AN INTEGRATED FRAMEWORK FOR PRODUCTION DATA ANALYSIS USING MACHINE LEARNING AND WAVELETS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ENERGY SCIENCE AND ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Dante Isaac Orta Alemán

December 2022

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Roland Horne, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Tapan Mukerji

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Daniel Tartakovsky

Approved for the Stanford University Committee on Graduate Studies.

Stacey F. Bent, Vice Provost for Graduate Education

Abstract

Modeling well responses through data-driven methods requires the handling of large amounts of data to capture the complexity of the physical processes and operation modes taking place in a well. Data-driven models often benefit from the careful design of input features that simplify the learning task for the modeling algorithm, but its design increases the complexity and can be nonscalable. Moreover, the data used for training such models often is long-term production data, which has imperfections such as noise, abrupt changes, and missing data gaps. Thus, most modeling frameworks require substantial data cleaning and selection of specific time periods potentially eliminating useful data.

This work combined a wavelet-based decomposition with machine learning and deep learning algorithms in a framework to generate full data-driven well models. Wavelet transforms are useful for preserving relevant information such as short-term localized events as well as longer duration effects present in data. These properties make them an ideal candidate for the design of useful features for modeling reservoir response in combination with machine and deep learning algorithms. This work focused on using pressure and flow rate data to build a data-driven model a well's response. A method for model feature design using wavelet transforms is introduced. The research shows that applying the Maximum Overlap Discrete Wavelet Transform Multiresolution Analysis (MODWT-MRA) to pressure and flow rate data from a well results in a decomposition analogous to a set of superposed wells in space. The MODWT-MRA thus splits the complex reservoir response signal into a set of simpler ones. This property is then leveraged to systematically design model features that are then used by machine learning and deep learning models to build data-driven models.

Two specific applications are presented, capturing a well's pressure response from flow rate data and reconstructing the flow rate history. Within each application, two scenarios are also shown, single-well and double-well. For creating the models, the linear Lasso and the deep-learning LassoNet models are explored. Both methodologies are shrinkage methods, however they differ in their complexity and capacity as function approximators. The results reveal significant benefits of applying the MODWT-MRA for input feature generation. Handling of noise becomes trivial with the use of the MODWT-MRA and there is an increase in model accuracy when compared with base scenarios without the use of the MODWT-MRA. The methodology also allowed for the use of incomplete datasets as the uncertainty caused by missing data gaps is encapsulated within a few levels of the decomposition.

Acknowledgements

This work is the product of a collective of people which I had the privilege to meet throughout my life. Some of them had direct involvement but most helped me the in subtle but impactful ways. As Dani Carbonell says in a song, “Una gota junto a otra hace oleajes, luego, mares, océanos”.

Special thanks to my advisor Roland Horne for his guidance and patience. He always gave me useful ideas and supported me in trying new ones. I also thank my committee for agreeing to revise my work, question it and giving me useful commentary. Recognition is also due to the Energy Resources Engineering department staff, who were always there to simplify the process and allow me to focus on the academics.

I thank my friends at Stanford, with whom I shared late nights both at the office and the hip neighborhoods of San Francisco. It was a pleasure to learn from you. To my friends in Mexico thank you as well. You always supported me from afar and shaped the path that led to my PhD.

Thank you also to my mom, dad and sister. Without you I would not be the person I am. This is your achievement as well.

To my partner Jordan, I met you as I was starting this journey. Thank you for coming along.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

Understanding the dynamics of subsurface flow systems such as oil and gas reservoirs is of importance during the development and operation of energy extraction projects. Quantifying hydrocarbon reserves is crucial for asset management and portfolio optimization in oil and gas companies. Identifying properties of the fluid and reservoir is necessary when designing a field development plan and ensuring optimal productivity during operations together with well integrity requires constant surveillance of the reservoir. All these applications demand models of the system whether it is a single well or an entire reservoir. Models allow us to make estimates, forecasts and quantify uncertainty so that informed decisions can be made at the right time.

Subsurface flow models come in multiple varieties and they can be classified according to their complexity, scale, objective, or data source. Models of varying spatial or time scale can encompass an entire field of hundreds of square miles or a be devoted to a single well and its drainage area. Models such as reservoir simulators estimate fluid flow years into the future whereas a single well model may focus on a few hours or days. Moreover, model complexity can be measured by the physical interactions being accounted for as well as the required computational cost. Additionally, models can be classified according to their objective. Mechanistic models are useful to understand the drivers of specific behaviors, such as the decrease or increase of oil production in a well. On the other hand, descriptive or predictive models such as production decline curves only try to replicate the behavior without focusing on its mechanics.

Historically, physics-based models have been the most common regardless of the modeling objective. These models use knowledge of the underlying physical principles and governing equations to generate predictions to project unrealized scenarios with quantifiable uncertainty. Because physics-based models encode conservation and constitutive laws they are extremely useful when data are limited and provide unique explanatory power that in the hands of an expert can provide insights into the mechanics of a system. As more complexity has been introduced, nonlinear physics-based models have become very computationally intensive. An example is reservoir simulators, which can take days to run even in high-performance computer clusters. In such problems, systems of highly nonlinear partial differential equations are solved over thousands of grid components to obtain fluid flow predictions [Sun and Zhang, 2020]. The physics may include coupled geomechanical and thermal components on top of the required flow dynamics. At a certain point, the computational cost of solving such a problem may be so high that the solution is no longer actionable for decision making.

Models and data are intrinsically connected. Physics-based models are characterized by uncertain parameters that must be determined from data. In subsurface flow applications, data can be time invariant such as well logs, geologic and seismic surveys or time-varying such as production volumetric flows, downhole or manifold pressure and temperature readings, etc. In recent decades, there has been massive advancements in data collection and streaming capabilities in the oil and gas industry. Remote sensing, cheap data storage, the explosion of machine learning and the declining cost of cloud computing have redefined modeling strategies and increased the number of data-driven model applications. Such models rely on statistical methodologies to capture a mathematical function that models the observed data. This function can be explicit such as in a linear regression or implicit and difficult to extract like a Neural Network. Significant work has been done in data-driven models with applications in geological parameterization, uncertainty quantification, reduced order modeling and well test interpretation [Jin et al., 2020], [Tian and Horne, 2019], [Liu et al., 2020], [Lee et al., 2018]

However, data always come with uncertainty. Noise, heterogeneous sampling, and biases in data collection are present in real life datasets. In the oil and gas industry, compromises are often made in data acquisition due to cost constraints, equipment limitations and conflicting operational objectives. Because of this, purely data-driven models have limitations and the modeling objectives must be specified accordingly. Predictive or descriptive applications are areas where these limitations can be managed properly to achieve reliable and generalizable results.

In the oil and gas industry, field production data are an often-overlooked source of information. Data are usually acquired for surveillance or monitoring purposes and they reflect the complexities of daily operations of a reservoir. Noise is inherent to production data and missing data due to equipment malfunction is commonly found. Because data quality is a crucial element for data-driven models, when using production data significant effort is put into choosing the right portions of a dataset, cleansing, and preprocessing the data to make it useful for modeling. Because of this information loss, it is more difficult to extract knowledge of physical parameters from production data. However, the data are still a product of the underlying physical processes so with the right approach it can still be possible to extract useful information from them.

In this work, the focus was on making use of oil and gas production time-series data to generate predictive and descriptive models of individual well responses both in isolation and with interactions between multiple wells. In [Willcox et al., 2021] three elements of the computational task of learning from data were defined: (1) the representation of the parameter-to-observable map in the learning problem, (2) the optimization of search algorithm, and (3) the nature of the underlying application and the characteristics of its data. This work tackled the three tasks jointly. For the first task, wavelet transforms were used to represent and encode the observable reservoir response from a well. Secondly, machine learning and neural networks were used as the algorithms for searching and obtaining an adequate representation of the mentioned response. Finally, the work explored how specific wavelet transforms can exploit the underlying structure of production data and leverage its apparent flaws to generate a streamlined data-driven methodology for extracting value of field production data.

1.2 Literature Review

The oil and gas industry is heavily data-driven and model-driven. Data-driven models have been used in the industry since its early days. Decline curves are an example of simple predictive data-driven models. In those, oil production is forecasted by choosing from a small pool of basic mathematical models and early production data is used to fit the parameters of the chosen model. In the modern day, the availability of large amounts of historical data has resulted in a proliferation of data-driven models of increasing complexity for a wide array of applications.

Overall, data-driven models can be classified according to their application. Due to the limitation for explaining the mechanics of a process, the applications have been focused around three general areas: production forecasting, production characterization and model identification. In production forecasting the goal is to come up with a predictive model for oil, gas or water flowrate. Production characterization updates estimates of production by analyzing past data. Examples of this are well flow allocation models or oil fraction estimation for a certain production period. Model identification is concerned with discovering unknown reservoir properties or relationships using sensor readings or production data. All these applications make use of a shared pool of methodologies and strategies when assembling data-driven models and have common set of challenges as well.

1.2.1 Data-driven models for production forecasting

Extensive work has been done in the field of production forecasting. There is a wide variability in the scope of the prediction whether in spatial or temporal scale. In [Kubota et al., 2019] the authors used historical data to predict oil rate at a field level for mature assets. The model relied on months of historical data to predict up to 6 months of production. Variables such as the number of producers and injection rates were used to fit the Recurrent Neural Network (RNN) that estimated production. Similarly, in [Aizenberg et al., 2016] a data-driven model was built to predict the monthly oil production of an entire asset in the coastal swamps of the Gulf of Mexico. The authors designed a Multi-Layer Neural Network with Multi-Valued Neuron model (MLMVN) which proved to have good results in multiple months ahead predictions at full field level. [Wei et al., 2022] generated predictions for pressure and saturation fields using a neural network architecture. The methodology required creating a two-dimensional training dataset from point time series data using interpolation via kriging and random forests. Then a ConvLSTM model was used to obtain future pressure distributions and saturation was estimated using relative permeability curves. The data-filling methodology ignored any heterogeneity in the reservoir potentially limiting its reliability. In [Davtyan et al., 2020] the authors also tackled the problem of forecasting oil production for a mature field but included both spatial and temporal variables when designing their model. A set of tailored features were then extracted to build a linear model capable of predicting up to 12 months ahead. The problem of missing data was also explored with the authors choosing a reduced version of the potential input variables due to lack of completeness in the dataset. In

[Yuan et al., 2021] a deep learning model was proposed for prediction of initial production for a field development block. The data used included time-series and static data and required multiple steps of preprocessing. However, missing data limited the applicability of the methodology and the proposed model proved to have high variance in prediction performance. Focusing on a smaller spatial scale, the authors in [de Oliveira Werneck et al., 2022] used a deep-learning, time-series approach to forecast daily oil production and bottom-hole pressure for individual wells using production data. In the work, multiple data preprocessing techniques were tested and heavy feature engineering was deemed necessary to fit the specific datasets.

Because of the lack of complete physical theory, the high availability of data and the relatively short lifespan of wells, unconventional reservoirs have been the focus of a lot of recent work in data-driven models for production forecasting. In [Cao et al., 2016] a neural network model that estimates oil production at a well level was developed for unconventional fields. The authors use tubing pressure estimates and adjacent well histories to predict potential production in undrilled locations. The authors in [Zhan et al., 2019] applied an LSTM model to predict oil rate for individual wells using limited initial production data in an unconventional asset. Their methodology requires careful choice of relevant adjacent wells to use as inputs as well as tubing pressure information for the target well. [Razak et al., 2021] expanded the inputs for a predictive model to include formation and fluid properties, well control data as well as early production history. Their model used a deep learning architecture to predict oil, water, and gas production as multivariate time-series under varying operating controls. They tackled the issue of limited data by applying transfer-learning to a model by training with data of nearby or similar wells. In [Chaikine and Gates, 2021] five-year cumulative production profiles were predicted for multistage hydraulically fractured wells using a combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) units in a deep learning model. However, performance results proved to be poor at individual well level and only being relatively accurate when aggregating results to field level.

The review of the past work in production forecasting revealed a common set of problems modelers need to deal with when trying to design a predictive model. In multiple works, only a small fraction of the total available data was utilized due to heterogeneous sampling and incomplete variables. Many of the proposed models used specifically designed inputs which relied on combinations of raw variables. This amplified the effect that missing data in single variables can have for the overall completeness of a dataset. A lot of the modeling methodologies used different varieties of neural networks to achieve the prediction, whether cases off-the-shelf architectures or purpose-built ones. This showcases the effectiveness of that class of models and the relative ease of use that such models have nowadays.

1.2.2 Data-driven models for model identification

Model identification is by definition an inverse problem. Most applications of this type use measurements to establish properties of the subsurface system such as boundary conditions, fracture presence or rock characteristics. Pressure transient formation and well testing are examples of model identification procedures where data-driven methodologies have historically coexisted with physical models. The objective of well testing is to estimate the productivity of a well and properties of the formation based on pressure and flow-rate measurements. These data are then matched to analytical solutions of fluid flow in porous media under varying flow regimes and reservoir boundary conditions. The analytical solutions contain the desired parameters so significant attention is put in accurately fitting the data through optimization. Contemporary efforts using data-driven methodologies have tried to automate the process and reduce the potential for human error. In [Dong et al., 2022] the authors propose an automated methodology for well test interpretation that identifies the curve type and associated parameters out of four types of well models. The model relies on a Convolutional Neural

Network trained with synthetic labeled cases and is capable of achieving high accuracy in test cases using also synthetic data. [Chu et al., 2019] compared the performance of a Fully-Connected Neural Network (FCNN) and a CNN for well test curve classification. The authors concluded that the CNN had a more robust performance in the classification results. In [Tian and Horne, 2015] the authors applied feature based machine learning to interpret pressure downhole gauge (PDG) data. Linear and Kernel Ridge Regression models were tested and they achieved success in deconvolving the data and reconstruct both pressure and flowrate histories. Later in [Tian and Horne, 2017] two RNN structures were used to learn the pressure response of a well from PDG data. The tested structures were a Nonlinear Autoregressive Exogenous Model (NARX) and a standard RNN. The model results helped correctly identify the reservoir model and accurately forecast the reservoir performance. [Tian and Horne, 2016] used machine learning based multi-well testing learn the correlation between producer rate features and injector pressures. The methodology was able to capture the injectors' influence on producers thus determining well connectivity.

1.2.3 Data-driven models for production characterization

The problem of production characterization deals with understanding past behavior of a reservoir. Often the desired variable is derived from incomplete measurements or proxies that require a model to translate the readings. Examples of this are well rate allocation in commingled structures and fluid determination in multiphase flow wells. These problems are well suited for data-driven modeling since historical is often available and discovering physical processes or properties is not the main objective. In [Alakeely and Horne, 2021] the authors tackle the well liquid rate allocation problem by building a deep-learning proxy model to estimate well liquid and multiphase flow rate in different choke conditions using surface measurements as model inputs. The results have a superior performance when with standard empirical methods such as Gilbert correlation. In [Kim and Durlofsky, 2021] bottomhole pressure sequences were used as input for estimating past water and oil rates at a well level. The model used a RNN proxy model that achieved close agreement with numerical simulation thus allowing the results to be used for well control optimization. [Li et al., 2021] tackled the problem of missing segments in well history data. A series of models including decision trees, boosting and random forest regressions were set to recover up to 30% of the missing dataset. The authors found that adaptive boosting and bagging gave the best performance, but a rich dataset was required for the methodology to work accurately.

1.2.4 Wavelet applications in subsurface flow literature

The use of wavelets has a long history in the oil industry. Its origin can be traced to the work of [Morlet et al., 1982a] on seismic signal decomposition. During the 1990's wavelets were applied to pressure data mainly for denoising and compression. For denoising, multiple methodologies were developed mostly based on thresholding with a significant amount of work devoted to choosing the appropriate wavelet, level of decomposition and threshold for different types of datasets. [Bernasconi et al., 1999] designed a wavelet-based lossy data compression algorithm for downhole drill bit data including pressure, torque and accelerations. [Kikani et al., 1998] developed a methodology for denoising pressure transient data by applying the wavelet transform and thresholding the resulting high frequency coefficients. However, in that methodology the threshold was chosen by dropping coefficients until the signal passed a visual inspection. [Ribeiro et al., 2008] explored different wavelet basis and decomposition levels to identify which one is more suitable for denoising and outlier removal in pressure transient data. No universally applicable wavelet basis was identified but it was found that the Daubechies family tended to generalize better for pressure transient signals. [Athichanagorn et al., 1999] proposed an encompassing methodology for processing and analysis of long-term pressure data. This methodology covers the full spectrum of applications provided the basis for much of the later work. The first three steps are focused on increasing the quality of the data and make ample use of the wavelet transform. The later steps do not involve the use of wavelets explicitly but depend on the completion of first steps to be successful. Keeping with the tradition of previous work, this study built upon these steps to tackle the long-term pressure transient problem. The steps are:

1. Outlier removal – The goal of the step is to remove aberrant data points that are not part of the signal. This stage is done by iteratively eliminating detail coefficients by using a shrinking threshold. The data is analyzed in a reversed-time fashion because changes in pressure are lower at later times, making it easier to identify outliers.

2. Noise removal - The purpose of this stage is to smoothen the data and extract the most representative features of it. In this application, wavelets are particularly useful when compared to traditional filters because of their multiresolution properties which preserve most of the sharp features of the data. [Athichanagorn et al., 1999] proposed a wavelet shrinkage strategy where the detail signals are shrunk toward zero, effectively reducing noise. The strategy uses a hybrid threshold at this step meaning that a soft thresholding criterion is applied in continuous (low variability) data regions and a hard thresholding criterion is used near discontinuities. In contrast, [Kikani et al., 1998] exclusively used soft thresholding for the denoising step. However, both approaches require knowledge of the noise level of the pressure and flow-rate signal, which can be unknown in practice. To solve this problem, [Ouyang et al., 2002] proposed to fit a logarithmic function using the polytope method to determine the noise level in pressure data and making it easier to set the threshold when the noise magnitude is unknown.

3. Transient identification – The goal of this step is to identify potentially unknown flow-rate changes based on sudden changes in pressure data. For this, [Athichanagorn et al., 1999]

proposed the use of the wavelet modulus maxima. The intuition behind this is that at intermediate levels of decomposition, the noise singularities disappear while the signal singularities are still present. Then, after a decomposition level and a threshold is chosen, the starting positions of new transients can be identified by picking those detail signals with wavelet modulus maxima higher than the threshold. [Ouyang et al., 2002] later proposed to use pressure ratios to estimate how the predictability of individual transients when using the methodology developed by Athichanagorn.

4. Flow history reconstruction – It is not uncommon for production data to have incomplete flow rate history or uncertainty in the flow rate measurement. In this step, the methodology developed by Athichanagorn does not make explicit use of the wavelet transform and instead formulated the problem of filling flow-rate gaps as a nonlinear regression with flow-rate being as the unknown parameter. For this step, it is key that the previous three steps have been executed successfully.

5. Behavioral filtering – The goal of this step is to identify inconsistent or aberrant pressure transients that lead to an incorrect reservoir parameter estimation. These anomalies might be caused by failures in the pressure gauge during sudden changes and thus should be excluded from analysis. [Athichanagorn et al., 1999] used the variance between a regression match and the data as a measure to find aberrant transients. By iteratively eliminating transients with the maximum variance and refitting a regression on pressure, the approach reduces the error in model parameter estimates. In machine-learning based modeling, this step is referred as concept drift.

6. Data interpretation – This step is focused in analyzing the long-term data concurrently by using a moving window of analysis. This was done to accommodate the possibility of reservoir parameters changing during the data acquisition period. The proposed window should contain

a few transients and the subsample of the data is then used to run a regression to determine the local values of reservoir parameters.

