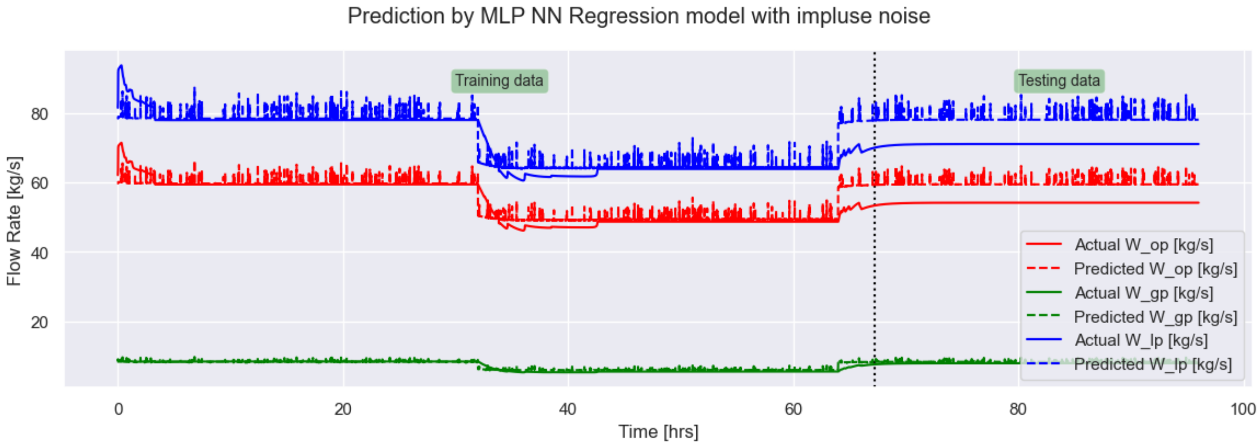


Figure 5.2: Effect of Impulse noise on MLP NN model

is reduced to 8.76% for well 1 and 5.13% for well 2. Fig 5.2 shows the model output. For XGBoost the MAPE is reduced to 7.51% for well 1 and 5.17% for well 2. Fig 5.3 shows the model output.

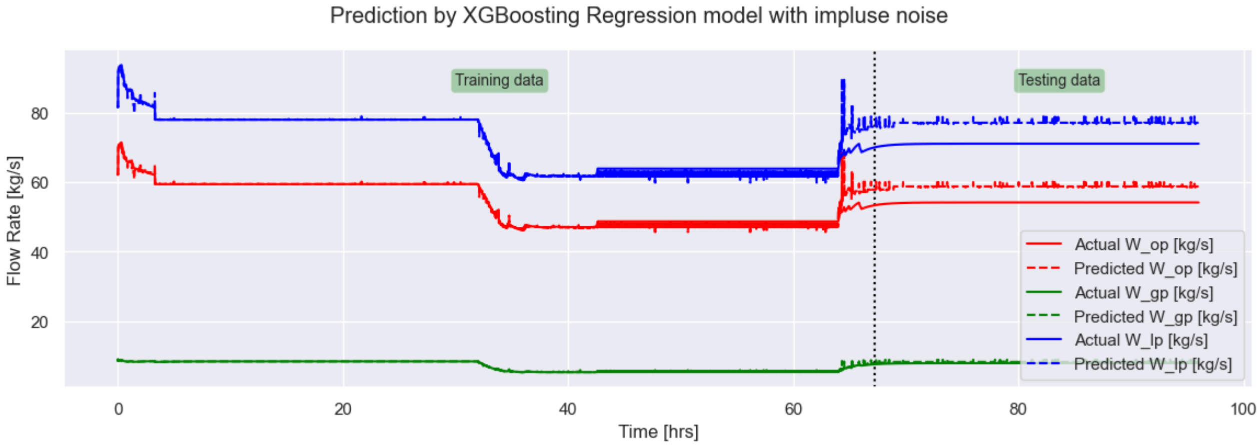


Figure 5.3: Effect of Impulse noise on XGBoost model

5.2.2 Filtering noise

To solve the problem of impulse noise, there are many filters that can be used. For example Median filter, Order statistic filters, and so on.

Here the Median Filter is used to reduce the impulse noise. SciPy is used her which has a median filter function is well-suited for removing impulse noise, as it replaces each data