The application of wavelet transforms demonstrated potential for modeling characteristic behaviors in long-term pressure transient data at varying scales. These can be time-localized events such as flow rate changes or long-term effects such as pressure depletion or combinations of short and long-term transients. However, as it is evident from the seven steps described previously, the task of analyzing production pressure data was heavily fragmented and multiple parameters had to be sequentially chosen to come up with a good understanding of the data. This sequential nature of the operation poses a challenge as errors made in the early steps can propagate to the later stages. Moreover, the need or identifying breakpoints for transient identification adds to the challenges, as breakpoints in real data tend to be slow changes and not sharp transitions. In addition to that, setting up thresholds for outlier detection and denoising makes it necessary to know the noise magnitude, which is unknown in real data.

1.3 Summary and Research Objectives

The oil and gas industry has a rich history of data-driven models. The recent availability of large scale datasets as well as the advancements in the field of artificial intelligence have allowed for the improvement of models for forecasting, model identification and production characterization. For model identification at the well level, multiple challenges remain including the need for data cleansing and preprocessing, design and selection of model input features that capture the relevant physics, as well as the limited ability of models to deal with missing data. This work aimed to tackle those challenges by leveraging the properties of wavelets and making use of advanced machine learning models. Specifically, the objectives of the conducted research were:

1. Design an integrated framework for modeling a well's response using long term pressure and

flow rate production data.

2. Build a fully data-driven methodology that captures the physical processes within a well.

3. Deal with noise and data imperfections as an inherent property of the data.

4. Create an automated process for designing model input features.

5. Allow for the building of models with incomplete data without loss of performance.

1.4 Dissertation Outline

This work is organized as follows:

Chapter 2 introduces the fundamentals of wavelets, wavelet filters and transforms as well as their applications to data driven modeling of flow rate and pressure data. The chapter outlines a series of structural requirements and constraints that guide the choice of a wavelet transform that can be integrated with machine learning models. The Maximal Overlap Discrete Wavelet Transform (MODWT) is introduced as well as the concept of a Multiresolution Analysis (MRA), specifically the MODWT-MRA. Finally, the implications of applying the MODWT-MRA to flow rate and pressure data are discussed in the context of its application to data-driven models.

Chapter 3 develops a methodology to combine the MODWT-MRA with machine learning and deep learning methods to model pressure response from flow rate history data. Two modeling methodologies are introduced, the Lasso and LassoNet and relevant theory is presented for such methodologies. Afterwards, the developed methodology is applied to both one-well and two-well scenarios. For each scenario, the effects of missing data gaps are explored as well.

Chapter 4 applies the proposed wavelet and machine learning methodologies to the problem of flow rate reconstruction. As in Chapter 3, the one-well and two-well scenarios are explored. The effects of missing data in the modeling process are also studied.

Chapter 5 contains the summary and conclusions of this work. The main ideas are revisited and a list of additional applications for the developed methodology are explored.

Chapter 2

Wavelet Transforms for Data-Driven Modeling

Modeling reservoir response through data-driven methods requires a set of input variables to codify the necessary information to obtain a reservoir's response from available data such as oil or liquids flow rate and make estimates of future behavior or fill in data gaps. This set of input variables, also referred to as features are fundamental in determining a model's performance and ability to capture complex behavior. Defining features to capture reservoir response can be a challenge due to the complexity of the physical process, changes in flow regimes or operation modes. Moreover, in a reservoir there is often a superposition of processes happening such as noise, well interference, water injection pressure support, etc. This makes it more difficult for the modeler to choose features but also can inform potential choices for model input variables. A useful set of features is one that allows the modeling technique to extract maximum information with minimum effort. This means preserving relevant information such as short term localized events as well as longer duration effects that might be present in data. Wavelet transforms have been shown to have some of those desirable properties, which make them an ideal candidate for using in the design of useful features for modeling reservoir response. This chapter showcases some of the properties of wavelet transforms and defines the relevant mathematics that give rise to their usefulness for feature building in data-driven models.

2.1 Introduction

Wavelet Transforms are part of a larger set of time-frequency transforms capable of decomposing and analyzing the structure of a time series. The Fourier Transform is the most popular of this longer set of transforms as it provides a way of decomposing data into a frequency-by-frequency basis. For discrete data, the Discrete Fourier Transform (DFT) provides the same representation of the data in the frequency domain. However, both the Fourier Transform and DFT do not preserve information in time, meaning that the time location of events such as abrupt changes is lost after applying the transform. Moreover, Fourier transforms require a time series to be stationary, which implies that all the identified frequencies must be present throughout the entire duration of the data series. This is a very strong assumption when for real data such as oil production, where operational changes and flow regimes are not necessarily present or constant during the entire data acquisition period.

To overcome the challenge of preserving time information, the Short-Time Fourier Transform (STFT) was developed [Allen, 1977]. The STFT takes a sliding window of the data and applies the Fourier transform to the window. By doing this, specific frequencies can be assigned to individual windows of time with a resolution equal to the width of the window. However, the STFT is not perfect since events happening within a specific window cannot be localized with higher time resolution. A visual way to observe this and compare the Fourier Transform with the STFT is to plot the time-frequency plane, as shown in Figure 2.1.

The Wavelet Transform [Morlet et al., 1982c],[Morlet et al., 1982b] proposed a better way of preserving both frequency and time information. It overcomes the fixed time-frequency resolution of the STFT by using a different basis function called the mother wavelet. This function is then dilated and translated to capture features local in time and frequency. This allows for an adaptive time-frequency partition that becomes long in time when low frequency behavior is present and short in time when high frequency events exist. Because of this, a more adequate frequency and time resolution is achieved in the transform as shown in Figure 2.1(d). The Wavelet Transform is a function of scale, as opposed to frequency like the Fourier Transform. Intuitively, scale is similar to frequency but inversely related in the sense that an increase in scale also increases the time support of the basis function thus reducing the frequency resolution of high frequencies. Inversely, a decrease in scale reduces the time support and increases the resolution of higher frequencies. Section 2.2 will formally define the wavelet transform in both continuous and discrete form and use it to derive other wavelet transforms such as the Maximum Overlap Wavelet Transform (MODWT) which can be useful to create inputs for data driven models.

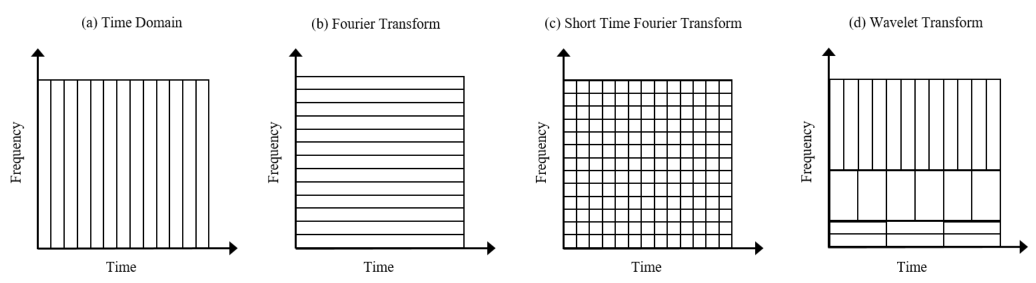

Figure 2.1: Partitioning of the time-frequency plane

2.2 Continuous and Discrete Wavelet Transforms

Similarly to the Fourier Transform, the Wavelet Transform is defined in both continuous and discrete form. In this section the Continuous Wavelet Transform (CWT) will be defined first followed by the Discrete Wavelet Transform (DWT), which can be thought as a discretized version of the DWT.

2.2.1 Wavelet Functions

Before defining the Wavelet Transform, it is necessary to specify the basis functions. As sine and cosines are the basis functions for the Fourier transform, the wavelet function forms the basis for all wavelet transforms. There are multiple types of wavelet functions but all of them share a set of defining characteristics. Loosely defined, a wavelet is a wave-like function that has a beginning and an end in time, similar to a wave packet. Formally, a wavelet $\psi(t)$ is a function of time t that satisfies the so-called admissibility condition [Mallat and Mallat, 1999]:

$$C_\psi = \int_0^\infty \frac{|\Psi(f)|}{f} df < \infty \tag{2.1}$$

where $\Psi(f)$ is the Fourier Transform of the wavelet function $\psi(t)$, and is a function of frequency f. For the admissibility condition to be met, $\Psi(f)$ must decline quickly as $f \to 0$. To guarantee that $C_\psi > 0$ then $\Psi(0) = 0$ which translates to [Gençay et al., 2002]:

$$\int_{-\infty}^\infty \psi(t)dt = 0 \tag{2.2}$$

Equation 2.2 implies that the wavelet function has a zero mean. Additionally, a wavelet function must satisfy the condition of unit energy:

$$\int_{-\infty}^\infty |\psi(t)|^2 dt = 1 \tag{2.3}$$

Once a valid wavelet function has been defined, the wavelet transform simply projects another function $x(t)$ onto the wavelet space by applying the convolution:

$$W(u,s) = \int_{-\infty}^\infty x(t)\psi_{u,s}(t)dt \tag{2.4}$$

where:

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right) \tag{2.5}$$

The resulting wavelet transform $W(u, s)$ is known as the Continuous Wavelet Transform (CWT) and is a function of two variables representing translation u and dilation s. In a similar form, the inverse transform can be defined by:

$$x(t) = \frac{1}{C_\psi}\int_0^\infty \int_{-\infty}^\infty W(u, s)\psi_{u,s}(t)du\frac{ds}{s^2} \tag{2.6}$$

Applying the wavelet transform to a function or data (Equation 2.4) is known as analyzing or decomposing. Conversely, applying the inverse transform (Equation 2.6) is known as synthesizing or reconstructing. The CWT's parameters u and s can take an infinite number of values and therefore contain a large amount of redundant information. In practice, only a limited number of parameters u and s is needed to capture all information present in the original function. The minimum number of wavelet coefficients needed to preserve all the original information is known as a "critical sampling" of the CWT and is what defines the Discrete Wavelet Transform (DWT). This critical sampling is obtained by discretizing s and u:

$$s = 2^{-j}, u = k2^{-j} \tag{2.7}$$

where j and k define the set of discrete translations and dilations defining the DWT. Intuitively, the CWT exists for all values in the time-frequency space, whereas the DWT only exists at certain points in that space defined by:

$$\psi_{j,k}(t) = 2^{j/2}\psi\left(2^j t - k\right) \tag{2.8}$$

2.2.2 Wavelet Filters

Wavelet filters are the discrete equivalent of wavelet functions. The same properties of integration to zero and unit energy for wavelet functions ψ described in Equations 2.1 through 2.3 apply to wavelet filters. For a filter h of length L, the zero integration condition becomes zero sum condition:

$$\sum_{l=0}^{L-1} h_l = 0 \tag{2.9}$$

and the unit energy condition can be expressed as:

$$\sum_{l=0}^{L-1} h_l^2 = 1 \tag{2.10}$$

The critical sampling condition requires the wavelet filter h_l to be orthogonal to its even shifts [Gençay et al., 2002]:

$$\sum_{l=0}^{L-1} h_l h_{l+2n} = 0, \quad \text{for all integers} \quad n > 0 \tag{2.11}$$

When dealing with filters it is useful to analyze them with respect to their frequency response. This characterizes the way different frequencies in a signal get filtered when the filter is applied. The frequency response function, also known as transfer function is defined as:

$$H(f) = \sum_{k=-\infty}^{\infty} w_k e^{-i2\pi fk} \tag{2.12}$$

where $i = \sqrt{-1}$, f is the frequency and w_k is the impulse response function of a filter. The impulse response function is a function of time and denotes the filter's response to the unit impulse signal:

$$x_t = \begin{cases} 1 & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

The frequency response function in Equation 2.12 can be rewritten as:

$$H(f) = G(f)e^{-i\theta(f)} \tag{2.14}$$

where $G(f)$ is called the gain function and it denotes the magnitude of the frequency response function: $|H(f)| = G(f)$ and θ is the phase angle of the filter.

Wavelet filters are high-pass filters, meaning that their gain function decreases monotonically as $f \to 0$. For the transform to cover the entire frequency spectrum, a low-pass or scaling filter is necessary. For most wavelet filters, the low-pass filter coefficients can be obtained from the high-pass ones by applying the quadrature mirror relationship:

$$g_l = (-1)^{l+1} h_{L-1-l} \quad \text{for} \quad l = 0, ..., L-1 \tag{2.15}$$

The simplest wavelet filter is the Haar wavelet [Haar, 1910] which is a filter of length $L = 2$ and its wavelet filter coefficients h and scaling coefficients g are defined as:

$$h_0 = \frac{1}{\sqrt{2}}, \quad h_1 = -\frac{-1}{\sqrt(2)} \tag{2.16}$$

$$g_0 = g_1 = \frac{1}{\sqrt(2)} \tag{2.17}$$

Figure 2.2 shows the wavelet and scaling functions in the time domain as well as the squared frequency response $|G(f)|^2$ of the corresponding filters. It can be seen that the wavelet filter high-pass as $|G(f)|^2 \to 0$ as $f \to 0$. Conversely, the scaling filter is low-pass with $|G(f)|^2 \to 0$ as $f \to 1/2$

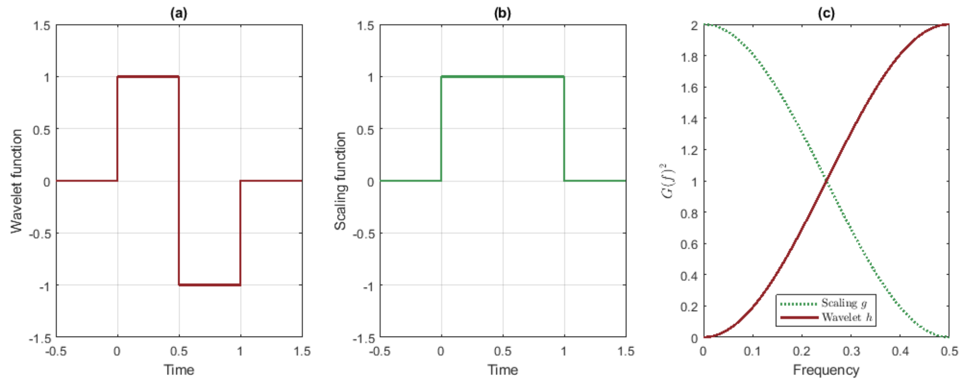

Figure 2.2: (a) Haar wavelet function. (b) Haar scaling function. (c) Squared frequency response function for the Haar wavelet and scaling filters

2.2.3 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) was defined previously by Equations 2.4, 2.5, 2.7 as a discretized version of the CWT. However, it can also be defined by using wavelet filters. The projection of the data over translations and dilations of the wavelet function is equivalent to a matrix operation executing the convolution of the data with a series of wavelet filters. These filters are associated with specific scales and as such can capture changes within those scales.

Assuming a time series $\boldsymbol{x}$ of length $N = 2^J$, we can obtain the wavelet coefficients $\boldsymbol{w}$ by applying the DWT operator W:

$$\boldsymbol{w} = W\boldsymbol{x} \tag{2.18}$$

where W is an $N \times N$ is an orthonormal matrix. The resulting coefficients $\boldsymbol{x}$ can be decomposed into $J+1$ vectors:

$$\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ...\boldsymbol{w}_J, \boldsymbol{v}_J]^T \tag{2.19}$$

where $\boldsymbol{w}_j$ is a vector of wavelet coefficients of length $N/2^j$ and $\boldsymbol{v}_J$ is a vector of length $N/2^J$ of scaling coefficients. The wavelet transform matrix W is composed by rows of the form:

$$\boldsymbol{h}_1 = [h_{1,N-1}, h_{1,N-2}, ..., h_{1,1}, h_{1,0}]^T \tag{2.20}$$

where $\boldsymbol{h}_1, 0, ...h_{1,L}$ are the filter coefficients for a wavelet of length L and all values $L < t < N$ are zero. By circularly shifting h_1 by factors of two, the next rows of the matrix W are obtained:

$$\boldsymbol{h}_1^{(2)} = [h_{1,1}, h_{1,0}, h_{1,N-1}, h_{1,N-2}, ..., h_{1,3}, h_{1,2}]^T \tag{2.21}$$

$$\boldsymbol{h}_1^{(4)} = [h_{1,3}, ..., h_{1,0}, h_{1,N-1}, h_{1,N-2}, ..., h_{1,5}, h_{1,4}]^T \tag{2.22}$$

Then, the matrix W_1 as a $N/2 \times N$ sized matrix of the circularly shifted versions of h_1 in the form:

$$W_1 = \left[\boldsymbol{h}_1^{(2)}, \boldsymbol{h}_1^{(4)}, ..., \boldsymbol{h}_1^{(N/2-1)}, \boldsymbol{h}_1\right]^T \tag{2.23}$$

In a similar fashion, we can define h_2 as the zero-padded scale 2 wavelet filter coefficients in the same way as Equation 2.20. Then W_2 can be constructed in a similar way as Equation 2.23 but shifting the vector h_2 by a factor of 4. Consecutively, one can construct W_j in the same way by shifting h_j by a factor of 2^j. The matrix $\mathcal{V}$ is a column vector with all elements being $1/\sqrt{N}$. Finally, the $N \times N$ dimensional matrix W is built as:

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_j \\ \mathcal{V}_j \end{bmatrix} \tag{2.24}$$

The practical implementation of the DWT follows the so-called pyramid algorithm [Mallat and Mallat, 1999]. Stated simply, the pyramid algorithm recursively applies the wavelet and scaling filters h_1 and g_1 to the data and subsamples the output of the filter to half its original length at each iteration. Therefore, the coefficients $\boldsymbol{w}_{j+1}$ are half the length of $\boldsymbol{w}_j$ which is known as decimation.

2.3 Maximal Overlap Discrete Wavelet Transform

The definition of the DWT in Equation 2.18 assumed a dataset length $N = 2^J$. For many applications, this length requirement cannot be met or is unrealistic. Another important limiting factor of the DWT is the length decimation of the wavelet coefficients. Because the DWT represents the critical sampling of the CWT, the resulting coefficients $\boldsymbol{w}_j$ decrease in length with increasing scale level j. To address this issues, an alternative to the DWT applicable to any sample size N and with no decimation was developed in the form of the Maximum Overlap Discrete Wavelet Transform (MODWT) [Percival and Walden, 2000].

The MODWT has the following defining properties:

1. Defined for any sample size N, as opposed to the DWT which requires $N = 2^J$.

2. No coefficient decimation with increasing j. Coefficients w_j for the MODWT are of length

N independently of level j. Because it is not a critical sampling, the MODWT does contain reduntant information.

3. The coefficients of the MODWT multiresolution analysis are zero-phase. This implies that events in the original series are captured in the MODWT multiresolution analysis at the same time point as in the original series (aligned).

4. Invariant to circular shifting or translation of the data. This property means that the shifting the input x by a k time steps results in a shift in the MODWT coefficients of the same k steps. This is not a property of the DWT where a shift in the input data results in different wavelet coefficients.

The MODWT is defined as:

$$\widetilde{\boldsymbol{w}} = \widetilde{W}\boldsymbol{x} \tag{2.25}$$

where $\widetilde{W}$ is a $(J{+}1)N{\times}N$ matrix defining the MODWT decomposition. The matrix of coefficients $\widetilde{\boldsymbol{w}}$ is composed by $J+1$ vectors in the way:

$$\widetilde{\boldsymbol{w}} = [\widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2, ..., \widetilde{\boldsymbol{w}}_J, \widetilde{\boldsymbol{v}}_J]^T \tag{2.26}$$

where $\widetilde{\boldsymbol{w}}_j$ is of length $N/2^j$. The wavelet coefficients $\widetilde{\boldsymbol{w}}_j$ is associated with changes on a scale of length $\lambda_j = 2^{j-1}$. The vector $\widetilde{\boldsymbol{v}}_J$ is also of length $N/2^J$ and contains the scaling coefficients that capture changes on a scale of length $2^J = 2\lambda_J$.

The matrix $\widetilde{W}$ then defines the MODWT decomposition is made of $J+1$ submatrices of $N \times N$ size, so that:

$$\widetilde{W} = \begin{bmatrix} \widetilde{W}_1 \\ \widetilde{W}_2 \\ \vdots \\ \widetilde{W}_J \\ \widetilde{\mathcal{V}}_J \end{bmatrix} \tag{2.27}$$

The wavelet and scaling filter coefficients of the MODWT are rescaled filters of the DWT:

$$\widetilde{\boldsymbol{h}}_j = \boldsymbol{h}_j/2^j \quad \widetilde{\boldsymbol{g}}_J = \boldsymbol{g}_J/2^j \tag{2.28}$$

The $N \times N$ submatrix $\widetilde{W}$ is built by shifting to the right the filter coefficients $\widetilde{\boldsymbol{h}}_1$ by integer units:

$$\widetilde{W}_1 = \left[\widetilde{\boldsymbol{h}}_1^{(1)}, \widetilde{\boldsymbol{h}}_1^{(2)}, ..., \widetilde{\boldsymbol{h}}_1^{(N-2)}, \widetilde{\boldsymbol{h}}_1^{(N-1)}, \widetilde{\boldsymbol{h}}_1\right]^T \tag{2.29}$$

In a similar fashion to Equation 2.29, the submatrices $\widetilde{W}_2, ..., \widetilde{W}_J$ are formed replacing $\widetilde{\boldsymbol{h}}_1$ with $\widetilde{\boldsymbol{h}}_j$. This matrices can be interpreted as interweaving versions of the DWT equivalent matrices W_j.

2.3.1 MODWT Multiresolution Analysis (MODWT-MRA)

So far, the DWT and MODWT have been introduced as wavelet transforms applicable to real discrete datasets. In the process, the DWT proved to be unsuitable for datasets of nondiadic length and coefficient length decimation makes it difficult to use of the DWT coefficients for further modeling. The MODWT overcomes those challenges but the resulting coefficients are still not directly interpretable in a physical sense because they exist in the scale domain. To use the output of wavelet transforms for modeling purposes, it is desirable to have the obtained decomposition in the same domain as the original data. Some wavelet transforms, including the DWT and MODWT allow for the construction of a multiresolution analysis (MRA) which can be thought as an approximation of a function or dataset at different resolutions in the original domain of the function.

To build the MODWT-MRA, we can represent the t element of a data series x as a linear combination of the MODWT wavelet coefficients:

$$x_t = \sum_{j=1}^{J+1} \widetilde{d}_{j,t}, \quad t = 0, ..., N-1 \tag{2.30}$$

where $\widetilde{d}_{j,t}$ is the element corresponding to time t of the MODWT inverse transform of the jth level coefficients $\widetilde{\boldsymbol{w}}_j$:

$$\widetilde{\boldsymbol{d}}_j = \widetilde{W}_j^T \widetilde{\boldsymbol{w}}_j \quad \text{for} \quad j = 1, ..., J \tag{2.31}$$

and the $J+1$ component of the MODWT-MRA is then $\widetilde{d}_{J+1} = \widetilde{\mathcal{V}}_J^T V_J$. It must be noted that the wavelet coefficients $\widetilde{\boldsymbol{w}}_j$ represent the wavelet decomposition corresponding to scale $\lambda_j = 2^j - 1$ so conversely, the MODWT-MRA components $\widetilde{d}_j = \widetilde{W}_j^T \widetilde{\boldsymbol{w}}_j$ contain the portion of the original data corresponding to scale λ_j. Scale λ in this context is relative to the sampling rate of the data.

Combining Equations 2.25 and 2.31 we can see that the j component of the MODWT-MRA is the result of the application of the inverse MODWT to the jth component of the MODWT:

$$\widetilde{\boldsymbol{d}}_j = \widetilde{W}_j^T \widetilde{W}_j \boldsymbol{x} \tag{2.32}$$

The described properties of the MODWT-MRA make it an ideal candidate for decomposing oil production data for use in data-driven modeling. The ability to capture events at multiple scales, lack of decimation and zero-phase lag make it straightforward to use as input for machine learning models.

2.4 MODWT-MRA Applied to Flow Rate and Pressure Data

Section 2.3.1 defined the MODWT-MRA as applied to any function or dataset. This section expands on those definitions as applied to the specific case of flow rate and pressure data from an individual well. Pressure and flow rate are related to each other through the pressure transient equation as derived using the Duhamel superposition theorem [Kuchuk et al., 2010]:

$$\begin{aligned} p(t) &= p_0 - \int_0^t \frac{dq}{d\tau}(\tau)\, p_u(t-\tau)\, d\tau \\ &= p_0 - \int_0^t q(t-\tau)\, \frac{dp_u}{d\tau}\, d\tau \end{aligned} \tag{2.33}$$

where p_u is the unit-rate pressure response of the system when $q = 1$. Equation 2.33 expresses the transient pressure as a convolution of the derivative of the flow rate and unit-rate pressure response. Also it can be read as the convolution between the flow rate and the derivative of the unit-rate pressure response. If we define the unit-rate impulse response as the time-derivative of the unit-flow rate response:

$$\frac{dp_u}{dt} \equiv g(t) \tag{2.34}$$

Then the integral given by Equation 2.33 can express pressure as the output of a system to the flow rate input:

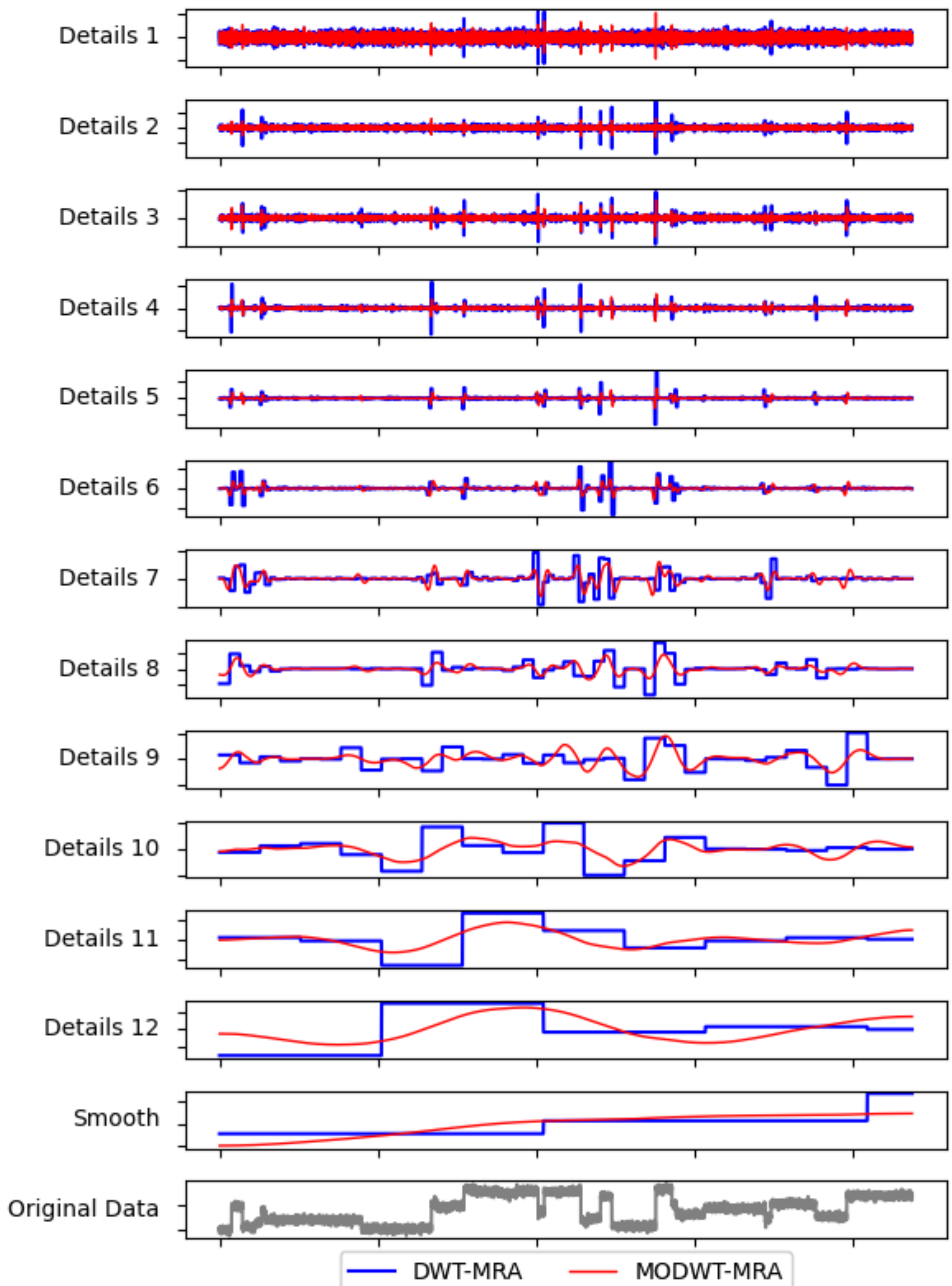

Figure 2.3: Comparison between the MODWT-MRA and the DWT-MRA. Because of decimation, the DWT-MRA shows longer step changes at higher levels of decomposition.

$$\begin{aligned} p(\mathbf{r},t) &= p_0 - \int_0^t q_{sf}\; g(r,t-\tau)\; d\tau \\ &= p_0 - q_{sf}(\tau)\; *\; g(\mathbf{r},t) \end{aligned} \tag{2.35}$$

In the Laplace domain, Equations 2.35 becomes:

$$\overline{p}(\mathbf{r},s) = \frac{p_0}{s} - \overline{q}_{sf}(s)\; \overline{g}(\mathbf{r},s) \tag{2.36}$$

Applying the jth level MODWT-MRA operator $\widetilde{W}_j$ in the Laplace domain is equivalent to multiplying by the Laplace transform of the filter $\widetilde{\boldsymbol{h}}_j$: $\mathcal{L}\{\widetilde{\boldsymbol{h}}_j\} = \overline{\mathbf{h}}_j$. Then the MODWT-MRA in the Laplace domain becomes:

$$\overline{\boldsymbol{d}}_j = \overline{\mathbf{h}}_j^*\, \overline{\mathbf{h}}_j \boldsymbol{x} \tag{2.37}$$

Combining Equations 2.36 and 2.37 the pressure response at scale j can be obtained in the Laplace domain by simply multiplying the equation by $\overline{\mathbf{h}}_j$ and its inverse $\overline{\mathbf{h}}_j^*$:

$$\overline{p}_j(\mathbf{r},s) = \overline{\mathbf{h}}_j^*\, \overline{\mathbf{h}}_j \frac{p_0}{s} \;-\; \overline{\mathbf{h}}_j^*\, \overline{\mathbf{h}}_j\; \overline{q}_{sf}\; \overline{g}(\mathbf{r},s) \tag{2.38}$$

In Equation 2.38, it must be noted that $\overline{q}_{sf(j)} = \overline{\mathbf{h}}_j^*\, \overline{\mathbf{h}}_j\; \overline{q}_{sf}$ is the Laplace transform of the jth MODWT-MRA level of flow rate. Therefore, Equation 2.38 establishes that the pressure response at scale λ_j is the output of the system characterized by $\overline{g}$ when the input is the flow rate MODWT-MRA component at scale λ_j. Therefore, Equation 2.38 can be rewritten as:

$$\overline{p}_j(\mathbf{r},s) = \overline{\mathbf{h}}_j^*\, \overline{\mathbf{h}}_j \frac{p_0}{s} \;-\; \overline{q}_{sf(j)}\; \overline{g}(\mathbf{r},s) \tag{2.39}$$

2.4.1 Virtual Wells

The implications of Equation 2.38 for a well production dataset containing pressure and flow rate time series is significant. Applying the MODWT-MRA to both pressure and flow rate data is mathematically analogous to decomposing the data coming from a single well into a set of "virtual wells" existing simultaneously in time and spatial location (Figure 2.4). The additive nature of the MODWT-MRA allows for the flow rate decomposition to be mass preserving because the original total volume is always obtained when adding the derived MODWT-MRA flow rate components. Similarly, a pressure MODWT-MRA decomposition follows the principle of superposition as the sum of the pressures "felt" by virtual wells are the actual pressure readings at the sensor location. These specific properties of the MODWT-MRA can be particularly useful when applying the decomposition to generate input features for a data driven model of reservoir response. Because of the preservation of the physical behavior within each decomposed flow rate-pressure data pairs, the MODWT-MRA becomes a data enhancement methodology, where information is not lost but instead potentially complex behavior is split into simpler components. This has the effect of reducing the complexity of the function space where a data-driven model would have to operate while learning the reservoir response model.

2.4.2 MODWT-MRA Parameters

To apply the MODWT-MRA to flow rate data, its necessary to specify the maximum scale level J of the transform. This determines the "depth" of the decomposition and the scale and duration of the events captured at each decomposition level. The maximum possible scale level J depends on the length of the data. For a time series of length $N = 2^J$, the maximum number of levels in the decomposition is J. However, for high frequency long-term well production data, N can be very large and because of it J can also be large. In practice, a level $J_0 < J$ is usually sufficient. A useful

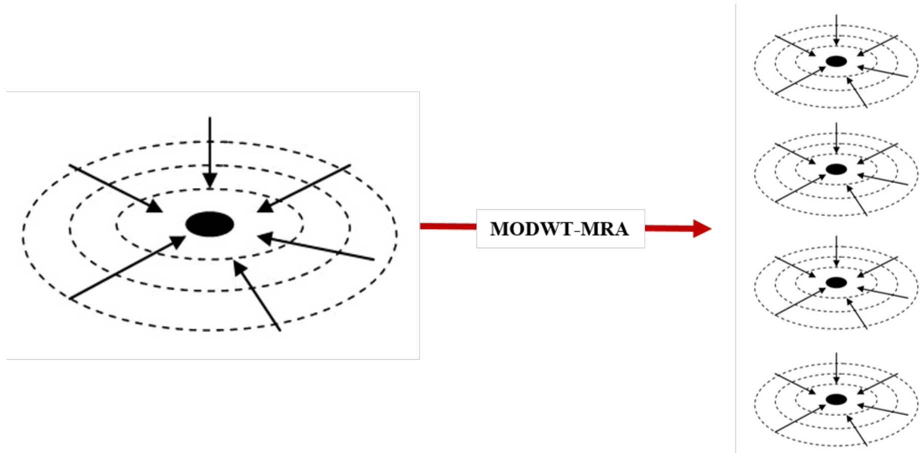

Figure 2.4: The MODWT-MRA decomposes flow rate and pressure data into analogous “virtual wells”

heuristic is to have $J_0 \leq \log_2(N)$ so the decomposition does not consider scales longer than the time series itself. A more restrictive condition requires that L_{J_0}, the length of the filter at the J_0th level, is less than the sample size. This condition $L_{J_0} < N$ which then guarantees the alignment of the MODWT wavelet coefficients with the original series and prevents “wrappings” of the time-series at level J_0. [Cornish et al., 2006]

Nevertheless, when applying the condition $L_{J_0} < N$, the resulting J_0 might still be too large for practical applications. A factor that must be considered when applying the MODWT or MODWT-MRA for data-driven modeling, is the length of the datasets used for validation and testing of the resulting model. It is common practice that the validation and testing datasets are of shorter length than the training or fitting dataset. Therefore, an updated condition $L_{J_0} < N_{test}$ must be applied so that the training, validation and testing datasets contain the same number of variables.

The conditions previously described set an upper limit to J_0. However, the maximum possible J_0 might not be the optimal number for every application or dataset. Ultimately, practical considerations and specific knowledge of the processes taking place in the reservoir and their respective duration can also be used to set J_0. The MODWT-MRA splits the data into frequency bands of the form:

$$\left[\frac{1}{2^{j+1}\ \Delta t}\ ,\ \frac{1}{2^{j}\ \Delta t}\right]$$

where Δt is the sampling period. The scaling or smooth component captures lowest frequencies in the band:

$$\left[0\ ,\ \frac{1}{2^{J+1}\ \Delta t}\right]$$

Conversely, the frequency bands correspond to periods of different duration as shown in Table 2.1. This implies that events of limited duration will be contained to the corresponding jth level of the decomposition. It must be noted that as J_0 becomes smaller, the scaling or low frequency band becomes wider but the resolution at the highest frequencies where $j = 1, 2, ...j << J_0$ keeps unchanged. This becomes relevant when data contains noise, as noise is by definition high-frequency. Therefore, regardless of the choice of J_0, noise will always be contained at the lowest j levels.

The wavelet filter is another parameter that needs to be chosen before the decomposition is applied to the data. The wavelet filter is the basis for the decomposition and it partly determines the quality of the decomposition. Because the wavelet function is used to represent the behavior of the data, it helps when the wavelet function mimics or somewhat resembles the behavior of the data. Common types of wavelet filters are the Haar and Daubechies filters. An example of this is

Table 2.1: Period length for each level of the MODWT where Δt corresponds to the sampling period

Level j	**Lower End**	**Upper End**
1	$2\Delta t$	$4\Delta t$
2	$4\Delta t$	$8\Delta t$
3	$8\Delta t$	$16\Delta t$
$\vdots$	$\vdots$	$\vdots$
J	$2^{J}\Delta t$	$2^{J+1}\Delta t$
J (Scaling)	$2^{J+1}\Delta t$	∞

flow rate data, which can be thought of a superposition of piece-wise constant functions. Because of this the Haar wavelet is a reasonable choice to use as a basis function for the decomposition in this particular application.

2.5 Summary

Data-driven models often require designing input variables that capture the relevant behavior of a reservoir while filtering out the noise and irrelevant information. Wavelet transforms are well suited to decompose the structure of a dataset in a way that preserves short and long-term events. They do so by projecting the data into a space defined by translations and dilations of a mother wavelet function thus partitioning the time-frequency space.

The most common wavelet transform is the DWT. However, it suffers from several disadvantages that make it unsuitable for data-driven modeling such as data decimation, translation variance and nonzero phase. The MODWT overcomes these disadvantages by keeping redundant information and allows for the construction of a multiresolution analysis (MODWT-MRA) that provides an additive decomposition while preserving the time-frequency partitioning of the DWT. The MODWT-MRA applied to flow rate and pressure data creates "virtual wells" that preserve the same physical behavior of real wells while splitting the complexity of the original data into simpler components.

Chapter 3

Modeling Pressure Response

3.1 Introduction

The pressure transient equation described in Equation 2.33 establishes the relationship between pressure and flow rate as an input-output system where the reservoir is characterized by a response function $g(t)$. As such, production measurement of flow rate and pressure can be seen as readings coming in and out of the system and can be used for determining the response function. This chapter deals with the problem of approximating the response function using a data-driven approach with flow rate data as the input and pressure as the output. Two types of function classes were proposed for the approximation, linear and nonlinear. This chapter deals with both approaches including feature (input) generation, model fitting, and results analysis.

To obtain such a model of the reservoir, a complete modeling framework was developed. This framework covers the entire data processing pipeline that a modeler must go through in a real life scenario. It also deals with challenges that appear when dealing with production data such as data imperfections or discontinuities. In addition to the fitting of the flow rate to pressure mapping, additional goals were established for the full modeling framework to expand its applicability to realistic scenarios. These objectives are:

1. Use of a full data-driven model. No explicit use of flow equations or assumptions on the reservoir model..

2. Minimize the process of data cleaning and avoid the selection of individual transients for modeling.

3. Seamlessly deal with the presence of noise in the data. Assume that noise is an inherent characteristic of the data.

4. Allow for the use of incomplete data in the model building process. This can be in the form of data discontinuities in time or uneven sampling frequencies between data variables.