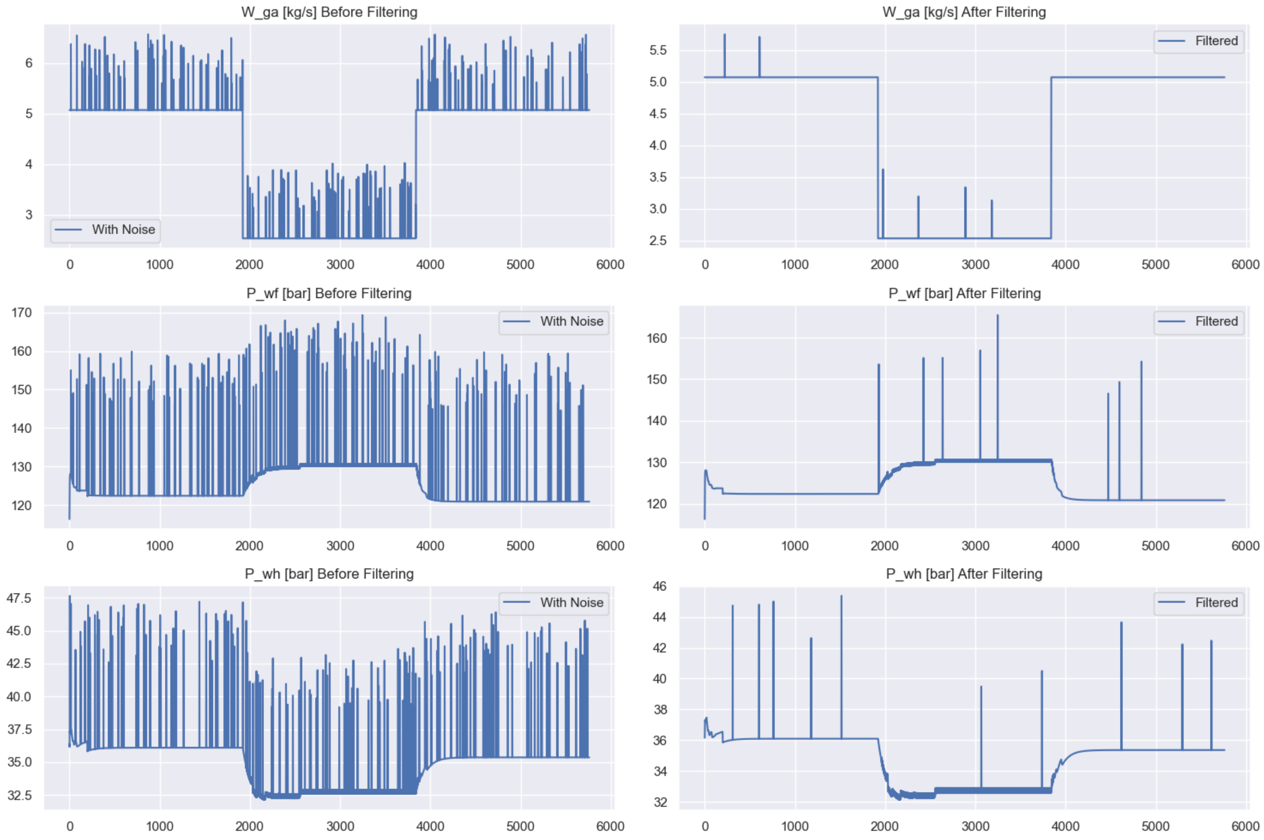


Figure 5.4: Median noise filter

point with the median of the neighboring data points within a specified window size. The results of the median filter is shown in Fig 5.4. A window size of 3 is used. Each data point is replaced with the median of itself and its two neighbors. Most of the impulses are filtered out. The prediction accuracy of the 3 models is improved.

Table 5.1: Impulse noise effects

Well no.	LSTM (%)	MLP NN (%)	XGBoost (%)
Well 1	5.98	8.76	7.51
Well 2	4.67	5.13	5.77

Table 5.2: Median Filter effects

Well no.	LSTM (%)	MLP NN (%)	XGBoost (%)
Well 1	1.87	4.97	6.47
Well 2	2.86	5.51	5.29

6 Uncertainty

Uncertainty refers to a state of limited knowledge or information, where it is impossible to precisely describe an existing state, future outcome, or multiple possible outcomes. There are two main types of uncertainty [62]:

- Aleatory Uncertainty: This type of uncertainty arises from the inherent randomness or variability in natural phenomena or processes. It is an irreducible uncertainty that cannot be reduced through additional measurements or increased knowledge. Aleatory uncertainty is best modelled using probability distributions and is often referred to as ”irreducible” or ”objective” uncertainty.
- Epistemic Uncertainty: Epistemic uncertainty stems from a lack of knowledge or incomplete information about a system or phenomenon. It is a ”reducible” uncertainty that can potentially be reduced through additional measurements, experiments, or increased understanding. Epistemic uncertainty can arise from various sources, such as imprecise measurements, incomplete data, inadequate models, or a lack of understanding of the underlying processes.

6.0.1 Uncertainty in machine learning

There are many methods to quantify the uncertainty in predictions for machine learning models. Some of them are: Confidence intervals, Quantile regression, Bootstrapping, Ensemble methods and Bayesian optimization.

Using XGBoost the confidence intervals can be easily added. For other algorithms like LSTM it is more difficult. Fig 6.1 and Fig 6.2 shows the confidence intervals of 95% for XGBoost model for well 1 and well 2.

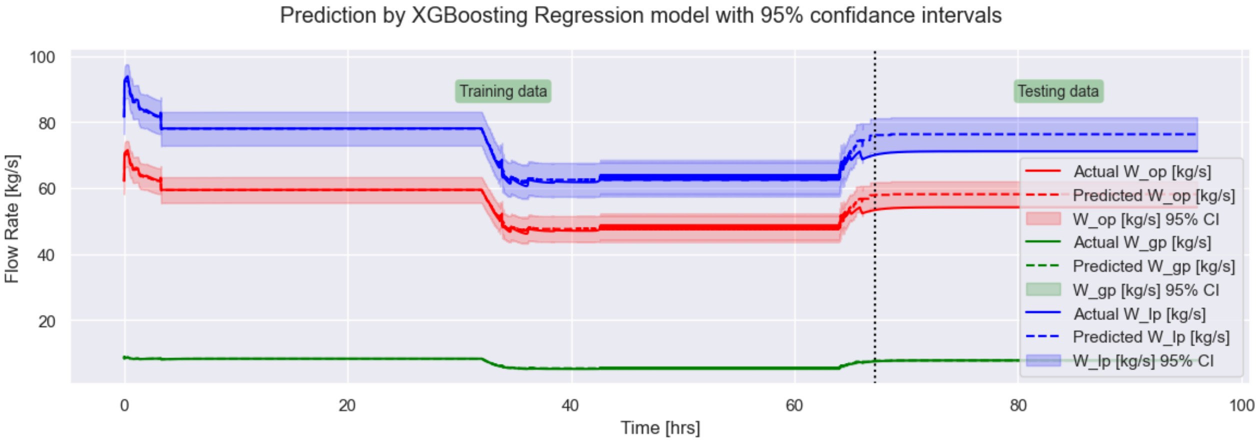


Figure 6.1: Confidence intervals for XGBoost (Well 1)

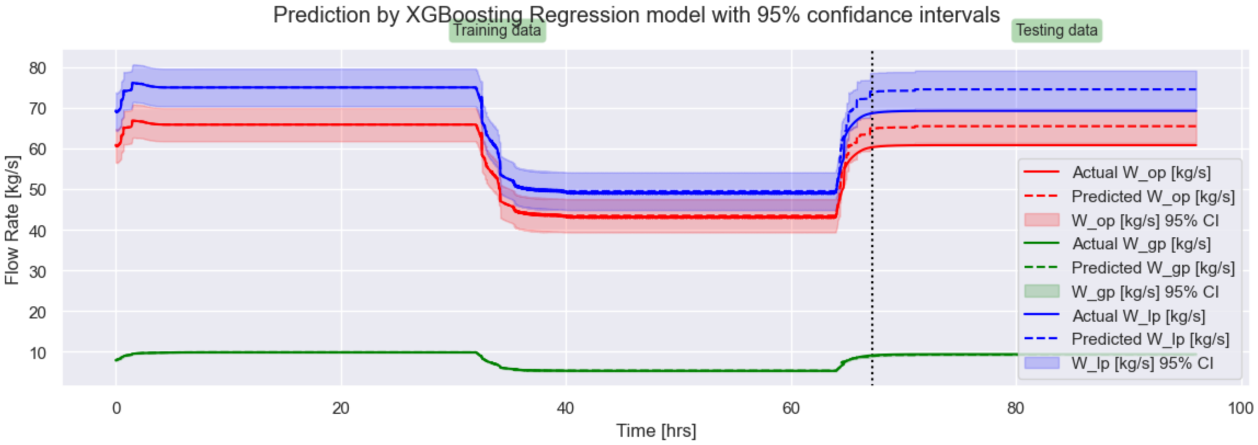


Figure 6.2: Confidence intervals for XGBoost (Well 2)

7 Results and Discussions

Ten algorithms were used to create models for Well 1 and Well 2. This is detailed in Chapter 4. MAPE is used to describe the performance. The LSTM model produces the best results. The disadvantages of using this is the training time is longer. Also to find the proper parameters is a time consuming process. It is observed that for each well the hyper-parameters has to be tuned. GridSearchCV helps with this, but it is still a complicated process.

For the algorithms that are generally used for classification tasks like SVM, kNN, some modification is required to enabling its use for regression. Many of these algorithms including linear regression, and tree based, require modification to predict multiple outputs. With modifications it is possible to get the results, but the downside is the hyperparameter tuning becomes more complex.

Neural networks and the LSTM model can be made more complex, giving better results. This takes more time and computation power. For finding the best hyper-parameters multiple runs are required. Since the programs were executed on a laptop, these take more time. For decrease in computation time a sample size of 5762 was used. If more samples were used in the modelling the results would probably be much better.

Table 7 shows the MAPE for each model for Well 1 and Well 2

Table 7.1: MAPE for Well 1 and Well 2

Algorithm	Well 1 (%)	Well 2 (%)
LSTM	1.96	1.53
MLP NN	2.43	5.49
MV Linear Regression	2.14	7.57
SVR	5.04	4.31
KNN	8.05	5.41
Decision Tree	9.26	5.43
Gradient Boost	4.95	5.55
XGBoost	4.23	5.56
PLS	9.54	7.57
PCR	9.52	16.69

The effects of impulse noise on the prediction performance is shown in Chapter 5. With the median filter the effects of noise can be removed. With further experimentation better filtering can be obtained.

Using 95% confidence intervals in XGBoost model the prediction uncertainty has been quantified. This is easy in XGBoost, but for other algorithms like LSTM, neural networks, SVM the implementation is harder.

7.1 Future Work

More filters can be used in removing measurement noise. Different methods of uncertainty quntification can also be tested.

The outlier detection and correction was not executed due to time constraints. This can be added in future. Unsupervised techniques like Local Outlier Factor, Isolation Forest, Kernel Density Estimation can be tested.

Data reconciliation can also be added. Here the process flow diagram is necessary, the constraints of the each well are also needed.

8 Conclusion

Considering the objectives of the Thesis as mention in Section 1.2, all the of them are completed. The literature review has shown that considerable work has been done on use of machine learning in the oil and gas industry. Research is ongoing to improve the algorithms for this estimation of various parameters. The data collection and preprocessing is the first step of any machine learning project. It can be said that with the proper data the future steps of machine learning are useless. For this thesis ten machine learning algorithms were studied. For the two wells the best performing algorithm is LSTM. As mentioned in the previous section, it has limitations. The effect of errors on the prediction performance in the form of impulse noise has been explored. This shows that filtering of the data is very important. The influence of errors can have a impact on the machine learning prediction. Finally the uncertainty in the prediction is quantified using confidence intervals.