All the previous objectives directly tackle issues commonly encountered when using field production data. Establishing them as properties of the methodology rather than working around them allows for a more robust and replicable process. The proposed framework incorporates the theory developed in the previous chapter and combines it with machine-learning and deep learning algorithms to create a complete modeling procedure for capturing pressure response. This chapter presents the modeling framework by first focusing into the model input or features development, then introducing relevant background on the modeling techniques and then evaluating the performance of the framework. Finally, the issue of missing data is explored.

3.2 Model Input Development

The objective of this stage is to obtain the set of input variables that will be used by the statistical model to approximate the reservoir pressure response function. The simplest and initial input variable to be tested is flow rate itself without any modification or preprocessing. This is the base scenario and therefore the application of the MODWT-MRA decomposition is tested against this scenario.

3.2.1 Decomposition Algorithm

Once the wavelet filter, decomposition depth and boundary conditions have been chosen, the features can be created accordingly. The feature generation process can be summarized in the following steps:

1. Select flow rate data to be used for modeling.
2. Split datasets into training, validation and testing if applicable.
3. Determine the maximum decomposition level J_0. The depth of the decomposition must be applicable to all training, validation and testing datasets.
4. Apply the J_0th level MODWT-MRA separately to each of the datasets.

3.3 Modeling Techniques

3.3.1 Lasso Regression

Lasso regression is a type of linear regression with shrinkage, a type of regularization where the objective is to select the smallest subset of input features with the strongest effects on the output variable. This is a desirable property when trying to identify the relevant MODWT-MRA flow rate levels that are relevant for the target well's pressure responses.

The Lasso estimate is defined as:

$$\begin{aligned} \hat{\beta}^{lasso} &= \arg\min_{\beta} \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\ \beta_j \Big)^2 \\ &\text{subject to } \sum_{j=1}^{p} |\beta_j| \ \leq \ s \end{aligned} \tag{3.1}$$

which can also be rewritten as:

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \Big\{ \frac{1}{2} \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\ \beta_j \Big)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \Big\} \tag{3.2}$$

where $\hat{\beta}^{lasso}$ are the regression coefficients corresponding to each of the input predictors, λ is the regularization parameter that controls the predictors coefficient shrinkage and i, j are the data and feature indexes.

The computation of the $\hat{\beta}^{lasso}$ coefficients can be done through coordinate descent or using the Least Angle Regression (LARS) algorithm. Implementations of this methodology are readily available for multiple programming languages like R or Python in packages such as glmnet [Friedman et al., 2010] and sckit-learn [Pedregosa et al., 2011].

3.3.2 LassoNet Regression

LassoNet is an extension of the Lasso that combines the linear component and a neural network component to preserve the feature sparsity properties and interpretability of the Lasso and the flexible function representation of neural networks. LassoNet does so by using a residual neural network architecture consisting of a single residual connection and an arbitrary feed-forward neural network. The functional form of the LassoNet is [Lemhadri et al., 2021]:

$$f_{\theta,W}(x) = \theta^T x + g_W(x) \tag{3.3}$$

where g_W is a feed-forward neural network with weights W, θ denotes the residual layer parameters and x are the input features. A graphic view of the LassoNet architecture is shown in Figure 3.1. The loss function for LassoNet is:

$$L(\theta, W) = \frac{1}{n} \sum_{i=1}^{n} l\Big(f_{\theta,W}(x_i),\ y_i\Big) \tag{3.4}$$

where n is the number of observations in x, and l denotes the appropriate loss function such as squared error. Finally, the objective function for LassoNet is defined as:

$$\begin{aligned} &\min_{\theta,W}\ L(\theta, W) + \lambda\ ||\theta||_1 \\ &\text{s.t.} \quad ||W_j^{(1)}||_\infty\ \leq M\ |\theta_j|, \quad j = 1, ..., d \end{aligned} \tag{3.5}$$

where $W_j^{(1)}$ are the weights for feature j in the first hidden layer, λ is the regularization parameter which encourages sparsity similar to that of the Lasso. The parameter M controls the strength of the linear and nonlinear components. A consequence of the constraint on W_j is that $W_j = 0$ if $\theta_j = 0$ so the feature j does not participate in the network if the residual layer weight θ_j is zero. When $M = 0$, only the residual layer is active and the formulation is that of the Lasso. Alternatively, when $M \to +\infty$, the regular unregularized neural network is obtained.

The LassoNet training fits the linear (residual) and nonlinear components simultaneously thus capturing the nonlinearities and selecting the relevant features at the same time. Algorithmically, a gradient descent step is done first to all model parameters followed by a hierarchical proximal operator applied to the input layer pair $\theta, W^{(1)}$.

The LassoNet can be implemented using deep-learning frameworks such as PyTorch or TensorFlow. An ongoing experimental implementation is available for Python through the LassoNet package and it incorporates the basic training modules using PyTorch as the neural network engine.

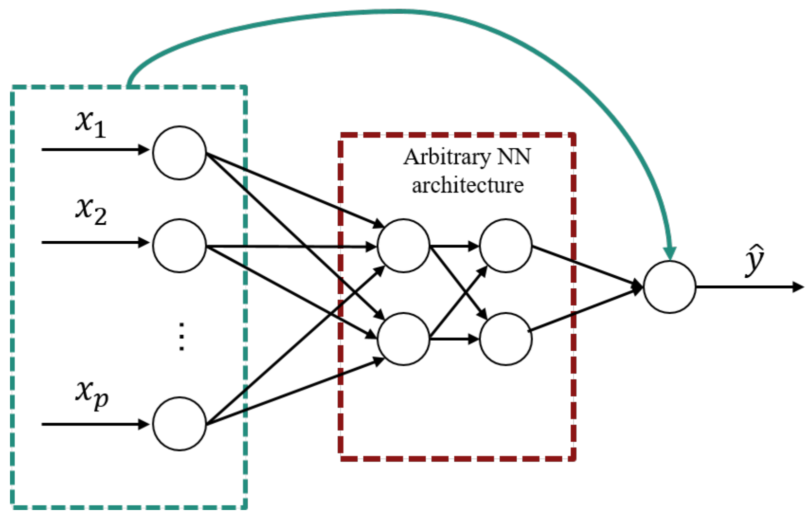

Figure 3.1: LassoNet architecture. Figure recreated from [Lemhadri et al., 2021]

LassoNet Training

In addition to the parameters λ and M, there are other parameters that need to be tuned for training the LassoNet. One of the most important one is the size of the feed forward neural network. In this work, only fully connected neural networks were tested with sizes ranging between 1-5 hidden layers and 100-500 units per layer. For the application in hand, the size of the hidden layers proved to be more important than the number of hidden layers. Overall, the best compromise of speed and accuracy was found by using a single hidden layer of size 100. This makes the full neural network a 3-layer network when counting the input and output layers as well.

Another highly relevant choice that must be made is the optimizer used for fitting the neural network. For this specific application, the Adam optimizer [Kingma and Ba, 2014] was used with a learning rate between 0.001 and 0.01. Carefully choosing the right set of parameters and using cross validation does not guarantee success for all training experiments. For this application, retraining the LassoNet with the same parameters can lead to significantly different results. In general, similar results are reached when retraining but it is not uncommon to obtain poor results if the optimizer converge in a local minimum. Setting a seed for the random number generator and fine tuning the optimizer learning rate help with alleviating this problem.

To identify a suitable regularization λ, the best process is to use cross-validation and trace a "regularization path" where λ is slowly increased until the model is fully regularized and all the features have been dropped out. Then, the λ value can be selected by simply choosing the value with the best cross-validation loss. However, tracing the regularization path can be challenging if the starting point is poorly chosen or the step taken at each iteration is too small or too large. The ideal path starts with all the features present in the model and ends with a model where all features have been dropped. Starting the path at $\lambda = 0$ can lead to time consuming or irrelevant results, specially if the step between consecutive λ is small. At each value of λ the LassoNet needs to be trained multiple times when using cross-validation, so identifying the right step and range for the regularization parameter is crucial for time management. For this exercise, it was found that defining the path as a geometric progression covering multiple orders of magnitude was best. Specifically, the path started at $\lambda = 400$ and had a maximum value of $\lambda = 6,600$. In practice this maximum value was rarely reached because most features were dropped at lower values of λ.

The linearity parameter M is one of the most relevant parameters in terms of model performance impact as it controls the flexibility of the model. The authors of the LassoNet suggest a value of $M = 10$ as best for most applications. For this work however, the best performance was obtained with a value of $M = 200$. Higher values of M could also be used but proved to be more prone to overfitting. Values lower than $M = 100$ showed high bias, especially in scenarios with large amounts of missing data.

3.4 Single Well Response

A single well scenario exists when a well interacts in isolation with the reservoir. It might be an unlikely scenario for a real oil field albeit it is the initial building block for modeling more complex systems. As stated earlier, the objective of the single well problem is to obtain the reservoir response function $g(t)$ from available flow rate and pressure readings. The problem is framed as a forward one, where the physical input of the system is the flow rate and the output is the pressure response. The model itself is a function mapping between these datasets and corresponds to a nonparametric representation of the function $g(t)$.

3.4.1 Data Generation

To test the methodology a set of synthetic datasets was generated. Each dataset consisted of a single well flow rate and pressure data. The flow rate data for the well was generated as a randomized step function with up to 35 step changes with a maximum value of 400 STB/day. The sampling frequency of the data was set to one observation per hour and a total duration of one year. The pressure solutions were obtained analytically on Laplace space and subsequently subjected to numerical inversion using the Gaver-Stehfest algorithm [Jacquot et al., 1983]. A total of 20 data sets were constructed with different flow rate sequences each to test the methodology in a variety of production scenarios. The 20 data sets shared the same reservoir model, which includes wellbore storage, skin effect, infinite acting radial flow and a constant pressure boundary. Moreover, Gaussian noise of 5% and 1% was added to both flow rate and pressure datasets correspondingly. The reservoir was assumed homogeneous with parameters summarized in Table 3.1. Figure 3.2 shows five of the 20 datasets generated with the described methodology.

3.4.2 Model Specifications

Four model specifications were fitted for the 20 synthetic datasets. Two of the models were based on the Lasso and two used the LassoNet for the fitting. In all cases, the train and validation sets

Table 3.1: Reservoir parameters for the synthetic data sets

Parameter	**Value**
k	100
S	0
C (STB/psi)	0.001
μ (cp)	1
h (feet)	50
ϕ	0.2
c_t (/psi)	5e-6
r_w (feet)	0.208
r_e (feet)	10,000
B	1

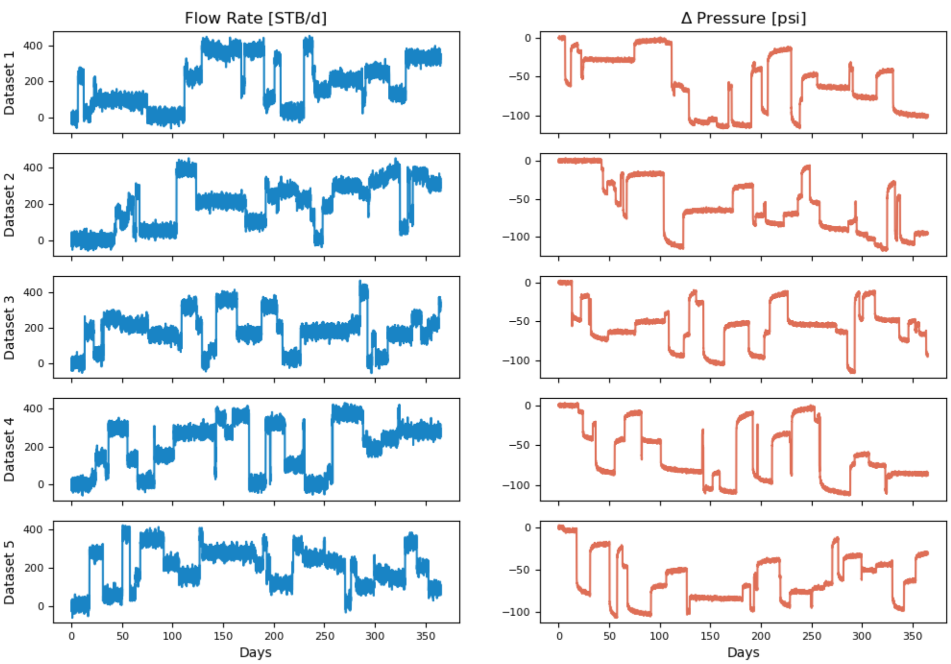

Figure 3.2: Examples of the flow rate and pressure synthetic datasets

used 80% of the data with the last 20% reserved for the test set. The baseline scenarios used flow rate as the only model feature. For the MODWT-MRA models, the wavelet filter was the Haar filter and the depth parameter J was set to 12 for a total of 13 features. Table 3.2 contains the different

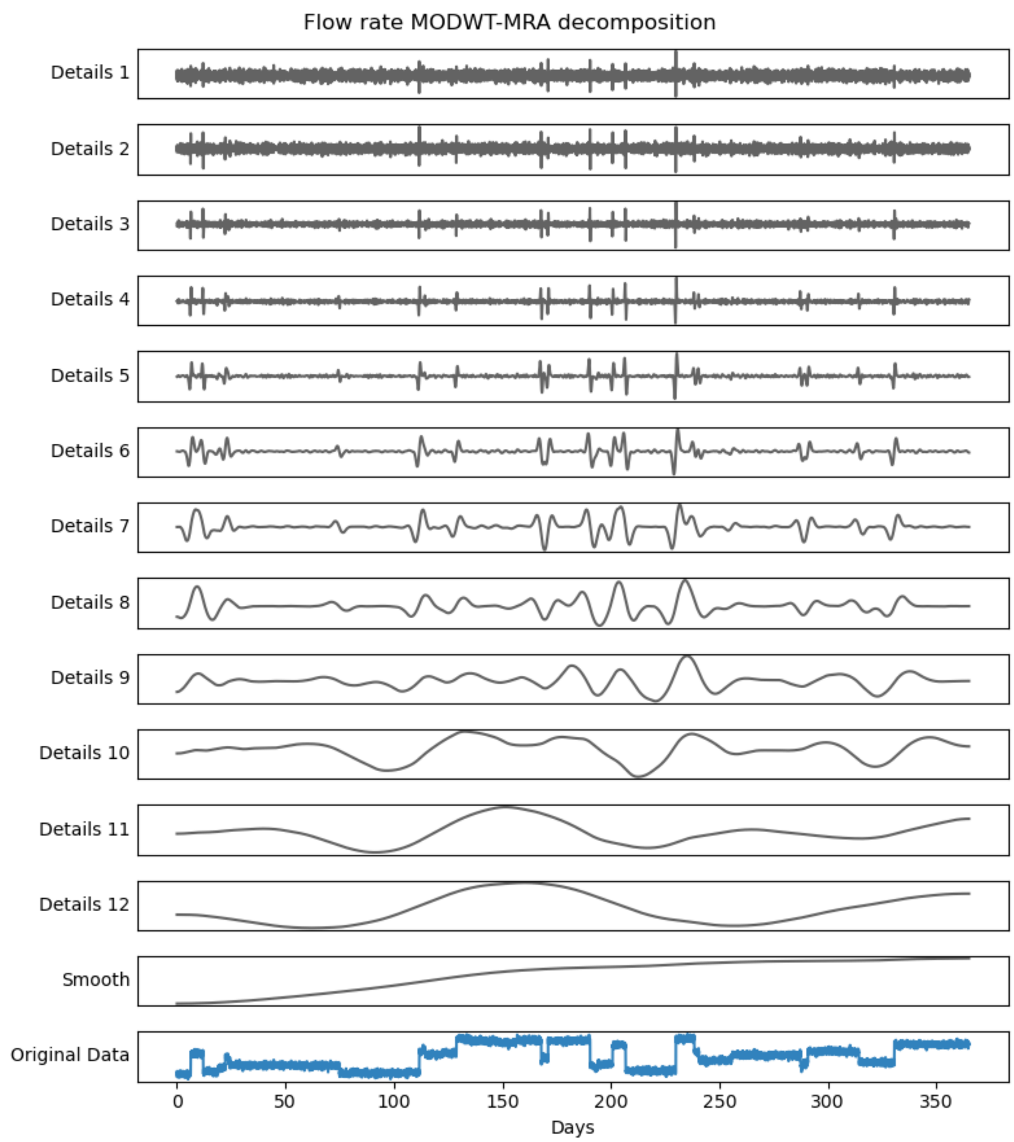

Figure 3.3: Flow rate MODWT-MRA decomposition of one of the synthetic datasets. Higher frequency components are captured by the details and the lowest frequencies are contained in the smooth.

Table 3.2: Model descriptions for the scenario with complete data

Model Name	Regression Method	Features	Target
A	Lasso	Flow rate (unprocessed)	Pressure
B	LassoNet		
C	Lasso	MODWT-MRA of flow rate	
D	LassoNet		

names and specifications for all four models, referred as models A through D.

Gamma Deviance as Performance Metric

To effectively compare model performance across different datasets, gamma deviance was chosen as the performance metric. Because the synthetic datasets contain a variety of ranges in pressure response, a metric that could be compared across scales is necessary. R2 is one of such metrics, however during testing of scenarios with missing data, R2 proved to be an unreliable metric due to reduced variance in the samples. Gamma deviance is an scaling-invariant metric that derives from the more general Tweedie deviance [Dunn and Smyth, 2008], [Dunn and Smyth, 2005]. In fact, both mean squared error and gamma deviance are both special cases of the Tweedie deviance. Equation 3.6 shows the definition for the gamma deviance:

$$D(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 2 \left(\log(\hat{y}_i / y_i) \; + \; y_i / \hat{y}_i \; - \; 1 \right) \tag{3.6}$$

3.4.3 Performance Results

Figures 3.4 to 3.7 show the results of the four proposed models for one of the 20 synthetic datasets. For this scenario, all four models show a good performance of capturing the pressure response from flow rate data. However, models A and B which use only the unprocessed flow rate display much higher noise in the pressure estimates when compared to models C and D although there is no observable bias in the estimate. Models C and D, which use the MODWT-MRA decomposition as inputs clearly display a significant noise reduction in the pressure estimate. This behavior is not unexpected, because the noisy components of the flow rate are contained in a few levels of the MODWT-MRA and thus can be deemed less relevant for the pressure estimate.