The application of machine learning for flow rate estimation in oil and gas productions is a complex process. From the data collection to uncertainty quantification, considerable has work to be done to obtain useful results. The applicability of the results depends on the situation. It may be best to use the predictions from the models as a backup for more robust systems. Each well has to be modelled individually since they have different characteristics. In addition more process data would probably improve the accuracy of the flow rate predictions.

References

[1] T. Bikmukhametov and J. Jäschke, 'First principles and machine learning virtual flow metering: A literature review,' *Journal of Petroleum Science and Engineering*, vol. 184, p. 106 487, Jan. 2020, ISSN: 09204105. DOI: 10.1016/j.petrol.2019.106487. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0920410519309088.

[2] J. Qiu and H. Toral, 'Three-phase flow-rate measurement by pressure transducers,' in *All Days*, Houston, Texas: SPE, 3rd Oct. 1993, SPE–26567–MS. DOI: 10.2118/26567-MS. [Online]. Available: https://onepetro.org/SPEATCE/proceedings/93SPE/All-93SPE/SPE-26567-MS/55306.

[3] 'What are multiphase flow meters?' (2018), [Online]. Available: https://www.arcweb.com/blog/what-multiphase-flow-meters.

[4] L. S. Hansen, S. Pedersen and P. Durdevic, 'Multi-phase flow metering in offshore oil and gas transportation pipelines: Trends and perspectives,' *Sensors*, vol. 19, no. 9, p. 2184, 11th May 2019, ISSN: 1424-8220. DOI: 10.3390/s19092184. [Online]. Available: https://www.mdpi.com/1424-8220/19/9/2184.

[5] 'Subsea multiphase flow meter (subsea-mpfm).' (2020), [Online]. Available: http://www.haimotech.com/Products-and-Services/mpfm/Subsea-MPFM.html.

[6] G. Falcone, C. Alimonti, G. Hewitt and B. Harrison, 'Multiphase flow metering: 4 years on,' Jan. 2005.

[7] A. Line and J. Fabre, 'Stratified gas liquid flow,' in Jan. 1997, pp. 1097–1101, ISBN: 0849393566. DOI: 10.1615/AtoZ.s.stratified_gas-liquid_flow.

[8] M. Meribout, A. Azzi, N. Ghendour, N. Kharoua, L. Khezzar and E. AlHosani, 'Multiphase flow meters targeting oil & gas industries,' *Measurement*, vol. 165, p. 108 111, Dec. 2020, ISSN: 02632241. DOI: 10.1016/j.measurement.2020.108111. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0263224120306497.

[9] C. Arnold, L. Biedebach, A. Küpfer and M. Neunhoeffer, 'The role of hyperparameters in machine learning models and how to tune them,' *Political Science Research and Methods*, pp. 1–8, ISSN: 2049-8470, 2049-8489. DOI: 10.1017/psrm.2023.61. [Online]. Available: https://www.cambridge.org/core/product/identifier/S2049847023000614/type/journal_article.

[10] T. A. AL-Qutami, R. Ibrahim, I. Ismail and M. A. Ishak, 'DEVELOPMENT OF SOFT SENSOR TO ESTIMATE MULTIPHASE FLOW RATES USING NEURAL NETWORKS AND EARLY STOPPING,' *International Journal on Smart Sensing and Intelligent Systems*, vol. 10, no. 1, pp. 1–24, 1st Jan. 2017, ISSN: 1178-5608. DOI: 10.21307/ijssis-2017-209. [Online]. Available: https://www.sciendo.com/article/10.21307/ijssis-2017-209.

[11] T. A. AL-Qutami, R. Ibrahim, I. Ismail and M. A. Ishak, 'Radial basis function network to predict gas flow rate in multiphase flow,' in *Proceedings of the 9th International Conference on Machine Learning and Computing*, Singapore Singapore: ACM, 24th Feb. 2017, pp. 141–146, ISBN: 978-1-4503-4817-1. DOI: 10.1145/3055635.3056638. [Online]. Available: https://dl.acm.org/doi/10.1145/3055635.3056638.

[12] T. A. AL-Qutami, R. Ibrahim, I. Ismail and M. A. Ishak, 'Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing,' *Expert Systems with Applications*, vol. 93, pp. 72–85, Mar. 2018, ISSN: 09574174. DOI: 10.1016/j.eswa.2017.10.014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417417306875.

[13] M. A. Ahmadi, M. Ebadi, A. Shokrollahi and S. M. J. Majidi, 'Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir,' *Applied Soft Computing*, vol. 13, no. 2, pp. 1085–1098, Feb. 2013, ISSN: 15684946. DOI: 10.1016/j.asoc.2012.10.009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1568494612004589.

[14] M. D. AlAjmi, S. A. Alarifi and A. H. Mahsoon, 'Improving multiphase choke performance prediction and well production test validation using artificial intelligence: A new milestone,' in *SPE Digital Energy Conference and Exhibition*, The Woodlands, Texas, USA: Society of Petroleum Engineers, 2015. DOI: 10.2118/173394-MS. [Online]. Available: http://www.onepetro.org/doi/10.2118/173394-MS.

[15] A. Al-Jasmi, H. K. Goel, H. Nasr *et al.*, 'Short-term production prediction in real time using intelligent techniques,' in *All Days*, London, UK: SPE, 10th Jun. 2013, SPE–164813–MS. DOI: 10.2118/164813-MS. [Online]. Available: https://onepetro.org/SPEEURO/proceedings/13EURO/All-13EURO/London,%20UK/177282.

[16] C. Alimonti and G. Falcone, 'Integration of multiphase flowmetering, neural networks, and fuzzy logic in field performance monitoring,' *SPE Production & Facilities*, vol. 19, no. 1, pp. 25–32, 1st Feb. 2004, ISSN: 1064-668X. DOI: 10.2118/87629-PA. [Online]. Available: https://onepetro.org/PO/article/19/01/25/110854/Integration-of-Multiphase-Flowmetering-Neural (visited on 22/04/2024).

[17] S. Mollaiy Berneti and M. Shahbazian, 'An imperialist competitive algorithm artificial neural network method to predict oil flow rate of the wells,' *International Journal of Computer Applications*, vol. 26, no. 10, pp. 47–50, 31st Jul. 2011, ISSN: 09758887. DOI: 10.5120/3137-4326. [Online]. Available: http://www.ijcaonline.org/volume26/number10/pxc3874326.pdf.

[18] M. Hasanvand and S. M. Berneti, 'Predicting oil flow rate due to multiphase flow meter by using an artificial neural network,' *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 37, no. 8, pp. 840–845, 18th Apr. 2015, ISSN: 1556-7036, 1556-7230. DOI: 10.1080/15567036.2011.590865. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/15567036.2011.590865.

[19] A. García, I. Almeida, G. Singh *et al.*, 'An implementation of on-line well virtual metering of oil production,' in *All Days*, Utrecht, The Netherlands: SPE, 23rd Mar. 2010, SPE–127520–MS. DOI: 10.2118/127520-MS. [Online]. Available: https://onepetro.org/SPEIE/proceedings/10IE/All-10IE/Utrecht,%20The%20Netherlands/106758.

[20] T. Denney, B. Wolfe and D. Zhu, 'Benefit evaluation of keeping an integrated model during real-time ESP operations,' in *All Days*, The Woodlands, Texas, USA: SPE, 5th Mar. 2013, SPE–163704–MS. DOI: 10.2118/163704-MS. [Online]. Available: https://onepetro.org/SPEDEC/proceedings/13DEC/All-13DEC/The%20Woodlands,%20Texas,%20USA/176910.

[21] P. Shoeibi Omrani, I. Dobrovolschi, S. Belfroid, P. Kronberger and E. Munoz, 'Improving the accuracy of virtual flow metering and back-allocation through machine learning,' in *Day 2 Tue, November 13, 2018*, Abu Dhabi, UAE: SPE, 12th Nov. 2018, D021S035R004. DOI: 10.2118/192819-MS. [Online]. Available: https://onepetro.org/SPEADIP/proceedings/18ADIP/2-18ADIP/Abu%20Dhabi,%20UAE/213263.

[22] G. Olivares, C. Escalona and E. Gimenez, 'Production monitoring using artificial intelligence, APLT asset,' in *All Days*, Utrecht, The Netherlands: SPE, 27th Mar. 2012, SPE–149594–MS. DOI: 10.2118/149594-MS. [Online]. Available: https://onepetro.org/SPEIE/proceedings/12IE/All-12IE/Utrecht,%20The%20Netherlands/157537.

[23] H. Shaban and S. Tavoularis, 'Measurement of gas and liquid flow rates in two-phase pipe flows by the application of machine learning techniques to differential pressure signals,' *International Journal of Multiphase Flow*, vol. 67, pp. 106–117, Dec. 2014, ISSN: 03019322. DOI: 10.1016/j.ijmultiphaseflow.2014.08.012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0301932214001608.

[24] N. Andrianov, 'A machine learning approach for virtual flow metering and forecasting,' *IFAC-PapersOnLine*, vol. 51, no. 8, pp. 191–196, 2018, ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.06.376. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896318307067.

[25] K. Loh, P. S. Omrani and R. van der Linden, 'Deep learning and data assimilation for real-time production prediction in natural gas wells,' 2018, Publisher: [object Object] Version Number: 2. DOI: 10.48550/ARXIV.1802.05141. [Online]. Available: https://arxiv.org/abs/1802.05141 (visited on 22/04/2024).

[26] J. Sun, X. Ma and M. Kazi, 'Comparison of decline curve analysis DCA with recursive neural networks RNN for production forecast of multiple wells,' in *Day 4 Wed, April 25, 2018*, Garden Grove, California, USA: SPE, 22nd Apr. 2018, D041S012R009. DOI: 10.2118/190104-MS. [Online]. Available: https://onepetro.org/SPEWRM/proceedings/18WRM/4-18WRM/D041S012R009/215403.

[27] L. Xu, W. Zhou and X. Li, 'Wet gas flow modeling for a vertically mounted venturi meter,' *Measurement Science and Technology*, vol. 23, no. 4, p. 045 301, 1st Apr. 2012, ISSN: 0957-0233, 1361-6501. DOI: 10.1088/0957-0233/23/4/045301. [Online]. Available: https://iopscience.iop.org/article/10.1088/0957-0233/23/4/045301.