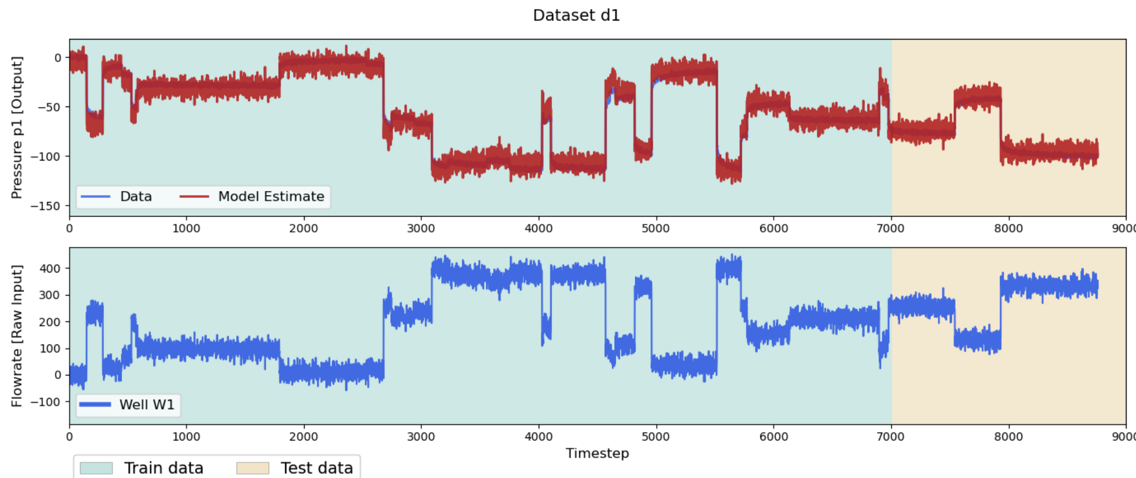

Figure 3.4: Model A: Lasso regression with unprocessed flow rate as input. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

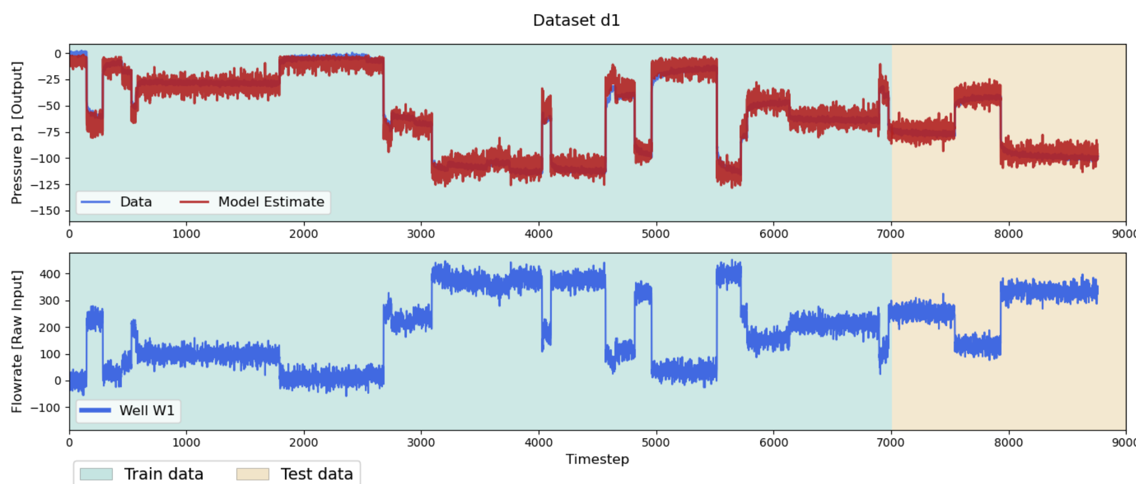

Figure 3.5: Model B: LassoNet with unprocessed flow rate as input. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

A more representative way to assess the performance of the different models is to compare them

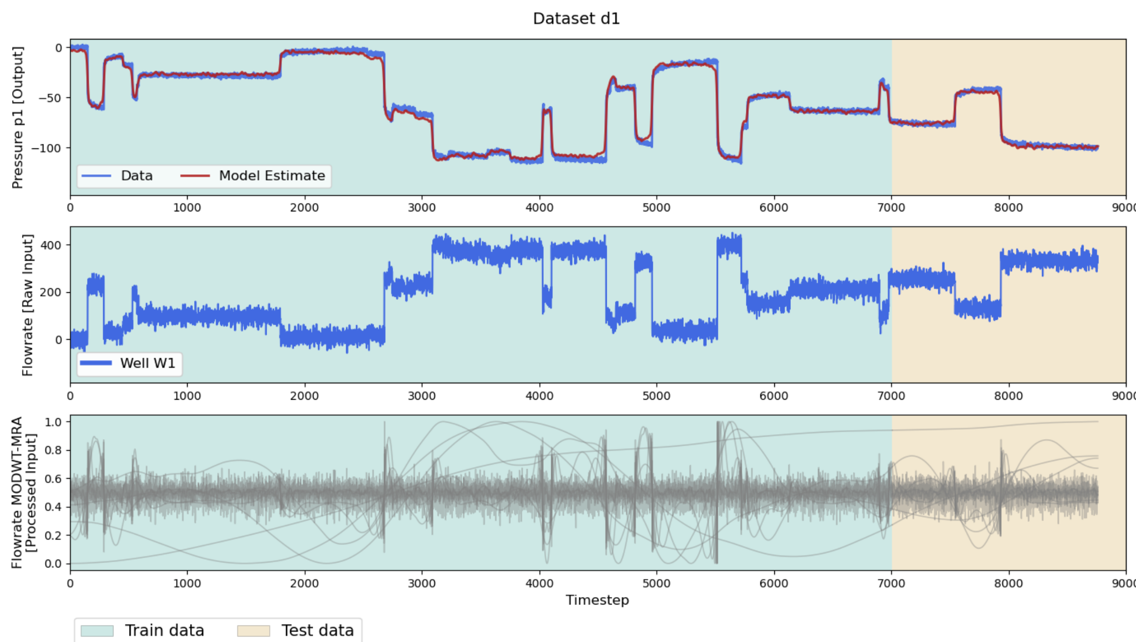

Figure 3.6: Model C: Lasso regression with flowrate MODWT-MRA as input. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

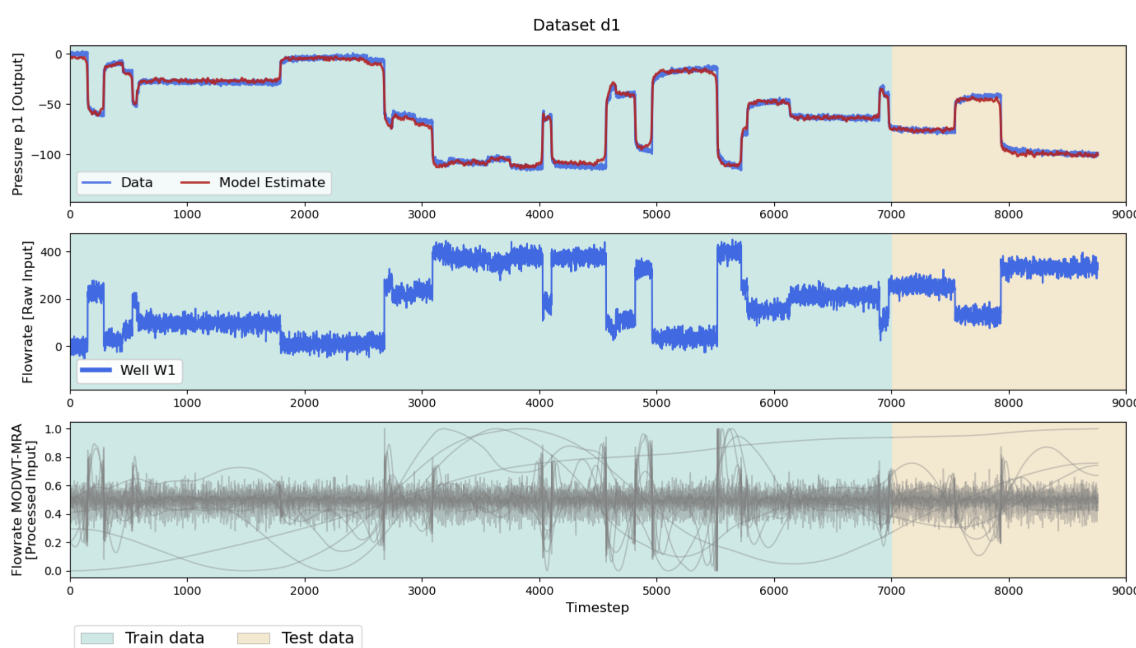

Figure 3.7: Model D: LassoNet with flow rate MODWT-MRA as input. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

across multiple datasets. Figure 3.8 shows the gamma deviance distribution of all four models across the 20 synthetic datasets. In training, all four models show a tight gamma deviance distribution. It is clear however that models C and D have a lower median gamma deviance and thus display better performance. The same trend is true for the test data, in which both models C and D also outperform models A and B. These results point that for the single well scenario, using the MODWT-MRA of flow rate as an input seems to be a better option than using only the unprocessed flow rate data.

By comparing the performance distribution of models B and C, it is evident that better performance is gained by using the decomposed MODWT-MRA. In this scenario, the linear Lasso with MODWT-MRA features outperforms the more complex LassoNet without the MODWT-MRA features. For test data, both models C and D present more consistent performance distributions.Model D (LassoNet with MODWT-MRA) has the best overall results. This is not unexpected because model D combines the high flexibility of the LassoNet with the benefits of using the MODWT-MRA decomposition as inputs.

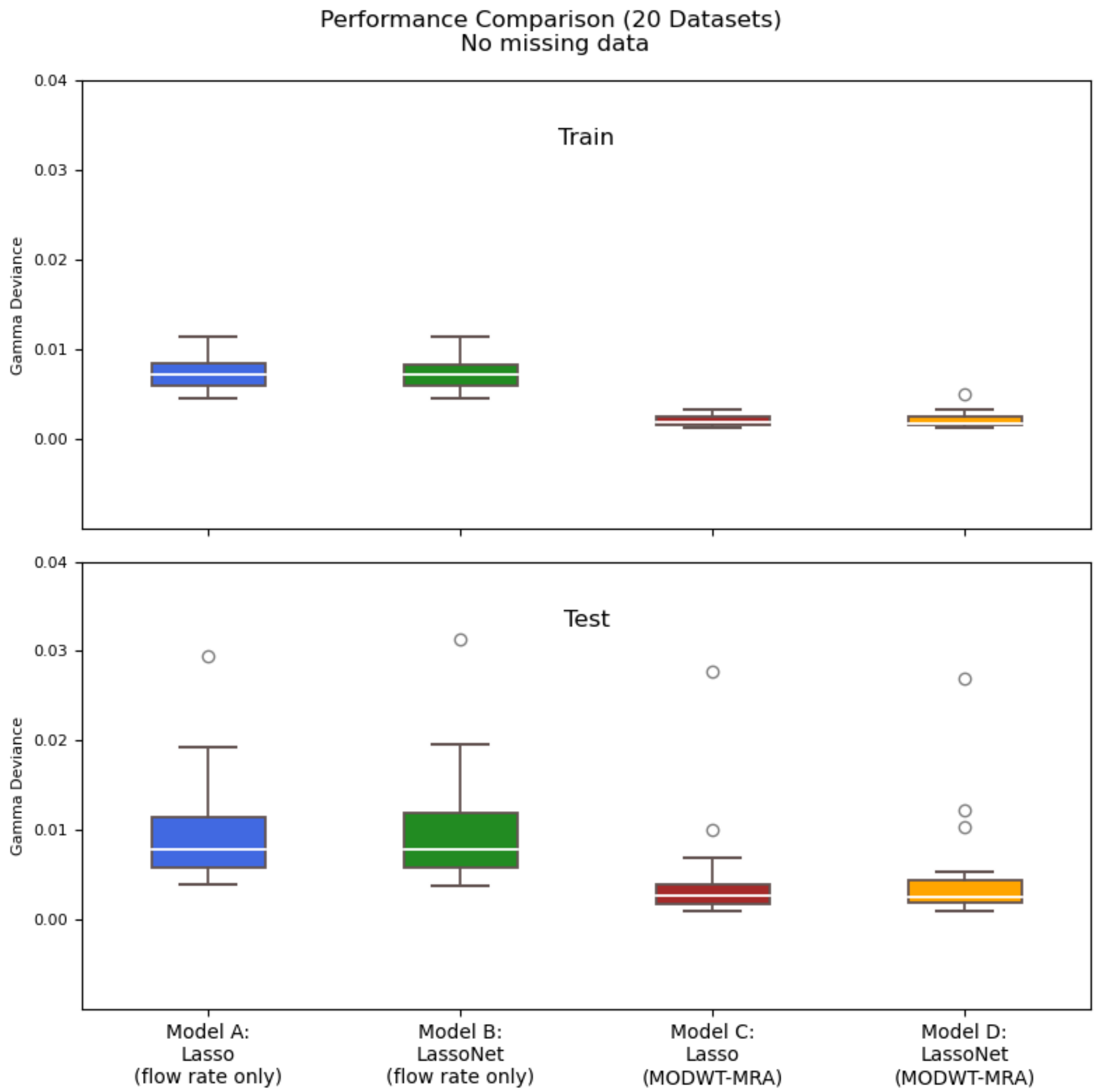

Figure 3.8: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

3.4.4 Handling Missing Data

It is not uncommon for production data to contain gaps of different duration. These can be caused by faulty equipment, operational constraints or unknown causes. With this in mind, the modeling framework was designed to allow for these gaps to exist without deeming the entire dataset unusable. The goal of the framework is not to fill the gaps or make assumptions about the missing data but instead to allow for the use of an incomplete dataset while still obtaining a model of the reservoir response.

Missing data gaps can be viewed as individual events of specific duration. The true physical event itself, such as a flow rate change is unknown but the absence of information has a beginning and end. Therefore, as noted in Section 2.4.2, the uncertainty about the event is contained in one of the bands of the MODWT-MRA decomposition depending on the time duration of the missing data gap. This property is the foundation to the proposed methodology's ability to deal with missing data. Conceptually, the MODWT-MRA decomposition encapsulates the uncertainty of the missing data gap in a limited set of the decomposition levels. This allows for the model features to still capture the higher or lower frequency events present in the data with minimum disruption. Using the virtual wells analogy presented in Section 2.4.1, the missing data are then missing only in only a few of the virtual wells.

However, the MODWT-MRA cannot be applied to a dataset with null data. Hence, a substitute for the missing data is needed. When inserting values, the objective is to minimize the disruption to each of the MRA decomposition levels and keep it as close as possible to what it would have been without missing data. For the proposed methodology, a linear interpolation is applied to the missing data before applying the MODWT MRA. The linear interpolation results in MRA detail series where the interpolated regions have values close or equal to zero. Because the MODWT-MRA details are centered in zero, applying linear interpolation does not affect the mean value at each level. For small amounts of missing gaps, the effect of the interpolation in the variance at each level becomes negligible. For data with many gaps or large amounts of missing data, the interpolation causes a decrease in the data variance at the relevant MRA scales [Percival, 2008]. However, for the purposes of designing input features for a pressure model the bias in the input MRA variances is acceptable. Figure 3.9 shows the MODWT-MRA decomposition of a dataset with missing data and compares it with the same decomposition done with the full data. In the high frequency levels, the MODWT-MRA of missing data shows flat lines for the missing gaps. At low frequencies the difference between the decompositions with or without missing data becomes imperceptible to the naked eye.

The updated framework to design models features when missing data is present then becomes:

1. Select flow rate data to be used for modeling.

2. Split datasets into training, validation and testing if applicable.

3. Identify the missing flow rate data gaps and apply linear regression to fill the gaps.

4. Determine the maximum decomposition level J_0. The depth of the decomposition must be applicable to all training, validation and testing datasets.

5. Apply the J_0th level MODWT-MRA separately to each of the datasets.

3.4.5 Missing Flow Rate Data

For this exercise, the original 20 synthetic datasets were modified to include missing data in flow rate. The goal of this exercise is to understand the impact of incomplete input in the pressure estimate of the model. To introduce missing gaps, ten windows of 300 hours were deleted from the flow rate data at randomized places for each well. The methodology results in no single dataset containing the same missing data points. The inserted missing data gaps can overlap with each other creating longer duration gaps. The total percentage of missing flow rate data per dataset ranged between 15% to 35% as noted in Table 3.4. Pressure data was not modified for any of the datasets. Because

Figure 3.9: Flow rate MODWT-MRA decomposition showing the effect of missing data. The shadowed area displays the missing data gaps. Differences in the MODWT-MRA levels decreases at lower frequencies.

Table 3.3: Model descriptions for the scenario with missing flow rate data

Model Name	**Regression Method**	**Features**	**Target**
A	Lasso	Flow rate	Pressure
B	LassoNet	(Linear interpolation only)	
C	Lasso	Flow Rate	
D	LassoNet	Lin. Interp. + MODWT-MRA	

both the Lasso and LassoNet cannot be fitted with missing data, the flow rate for models A and B (no MODWT-MRA) was also linearly interpolated to allow for model fitting. For models C and D, both the linear interpolation and MODWT-MRA decomposition were applied to flow rate. Table 3.3 shows the features and target descriptions for all four models A-D for the missing flow rate data scenario.

Table 3.4: Fraction of total missing flow rate data for each of the synthetic data sets.

Dataset	**Missing Flow Rate %**
d1	0.25
d2	0.30
d3	0.25
d4	0.23
d5	0.16
d6	0.21
d7	0.34
d8	0.29
d9	0.29
d10	0.34
d11	0.33
d12	0.35
d13	0.30
d14	0.29
d15	0.20
d16	0.30
d17	0.30
d18	0.32
d19	0.29
d20	0.35

Performance Results: Missing Flow Rate Data

Figure 3.10 and 3.11 show the the pressure models fitted with the Lasso and LassoNet respectively using Dataset d1 with incomplete data. In both, there was only a linear interpolation in the missing flow rate data but no MODWT-MRA was applied. It can be noted that in both train and test datasets, the models show a relatively unbiased pressure estimate in the sections where flow rate data was available. However, the two models display highly noisy estimates, especially when compared with models C and D in Figures 3.12 and 3.13 which incorporate the MODWT-MRA of flow rate as inputs. Models C and D are able to reduce the noise levels significantly in the test datasets and show little decrease in performance overall. The results are similar across the rest of the 20 datasets, with models C and D showing less noisy pressure estimates.

Figure 3.14, shows the gamma deviance distributions across all datasets. It is noticeable that model D, the LassoNet with flow rate MODWT-MRA inputs has the best median performance as well as the lowest 75 percentile for test data. The models that used the MODWT-MRA decomposition of flow rate as input show significantly better performance than those using the unprocessed flow rate. Overall, the introduction of missing data affected negatively the performance distributions of all models by increasing the variance. However, it is clear from Figure 3.14 that the gamma deviance distributions for models C and D are shifted towards lower (better) values compared to models A and B.

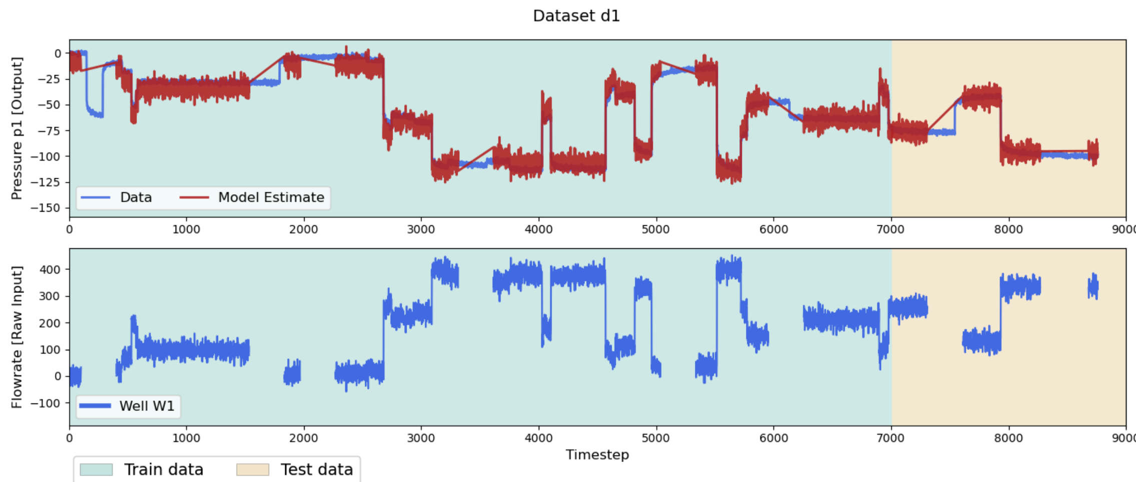

Figure 3.10: Model A: Lasso regression with missing flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

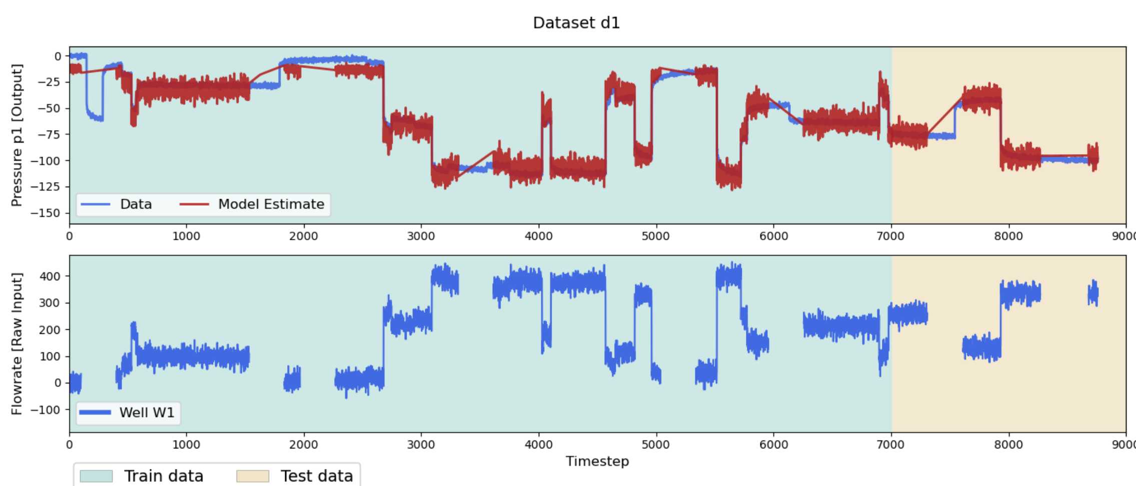

Figure 3.11: Model B: LassoNet with missing flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

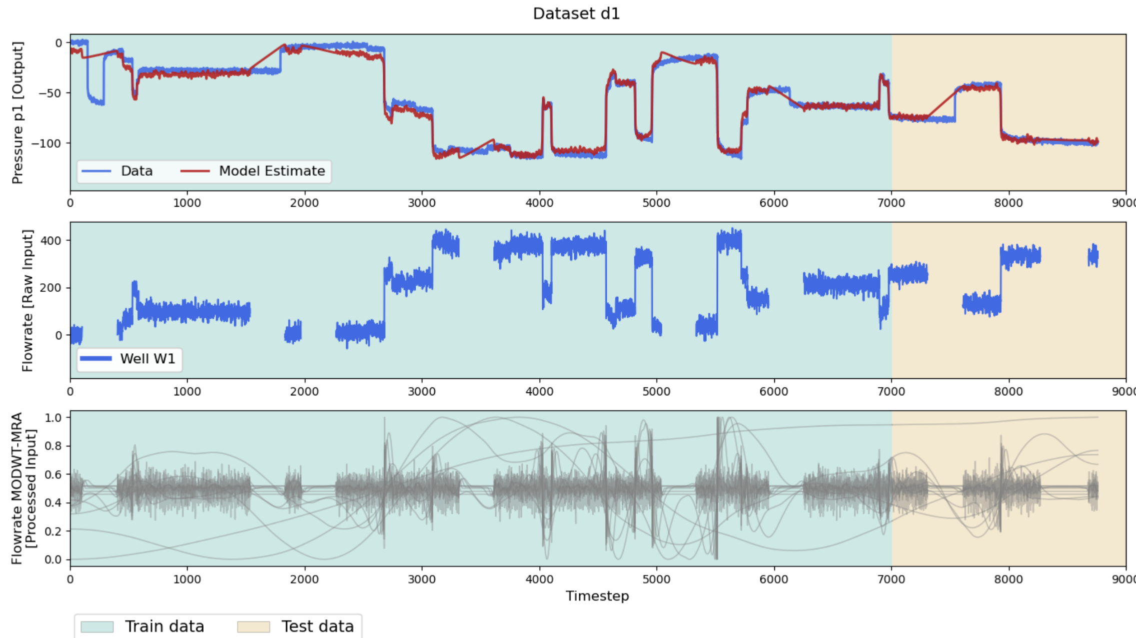

Figure 3.12: Model C: Lasso regression with missing flow rate data. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

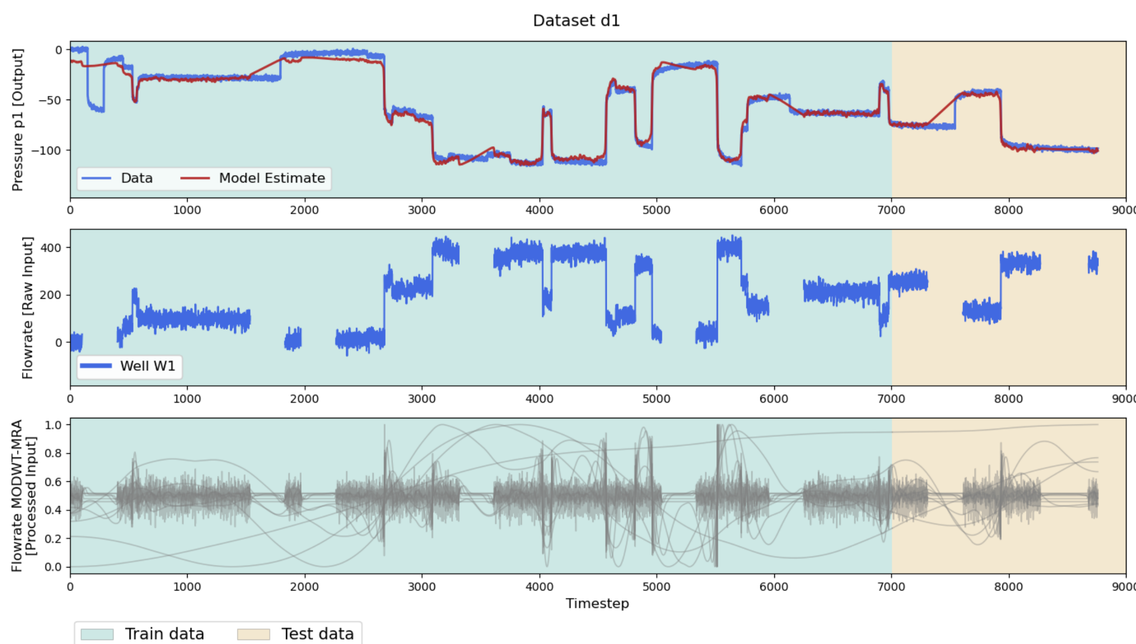

Figure 3.13: Model D: LassoNet with missing flow rate data. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

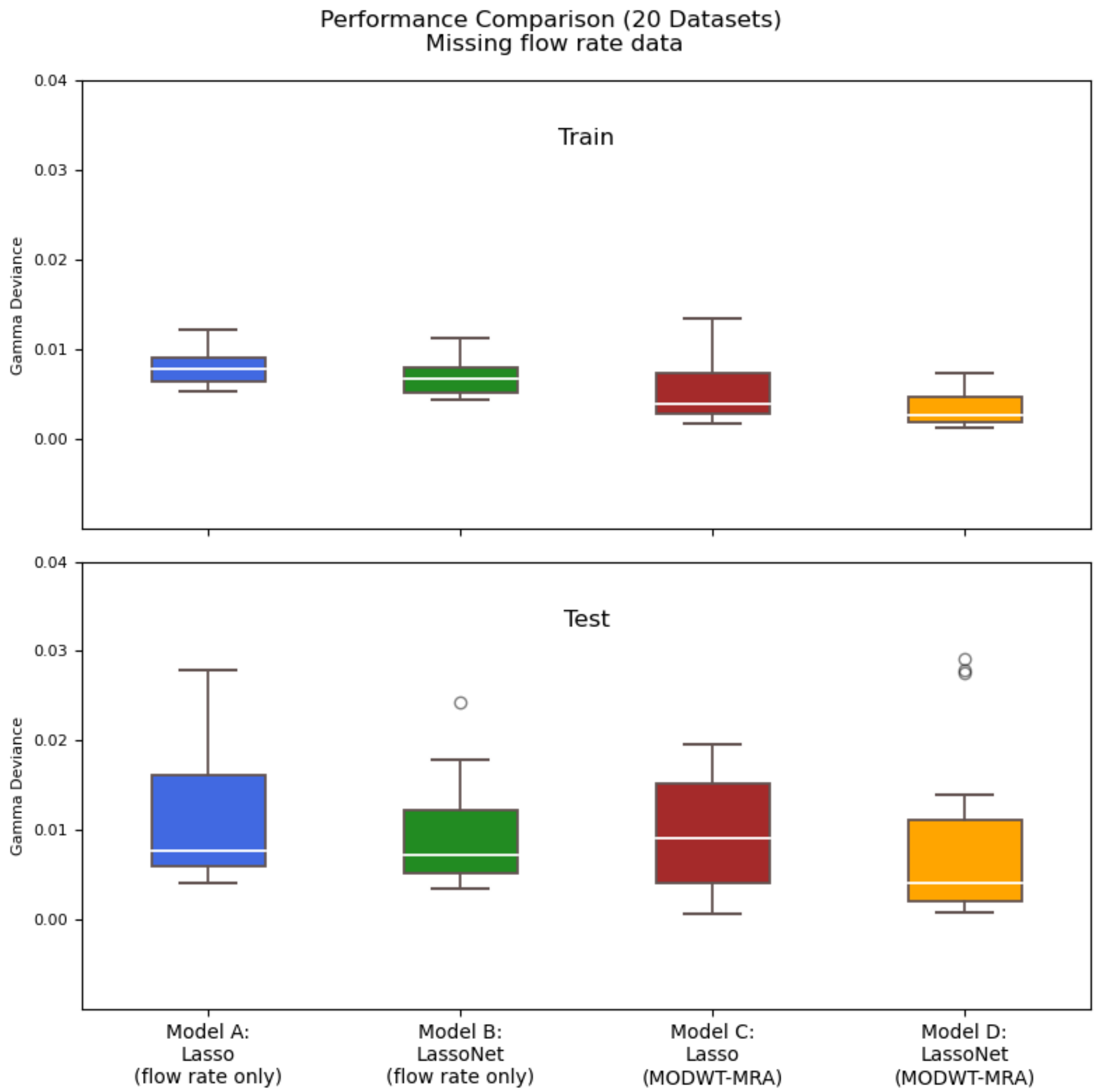

Figure 3.14: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

3.4.6 Missing Flow Rate and Pressure Data

The next scenario to test the methodology is one where both the flow rate inputs and pressure outputs have some amount of missing data. Having missing data in both inputs and outputs is a much more challenging modeling scenario due to data fragmentation. Even as individual variables have relatively small amounts of missing data, the remaining data may not be contiguous and the missing gaps might not occur at the same time for each variable. Since a model can only effectively “learn” from the parts of the dataset with both flow rate and pressure, the modeling difficulty becomes higher with increasing data fragmentation. Nonetheless, applying the MODWT-MRA can still encapsulates the uncertainty within a few levels for the flow rate data. However, for pressure data this is not the case as it is used without processing other than interpolation.

For this scenario, the strategy for introducing missing data was applied to both flow rate and pressure datasets by randomly deleting consecutive periods of 300 hours. Within a single dataset, the process was applied independently to flow rate and pressure. This results in missing gaps that are not necessarily aligned in time for both variables thus increasing data fragmentation. Table 3.5 shows relevant statistics for the resulting datasets. It can be seen that for each variable (flow rate or pressure) the total missing fraction is between 17% to 38% per variable. However, when looking at the fraction of the data that is fully complete, the percentage oscillates between 39% and 61%. Therefore, without applying some measure to fill those gaps or contain them around 50% of the data could be unusable for learning.

Similarly to Section 3.4.5, linear interpolation was used to fill the missing gaps in flow rate and pressure data for models A and B since both the Lasso and LassoNet cannot natively handle missing data. For models C and D, both linear interpolation and the MODWT-MRA decomposition were applied to flow rate while pressure was only interpolated. Table 3.6 contains the descriptions of all four models A-D for this scenario.

Table 3.5: Missing data statistics for each of the synthetic data sets.

Dataset	**Missing Flow Rate**	**Missing Pressure**	**Full Data**	**No data (any variable)**
d1	0.32	0.32	0.48	0.11
d2	0.17	0.27	0.61	0.06
d3	0.36	0.25	0.52	0.12
d4	0.34	0.22	0.53	0.09
d5	0.30	0.38	0.41	0.09
d6	0.29	0.27	0.45	0.01
d7	0.23	0.33	0.47	0.03
d8	0.38	0.35	0.39	0.12
d9	0.29	0.30	0.46	0.05
d10	0.29	0.30	0.48	0.07
d11	0.36	0.28	0.45	0.08
d12	0.28	0.26	0.50	0.05
d13	0.27	0.31	0.47	0.06
d14	0.33	0.31	0.46	0.11
d15	0.34	0.34	0.42	0.10
d16	0.24	0.35	0.47	0.06
d17	0.24	0.26	0.57	0.07
d18	0.24	0.28	0.52	0.04
d19	0.29	0.29	0.50	0.08
d20	0.31	0.22	0.57	0.11

<table>
<tr><th>Model Name</th><th>Regression Method</th><th>Features</th><th>Target</th></tr>
<tr><td>A</td><td>Lasso</td><td rowspan="2">Flow rate
(Linear interpolation only)</td><td rowspan="4">Pressure
(Linear
interpolation)</td></tr>
<tr><td>B</td><td>LassoNet</td></tr>
<tr><td>C</td><td>Lasso</td><td rowspan="2">Flow Rate
Lin. Interp. + MODWT-MRA</td></tr>
<tr><td>D</td><td>LassoNet</td></tr>
</table>

Table 3.6: Model descriptions for the scenario with missing flow rate and pressure data

Figure 3.15 shows the results for model A applied to dataset d1. In the areas where flow rate data were present, a noisy yet unbiased estimate was achieved. The pressure response when there is no flow rate data available cannot be estimated and results in a straight line. Figure 3.16 displays the results for model B applied to dataset d1. A noisy but unbiased pressure estimate is also obtained with the LassoNet when the input is only flow rate and the pressure estimates where there is no data result in a straight line. Figure 3.17 shows model C applied to dataset d1. In contrast to previous scenarios without missing data or missing flow rate only, model C shows a noisy pressure estimate even when using as inputs the MODWT-MRA of flow rate. There is little to no difference in the estimates of model A and C for this dataset. In contrast, Figure 3.18 shows a much less noisy pressure estimate even when both flow rate and pressure data are missing. The LassoNet is able to filter out the noisy components of flow rate in a better way than the Lasso (model C) and is able to create an unbiased an less noisy estimate compared to models A B and C. These results suggest that the difficulty of capturing pressure behavior with fragmented data requires a methodology of higher complexity such as the LassoNet.

These results are further confirmed when looking at the gamma deviance distributions across the 20 different datasets. Figure 3.19 shows that models B and D, which were fitted with the LassoNet have the best test-set performance. Models A and C, fitted with the Lasso have the highest dispersion. Model D has the lowest median gamma deviance but it is noticeable that model B shows the least variance in pressure estimates. Overall, the models that used the LassoNet performed better and the use of the MODWT-MRA decomposition of flow rate resulted in lower noise even when showing a slightly higher variance across datasets.

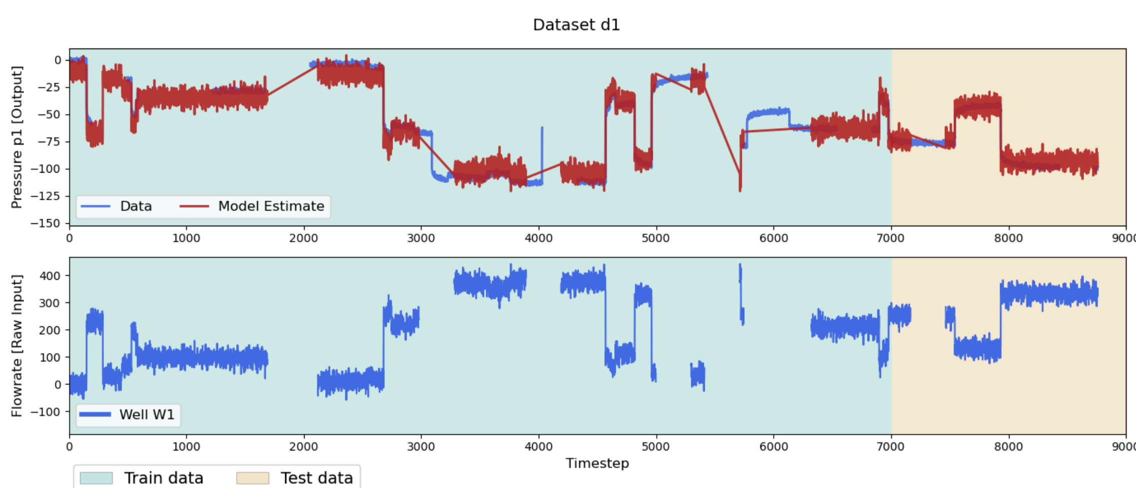

Figure 3.15: Model A: Lasso regression with missing gaps in both pressure and flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

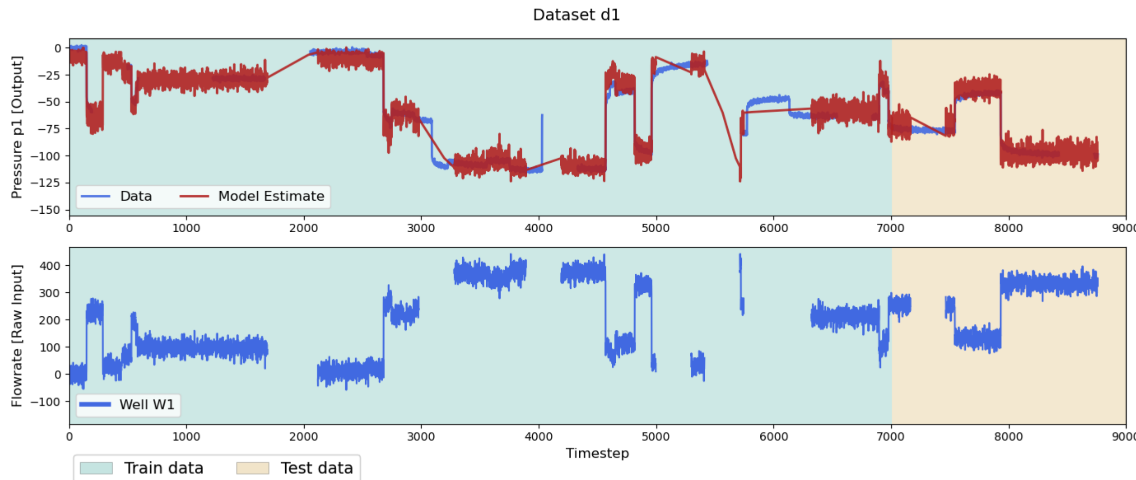

Figure 3.16: Model B: LassoNet with missing gaps in both pressure and flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