[28] G. Zangl, R. Hermann and C. Schweiger, 'Comparison of methods for stochastic multiphase flow rate estimation,' in *All Days*, Amsterdam, The Netherlands: SPE, 27th Oct. 2014, SPE–170866–MS. DOI: 10.2118/170866-MS. [Online]. Available: https://onepetro.org/SPEATCE/proceedings/14ATCE/All-14ATCE/Amsterdam,%20The%20Netherlands/211819.

[29] C. Gerrard, I. C. Taylor, K.-C. Goh and F. de Boer, 'Implementing real-time production optimisation in shell e&p in europe—changing the way we work and run our business,' in *All Days*, Aberdeen, Scotland, U.K.: SPE, 4th Sep. 2007, SPE–108515–MS. DOI: 10.2118/108515-MS. [Online]. Available: https://onepetro.org/SPEOE/proceedings/07OE/All-07OE/SPE-108515-MS/142527.

[30] H. Poulisse, P. van Overschee, J. Briers, C. Moncur and K. .-. Goh, 'Continuous well production flow monitoring and surveillance,' in *All Days*, Amsterdam, The Netherlands: SPE, 11th Apr. 2006, SPE–99963–MS. DOI: 10.2118/99963-MS. [Online]. Available: https://onepetro.org/SPEIE/proceedings/06IE/All-06IE/SPE-99963-MS/141244.

[31] B. Grimstad, P. M. Robertson and B. Foss, 'Virtual flow metering using b-spline surrogate models,' *IFAC-PapersOnLine*, vol. 48, no. 6, pp. 292–297, 2015, ISSN: 24058963. DOI: 10.1016/j.ifacol.2015.08.046. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896315009131.

[32] T. Bikmukhametov and J. Jäschke, 'Oil production monitoring using gradient boosting machine learning algorithm,' *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 514–519, 2019, ISSN: 24058963. DOI: 10.1016/j.ifacol.2019.06.114. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896319302009.

[33] O. Bello, S. Ade-Jacob and K. Yuan, 'Development of hybrid intelligent system for virtual flow metering in production wells,' in *All Days*, Utrecht, The Netherlands: SPE, 1st Apr. 2014, SPE–167880–MS. DOI: 10.2118/167880-MS. [Online]. Available: https://onepetro.org/SPEIE/proceedings/14IE/All-14IE/Utrecht,%20The%20Netherlands/212206.

[34] T. A. Al-Qutami, R. Ibrahim and I. Ismail, 'Hybrid neural network and regression tree ensemble pruned by simulated annealing for virtual flow metering application,' in *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Kuching: IEEE, Sep. 2017, pp. 304–309, ISBN: 978-1-5090-5559-3. DOI: 10.1109/ICSIPA.2017.8120626. [Online]. Available: http://ieeexplore.ieee.org/document/8120626/.

[35] N. Janatian, K. Jayamanne and R. Sharma, 'Model based control and analysis of gas lifted oil field for optimal operation,' presented at the The First SIMS EUROSIM Conference on Modelling and Simulation, SIMS EUROSIM 2021, and 62nd International Conference of Scandinavian Simulation Society, SIMS 2021, September 21-23, Virtual Conference, Finland, 31st Mar. 2022, pp. 241–246. DOI: 10.3384/ecp21185241. [Online]. Available: https://ecp.ep.liu.se/index.php/sims/article/view/351.

[36] N. Janatian and R. Sharma, 'A reactive approach for real-time optimization of oil production under uncertainty,' in *2023 American Control Conference (ACC)*, San Diego, CA, USA: IEEE, 31st May 2023, pp. 2658–2663, ISBN: 9798350328066. DOI: 10.23919/ACC55779.2023.10156274. [Online]. Available: https://ieeexplore.ieee.org/document/10156274/.

[37] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Third edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2023, 834 pp., ISBN: 978-1-09-812597-4.

[38] 'Multiple linear regression (mlr) definition, formula, and example.' (2023), [Online]. Available: https://www.investopedia.com/terms/m/mlr.asp.

[39] 'What is the k-nearest neighbors (knn) algorithm?' (2024), [Online]. Available: https://www.ibm.com/topics/knn.

[40] 'K-nearest neighbor.' (2009), [Online]. Available: http://www.scholarpedia.org/article/K-nearest_neighbor.

[41] 'Knn algorithm: When? why? how?' (2020), [Online]. Available: https://towardsdatascience.com/knn-algorithm-what-when-why-how-41405c16c36f.

[42] 'Support vector machine (svm) algorithm.' (2023), [Online]. Available: https://www.geeksforgeeks.org/support-vector-machine-algorithm/.

[43] 'Knn algorithm: When? why? how?' (2020), [Online]. Available: https://www.niser.ac.in/~smishra/teach/cs460/2020/lectures/lec13_1/.

[44] A. J. Smola and B. Schölkopf, 'A tutorial on support vector regression,' *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004, ISSN: 0960-3174. DOI: 10.1023/B:STCO.0000035301.49549.88. [Online]. Available: http://link.springer.com/10.1023/B:STCO.0000035301.49549.88.

[45] 'Decision tree.' (2023), [Online]. Available: https://www.geeksforgeeks.org/decision-tree/.

[46] 'Decision tree algorithm, explained.' (2022), [Online]. Available: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html/.