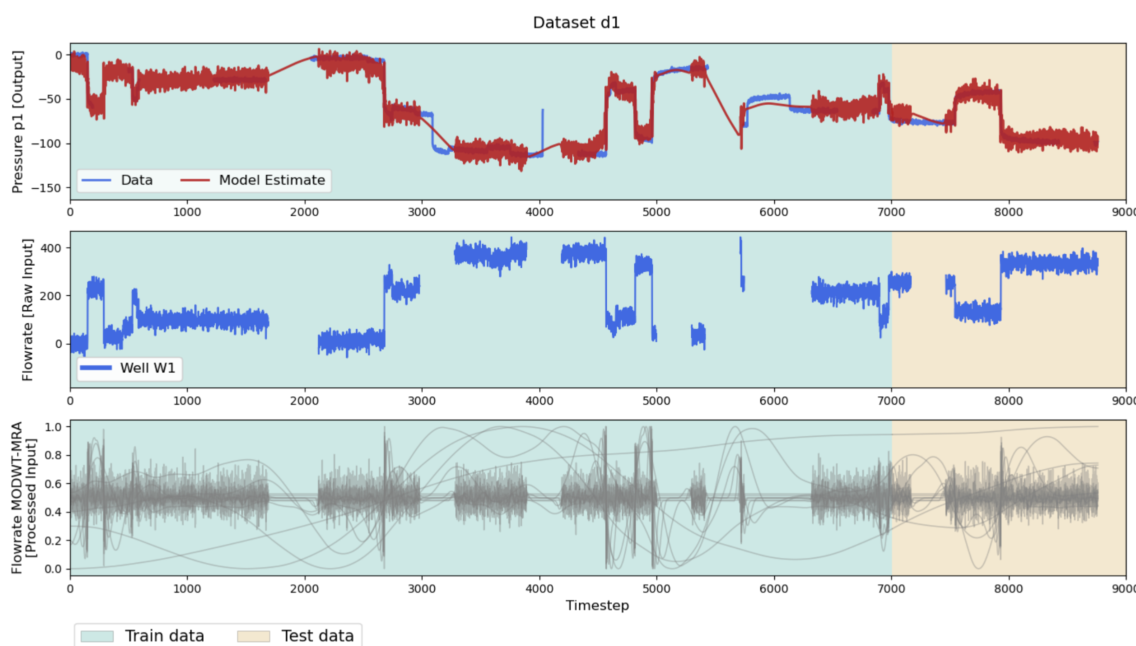

Figure 3.17: Model C: Lasso regression, missing gaps in both pressure and flow rate. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

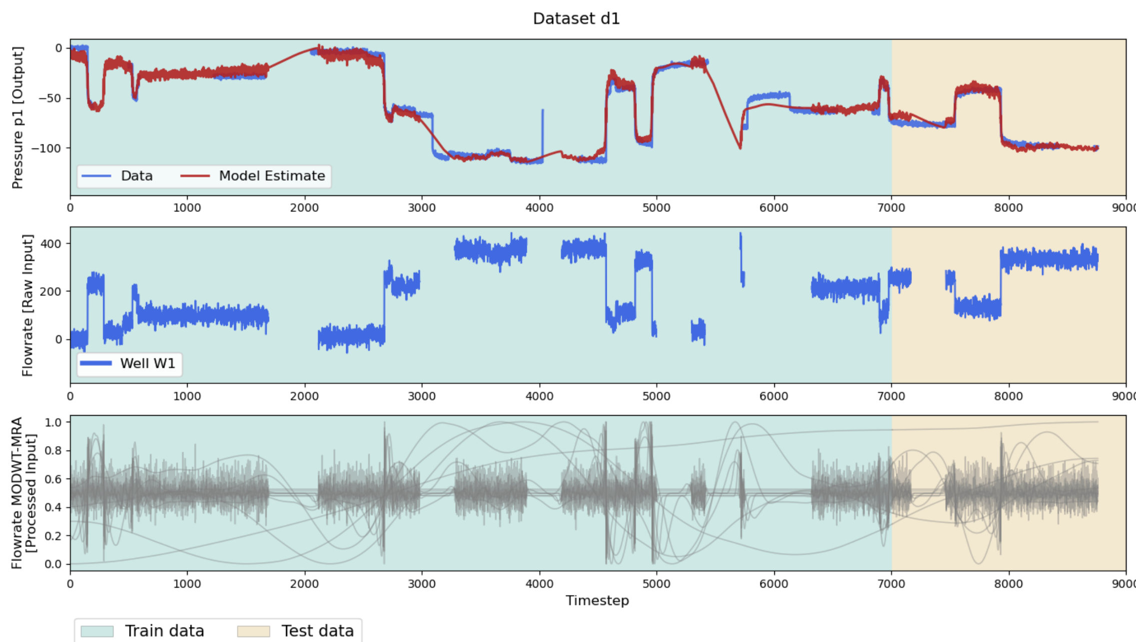

Figure 3.18: Model D: LassoNet, missing gaps in both pressure and flow rate. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

3.5 Two Well Response

The next step in the modeling framework is the extension of the single well scenario to a multi-well scenario, in this case the simplest one composed by two wells. The additional challenge in the two well scenario is presented when the both wells are connected in the reservoir. The principle of superposition establishes that the pressure response at any point in the reservoir is the caused by the sum of the responses caused by each well. In the case of a specific well, the pressure readings at the well will show interference from the second well at a distance. Section 3.4 showed that the MODWT-MRA flow rate features are useful to build models of single wells. A natural extension is then to evaluate the applicability of the framework to multiple wells and assess the applicability of a similar methodology to obtain a pressure response model for one well in the presence of another.

Specifically, the two-well scenario is composed by a pair of wells called W1 and W2. Both wells

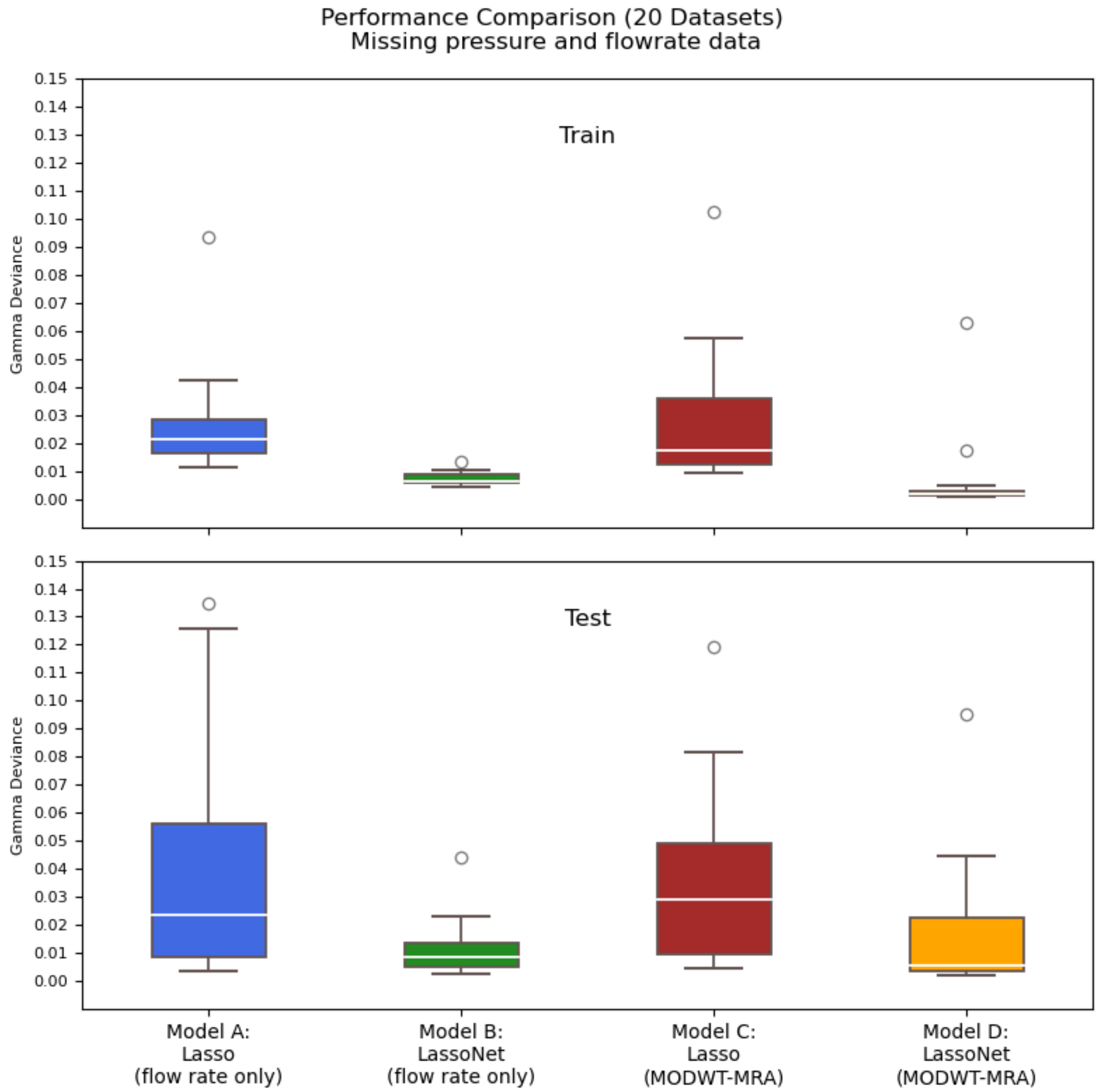

Figure 3.19: Models comparison for the single well scenario with missing flow rate and pressure. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

can produce at the same time with different flow rates and are connected through the reservoir, although that might be an unknown fact for the modeler. The objective is two-fold:

1. Identify if a well is connected

2. Obtain the pressure response model $g(t)$ of one of the wells using a data driven methodology.

The scenario assumes that at least the flow rate schedules of the two wells are known and the pressure history of at least one of the wells is available. Similarly to the single well scenario, the methodology is tested in synthetic data to be able to identify it's potential and measure it's performance when the true answer is known.

3.5.1 Data Generation

The data for the two well scenario were generated using a modified version of the data generating function for the single-well scenario. The reservoir parameters were kept constant as shown in Table 3.7. For each well, a flow rate schedule was created using a randomized step function of up to 35 step changes and a maximum rate value of 400 STB/day. The sampling frequency for each well is one observation per hour and the total duration of the data is 365 days. The pressure solutions were obtained analytically on Laplace space, including superposition of the two wells and then subjected to numerical inversion. A total of 20 datasets consisting of two wells each were created with different flow rate schedules each. The same reservoir model and parameters were used as in the single well scenario and Gaussian noise of 5% was added to flow rate data and 1% noise to pressure data.

3.5.2 Model Specifications

Similarly to Section 3.4, four model specifications were fitted for the 20 synthetic datasets. Two of the models were based on the Lasso and two used the LassoNet for the fitting. The training and validation sets used 80% of the data with the last 20% reserved for the test set. The baseline scenarios used flow rate as the only model feature. For the MODWT-MRA models, the wavelet filter was the Haar filter and the depth parameter J was set to 12 for a total of 13 features. Table contains the different names and specifications for all four models, referred as models A through D.

3.5.3 Performance Results

For the two well scenario, the goal is to capture the target well's pressure response interacting with the interfering well and also in isolation. A correctly fitted model of a well's response should be able to identify the effect of the interfering well in conjunction with the self-response of the target well. Model performance can be evaluated for the case of both wells flowing and thus interacting with each other. In this case, there is not an assumption of a shut-in well.

Figures 3.20 to 3.23 show the results of the four models for the double well scenario for one of the 20 datasets generated. For this dataset, all four models display unbiased estimates in both test and train data for the target well W1. However, models A and B (Figures 3.20 and 3.21) which do not use the MODWT-MRA in the flow rate inputs have a significantly higher noise in the pressure estimates when compared with models C and D (Figures 3.22 and 3.23).

Table 3.7: Reservoir parameters for the synthetic data sets in the two-well scenario

Parameter	**Value**
k	100
S	0
C (STB/psi)	0.001
μ (cp)	1
h (feet)	50
ϕ	0.2
c_t (/psi)	5e-6
r_w (feet)	0.208
r_e (feet)	10,000
B	1

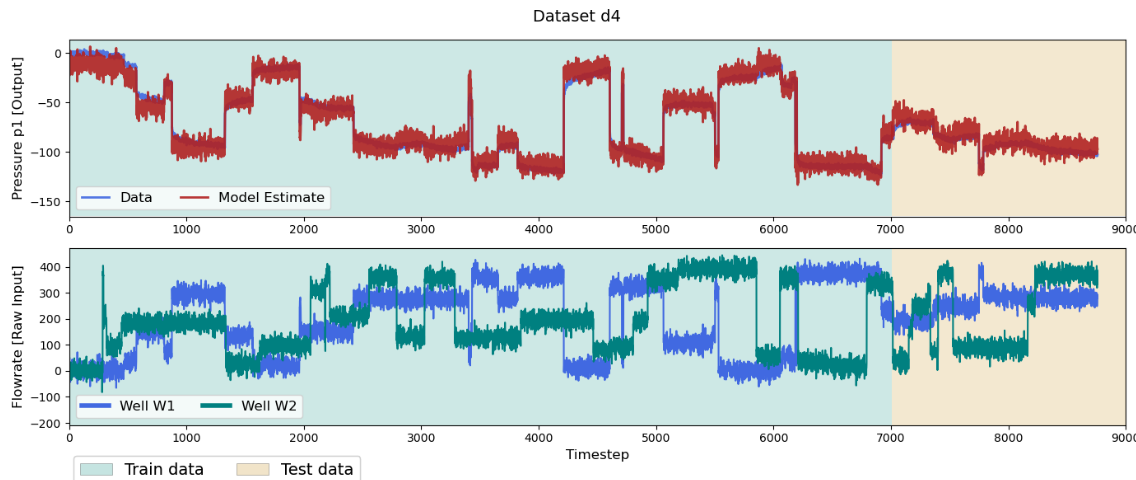

Figure 3.20: Model A: Lasso regression with flow rate as input. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

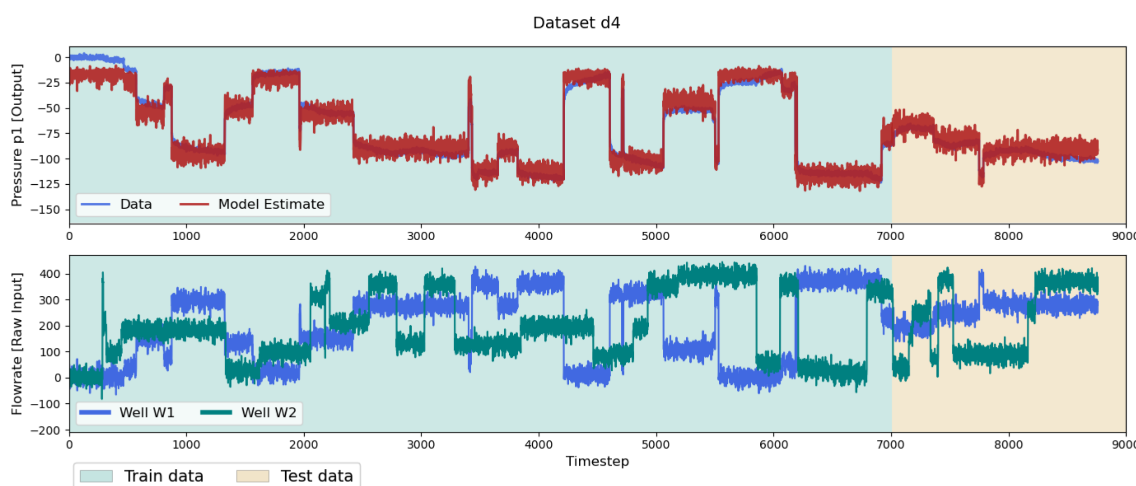

Figure 3.21: Model B: LassoNet with flow rate as input. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

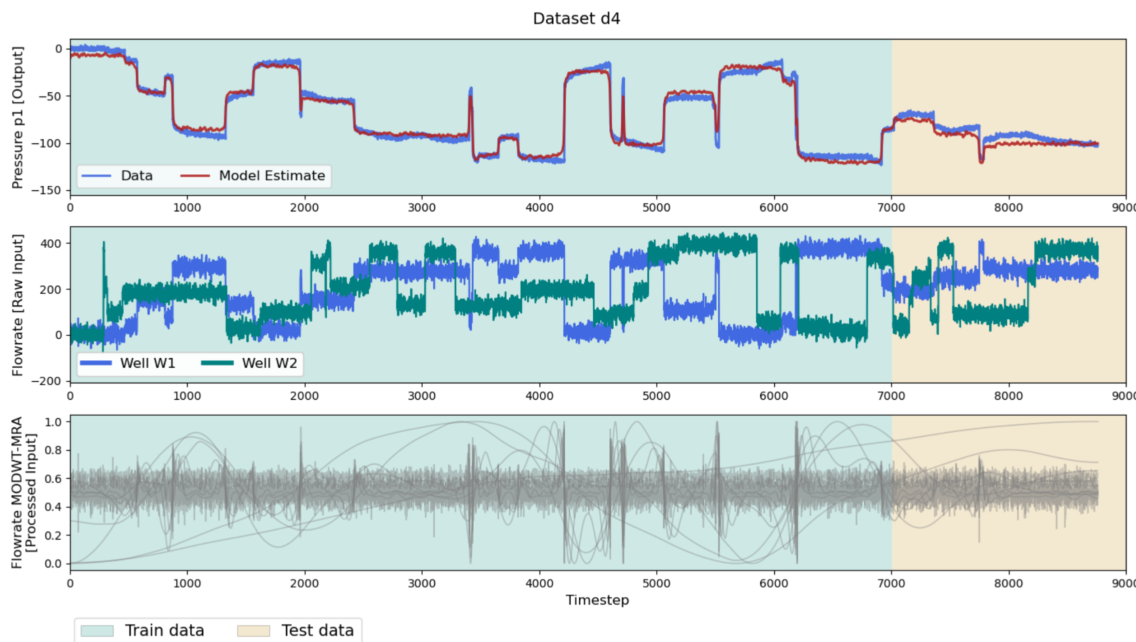

Figure 3.22: Model C: Lasso regression with flow rate MODWT-MRA as input. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

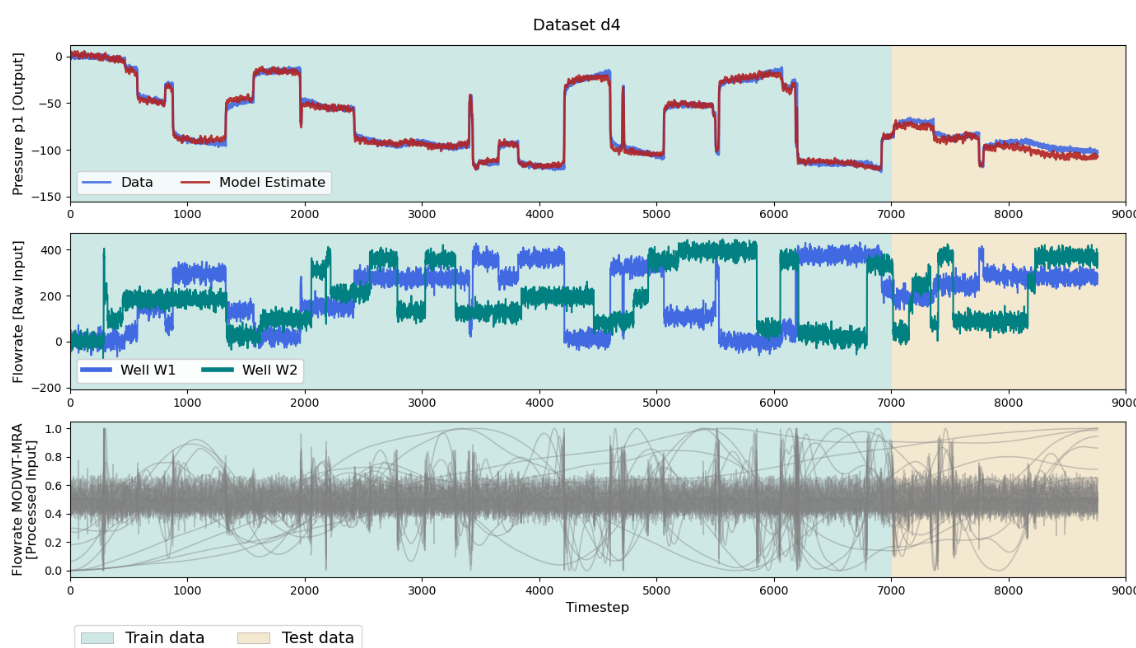

Figure 3.23: Model D: LassoNet with flow rate MODWT-MRA as inpu. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

Table 3.8: Model descriptions for the scenario with complete data

Model Name	Regression Method	Features	Target
A	Lasso	Flow rate (unprocessed)	Pressure - Well W1
B	LassoNet	Well W1 and W2	
C	Lasso	MODWT-MRA of flow rate	
D	LassoNet	Well W1 and W2	

3.5.4 Interference filtering

Another way to measure model performance is to evaluate the pressure response when only the target well flowing. This measures the capacity of a model to capture a well's self-response or the pressure response of the well in isolation. A successful model should be able to estimate the correct pressure self-response from data where two wells are flowing thus filtering out the interference in the pressure signal. Evidently, estimate performance in this scenario is only possible using synthetic data or by shutting the interfering well. In a real scenario, shutting a well might not be possible or prohibitively expensive.