[47] 'How decision tree classification and regression algorithm works.' (2023), [Online]. Available: https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-decision-tree-classification-and-regression-works.htm.

[48] 'Gradient boost for regression explained.' (2021), [Online]. Available: https://www.numpyninja.com/post/gradient-boost-for-regression-explained.

[49] 'All you need to know about gradient boosting algorithm.' (2022), [Online]. Available: https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502.

[50] 'Gradient boosting regression.' (2023), [Online]. Available: https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html.

[51] T. Chen and C. Guestrin, 'XGBoost: A scalable tree boosting system,' 2016, Publisher: [object Object] Version Number: 3. DOI: 10.48550/ARXIV.1603.02754. [Online]. Available: https://arxiv.org/abs/1603.02754.

[52] 'Xgboost.' (2023), [Online]. Available: https://www.nvidia.com/en-us/glossary/xgboost/.

[53] 'Xgboost for regression.' (2021), [Online]. Available: https://machinelearningmastery.com/xgboost-for-regression/.

[54] 'Lesson 11: Principal components analysis (pca).' (2020), [Online]. Available: https://online.stat.psu.edu/stat505/book/export/html/670.

[55] 'Chapter 7 principal component analysis.' (2019), [Online]. Available: https://bookdown.org/hailiangdu80/Machine_Learning_and_Neural_Networks/pcr.html.

[56] A.-L. Boulesteix and K. Strimmer, 'Partial least squares: A versatile tool for the analysis of high-dimensional genomic data,' *Briefings in Bioinformatics*, vol. 8, no. 1, pp. 32–44, 26th May 2006, ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/bbl016. [Online]. Available: https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbl016.

[57] 'Partial least squares regression.' (2019), [Online]. Available: https://allmodelsarewrong.github.io/pls.html.

[58] 'Multi-layer perceptrons explained and illustrated.' (2023), [Online]. Available: https://towardsdatascience.com/multi-layer-perceptrons-8d76972afa2b.

[59] 'Multilayer perceptron explained with a real-life example and python code: Sentiment analysis.' (2021), [Online]. Available: https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141l.

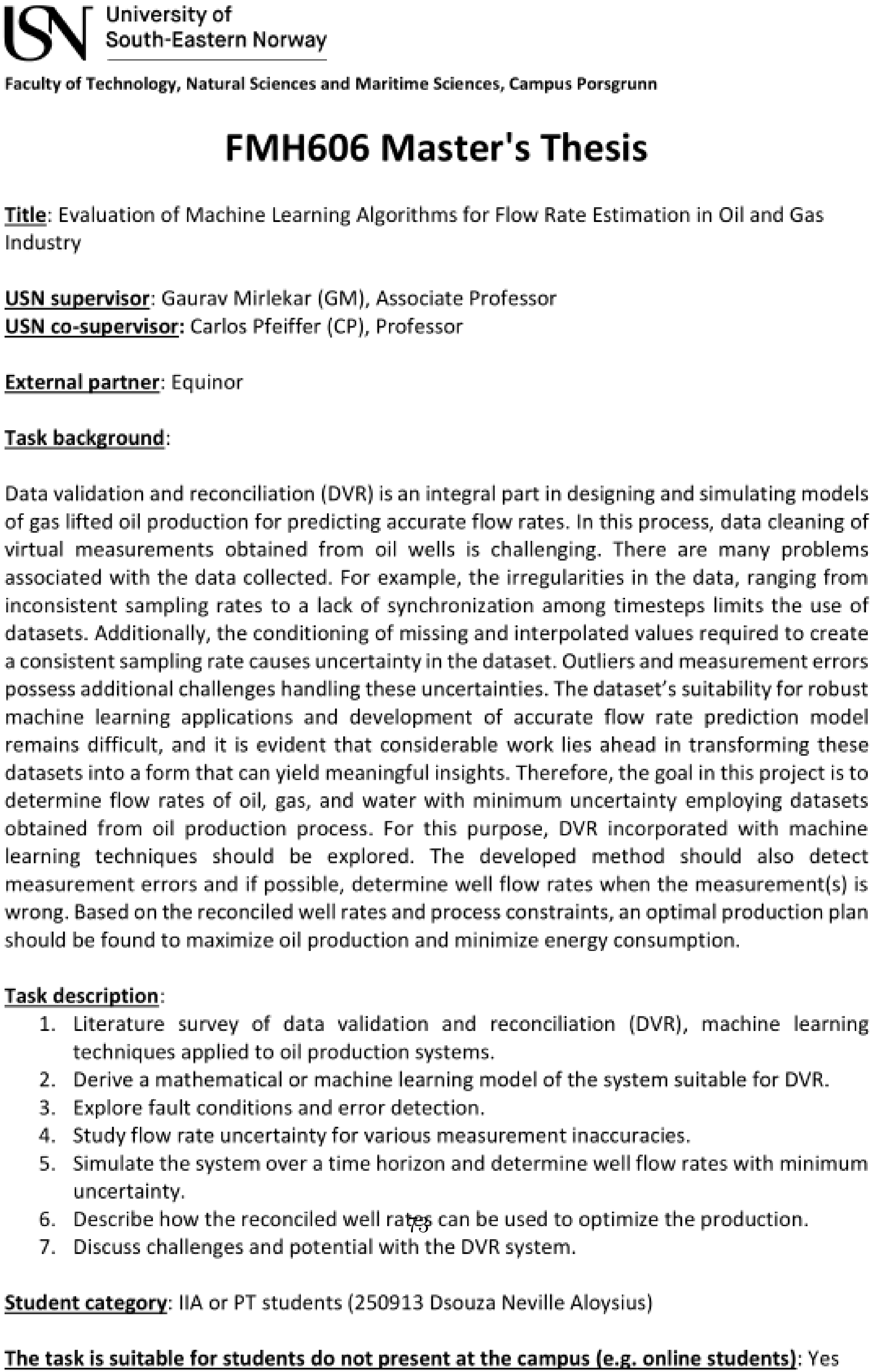
[60] 'A gentle introduction to long short-term memory networks by the experts.' (2021), [Online]. Available: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/.

[61] 'Understanding lstm and its diagrams.' (2016), [Online]. Available: https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714.

[62] P. F. Pelz, M. E. Pfetsch, S. Kersting *et al.*, 'Types of uncertainty,' in *Mastering Uncertainty in Mechanical Engineering*, P. F. Pelz, P. Groche, M. E. Pfetsch and M. Schaeffner, Eds. Cham: Springer International Publishing, 2021, pp. 25–42, ISBN: 978-3-030-78354-9. DOI: 10.1007/978-3-030-78354-9_2. [Online]. Available: https://doi.org/10.1007/978-3-030-78354-9_2.

Appendix A

Task Description



University of
South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: Evaluation of Machine Learning Algorithms for Flow Rate Estimation in Oil and Gas Industry

USN supervisor: Gaurav Mirlekar (GM), Associate Professor
USN co-supervisor: Carlos Pfeiffer (CP), Professor

External partner: Equinor

Task background:

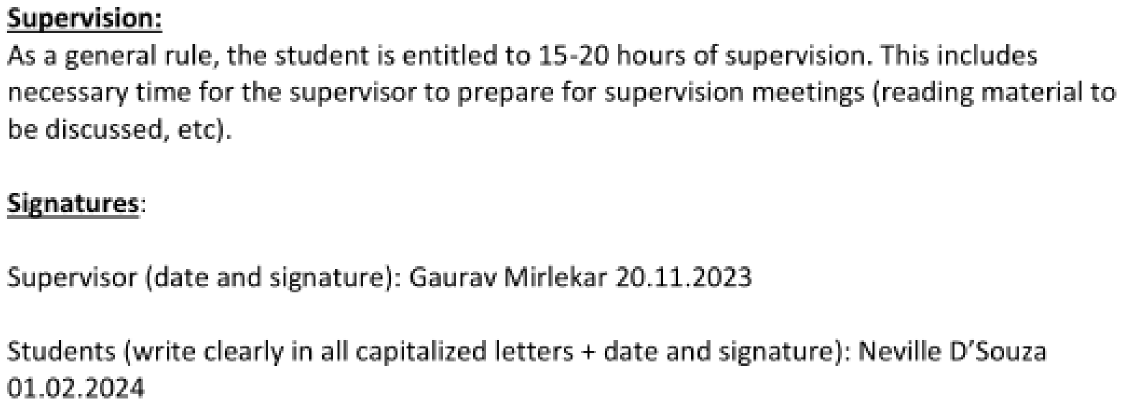
Data validation and reconciliation (DVR) is an integral part in designing and simulating models of gas lifted oil production for predicting accurate flow rates. In this process, data cleaning of virtual measurements obtained from oil wells is challenging. There are many problems associated with the data collected. For example, the irregularities in the data, ranging from inconsistent sampling rates to a lack of synchronization among timesteps limits the use of datasets. Additionally, the conditioning of missing and interpolated values required to create a consistent sampling rate causes uncertainty in the dataset. Outliers and measurement errors possess additional challenges handling these uncertainties. The dataset's suitability for robust machine learning applications and development of accurate flow rate prediction model remains difficult, and it is evident that considerable work lies ahead in transforming these datasets into a form that can yield meaningful insights. Therefore, the goal in this project is to determine flow rates of oil, gas, and water with minimum uncertainty employing datasets obtained from oil production process. For this purpose, DVR incorporated with machine learning techniques should be explored. The developed method should also detect measurement errors and if possible, determine well flow rates when the measurement(s) is wrong. Based on the reconciled well rates and process constraints, an optimal production plan should be found to maximize oil production and minimize energy consumption.

Task description:

1. Literature survey of data validation and reconciliation (DVR), machine learning techniques applied to oil production systems.
2. Derive a mathematical or machine learning model of the system suitable for DVR.
3. Explore fault conditions and error detection.
4. Study flow rate uncertainty for various measurement inaccuracies.
5. Simulate the system over a time horizon and determine well flow rates with minimum uncertainty.
6. Describe how the reconciled well rates can be used to optimize the production.
7. Discuss challenges and potential with the DVR system.

Student category: IIA or PT students (250913 Dsouza Neville Aloysius)

The task is suitable for students do not present at the campus (e.g. online students): Yes



Supervision:

As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:

Supervisor (date and signature): Gaurav Mirlekar 20.11.2023

Students (write clearly in all capitalized letters + date and signature): Neville D'Souza 01.02.2024

Appendix B

Program codes

The Matlab codes for the simulator and the python machine learning code can be accessed here: https://github.com/dsouzaneville/FMH606-1-Masters-Thesis