To test performance of interference filtering, the previously trained models were used to estimate the pressure response of well W1 with the interfering well W2 is shut. The true self-response for the target well W1 was obtained numerically for all 20 datasets using the same flow rate schedules. This allowed to test the ability of the 4 models to filter out interference.

Figure 3.24 shows the results for model A, the Lasso with unprocessed flow rate as input. The yellow line in the top panel represents the true self response for well W1 and the grey line is the pressure data with interference. The difference between these two lines can be attributed to the effect of the interfering well W2. It can be seen that the model estimate is noisy and mostly follows the line with pressure data with interference. Moreover, the noise in the pressure estimate is of similar magnitude to the actual interference effect, making model A a poor model for interference filtering.

Figure 3.25 shows the results for model B, the LassoNet with unprocessed flow rate as input.

Similarly to model A, the pressure estimates mostly follows the pressure obtained with data with interference. Model B does not display the ability to successfully capture the self response of the target well W1. Moreover, it shows high levels of noise of comparable magnitude with the interference effect themselves.

Figure 3.26 shows the results for model C, the Lasso with the flow rate MODWT-MRA decomposition as input. Even if the pressure estimate is less noisy than models A and B, model C still follows the pressure line with interference. it is not a successful model to capture the self response of the target well. In fact, because of the lack of noise, the difference between the model estimate and the true self-response is more evident than models A and B.

Figure 3.27 shows the results for model D, the LassoNet with the flow rate MODWT-MRA decomposition as input. The pressure estimates closely follow the true self-response line and the difference between the self-response pressure data and the data with interference is clear. Out of the four models, model D is the only one that displays the ability to successfully capture the target well's self-response.

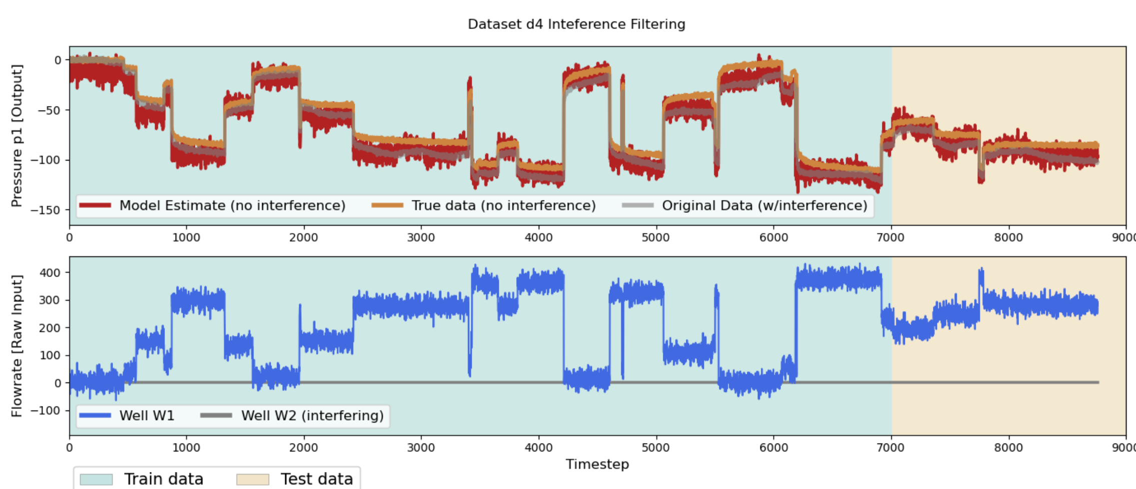

Figure 3.24: Model A: Lasso regression with flow rate as input. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

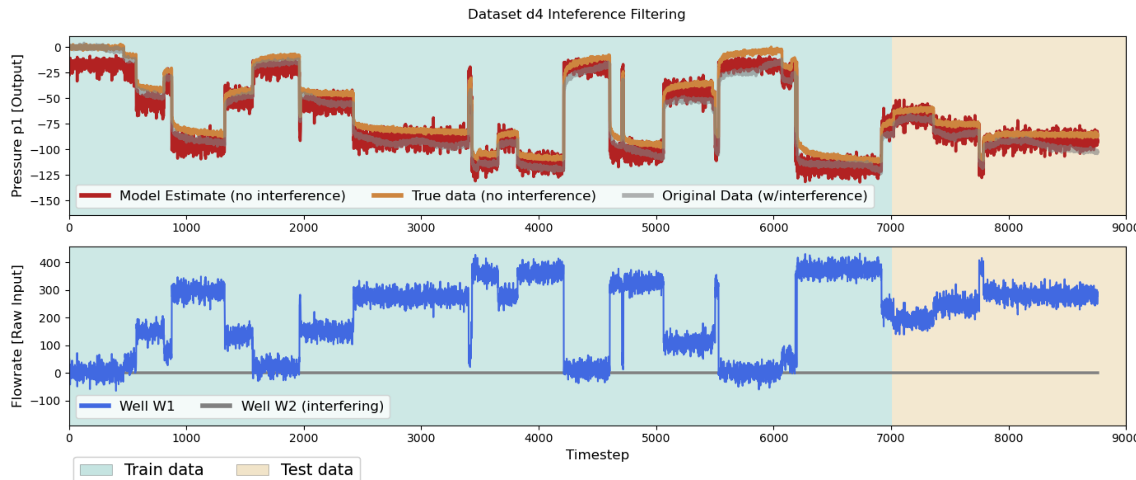

Figure 3.25: Model B: LassoNet with flow rate as input. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

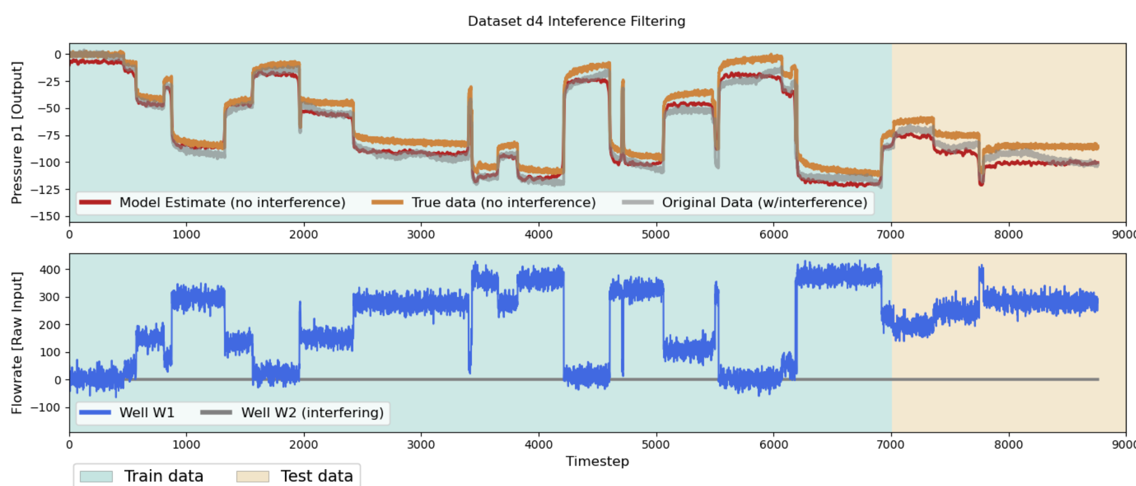

Figure 3.26: Model C: Lasso regression with flow rate MODWT-MRA as input. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate schedule.

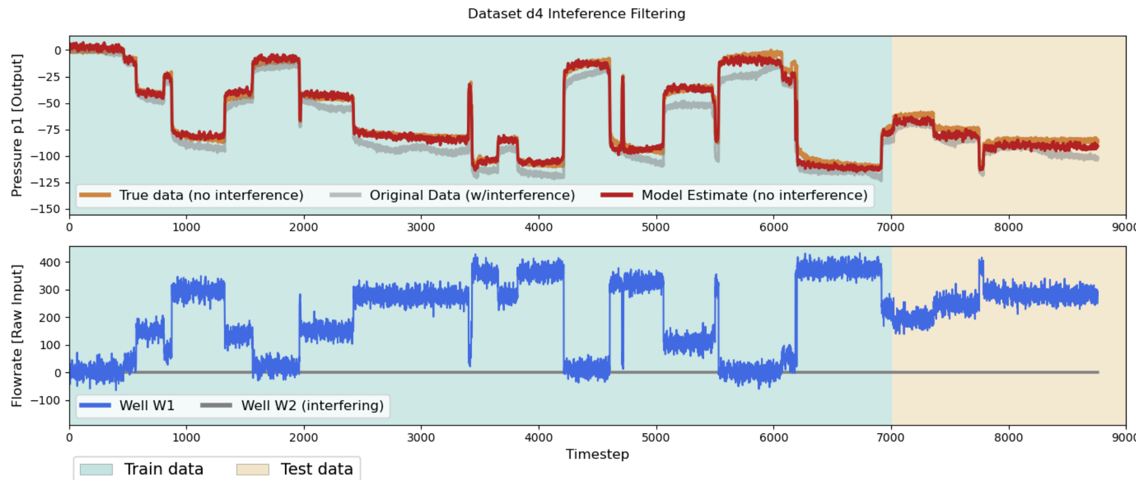

Figure 3.27: Model D: LassoNet with flow rate MODWT-MRA as input. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

The results from Figures 3.24 to 3.26 are further confirmed by the gamma deviance distributions across the 20 datasets. Figure 3.28 shows the distributions for the four models train, test datasets as well as the no interference scenario. The test results show that when both wells are flowing, all models have good performance, even if models A and B present noisy estimates. Model D has the widest dispersion among all 4 models for test data.

However, the interference filtering scenario shows highly different results. Model D has a much better performance across all 20 datasets. In fact it is the only model that showed a consistent capacity to capture the target well's self-response and filter out interference.

Overall, the results point to the benefit of using the MODWT-MRA flow rate decomposition as input for modeling pressure. It allows for a better estimates in terms of noise filtering and when combined with a flexible enough model such as the LassoNet, complex behaviors such as the capturing of a well's self response can be achieved. Model B showed that using only the LassoNet is not enough for easily filtering out interference.

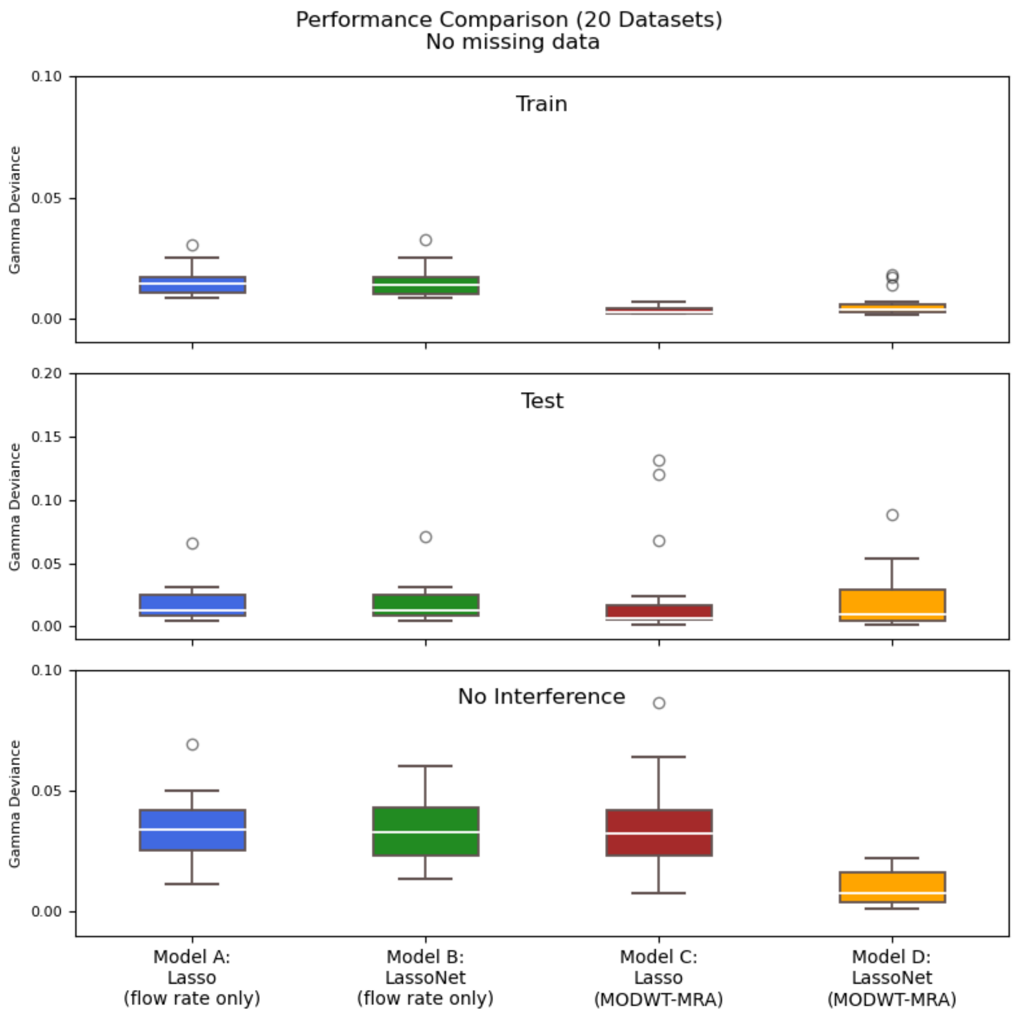

Figure 3.28: Models comparison for the two well scenario. The top panel corresponds to training data performance, medium panel shows the test data performance and bottom panel shows the interference filtering results. A lower gamma deviance denotes better performance.

3.5.5 Handling Missing Data

Similarly to the single well scenario, a the two-well scenario can also be modified to include varying amounts of missing data. The challenge for this scenario is two-fold: identify the correct pressure response model and deal with highly fragmented data. Data fragmentation is potentially higher in the two well scenario because there are more variables in play. To test the performance of the modeling framework, two missing data examples were designed: missing flow rate data and missing both flow rate and pressure data.

3.5.6 Missing Flow Rate Data

The same 20 datasets from the two-well scenario were used to test the framework performance with missing data. However, the datasets were modified to include missing data in both wells' flow rate data. A total of 10 periods of 300 hours of duration were eliminated from the flow rate data of each well at randomized time locations. Missing data periods were allowed to overlap and create longer overall missing data gaps, resulting in missing flow rate fractions between 17% to 39% per well. Because each dataset is comprised of two wells, each with missing data, data fragmentation is higher and fully complete data was reduced by about half per dataset as shown in Table 3.9. The fully complete data fractions varies between 39% to 61% among the 20 datasets.

The same four model combination A-D was tested including the Lasso and LassoNet with pure flow rate data as input as well as the MODWT-MRA of flow rate as input. Table 3.10 specifies the full descriptions of all four models. Similarly to Section 3.4.5, linear interpolation was used to fill the gaps in the missing data points so that both the Lasso and LassoNet could be fitted. Also, linear interpolation was applied before the MODWT-MRA decomposition. The same Haar wavelet filter and depth parameter J of 12 was maintained as in the two-well scenario with full data.

Figure 3.30 show the results of model A. It can be seen that the pressure estimate is centered around the true data when the flow rate data is present. Periods of missing flow rate data are filled with straight lines. The pressure estimate of model A is consistently more noisy than the true data, in a similar fashion to the case with full data. Figure 3.30 shows the results for model B. Its results are similar to those of model A, with a high noise but unbiased estimate where flow rate data is present. Model C, shown in Figure 3.31 shows little noise in the estimates and high accuracy. These results are consistent with the results in the full-data scenario. Model D has the least amount of noise of all the models and it also displays an unbiased estimate of pressure. Both models that incorporate the flow rate MODWT-MRA show less noisy estimates even in a high data fragmentation scenario.

Table 3.9: Fraction of total missing flow rate data for each of the synthetic data sets.

Dataset	**Missing Flow Rate W1**	**Missing Flow Rate W2**	**Full Data**
d1	0.27	0.27	0.56
d2	0.27	0.32	0.48
d3	0.32	0.28	0.54
d4	0.17	0.27	0.61
d5	0.26	0.30	0.55
d6	0.36	0.25	0.52
d7	0.33	0.28	0.45
d8	0.34	0.22	0.53
d9	0.33	0.25	0.49
d10	0.30	0.38	0.41
d11	0.35	0.31	0.44
d12	0.29	0.27	0.45
d13	0.26	0.24	0.55
d14	0.23	0.33	0.47
d15	0.30	0.33	0.50
d16	0.38	0.35	0.39
d17	0.29	0.20	0.53
d18	0.29	0.30	0.46
d19	0.39	0.39	0.39
d20	0.29	0.30	0.48

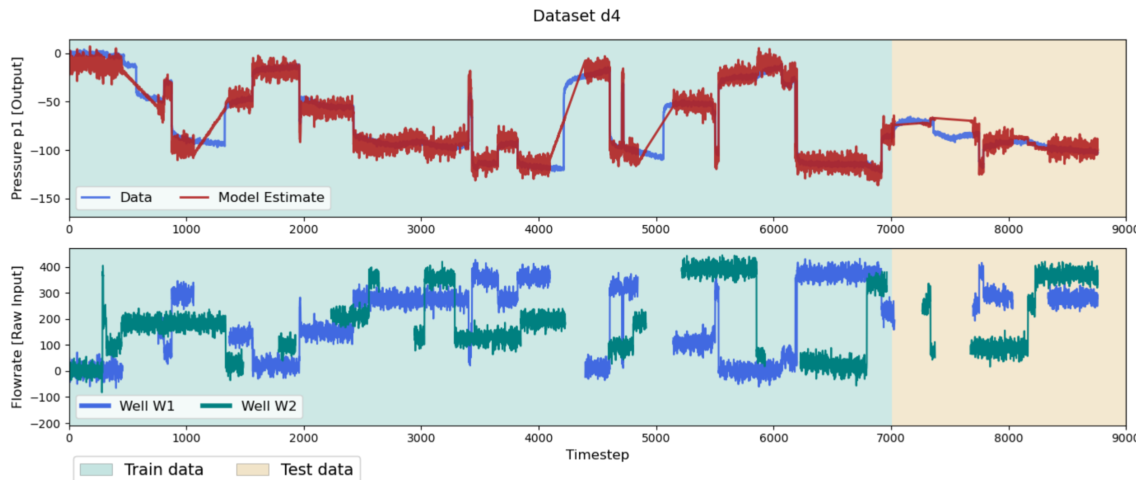

Figure 3.29: Model A: Lasso regression with missing flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

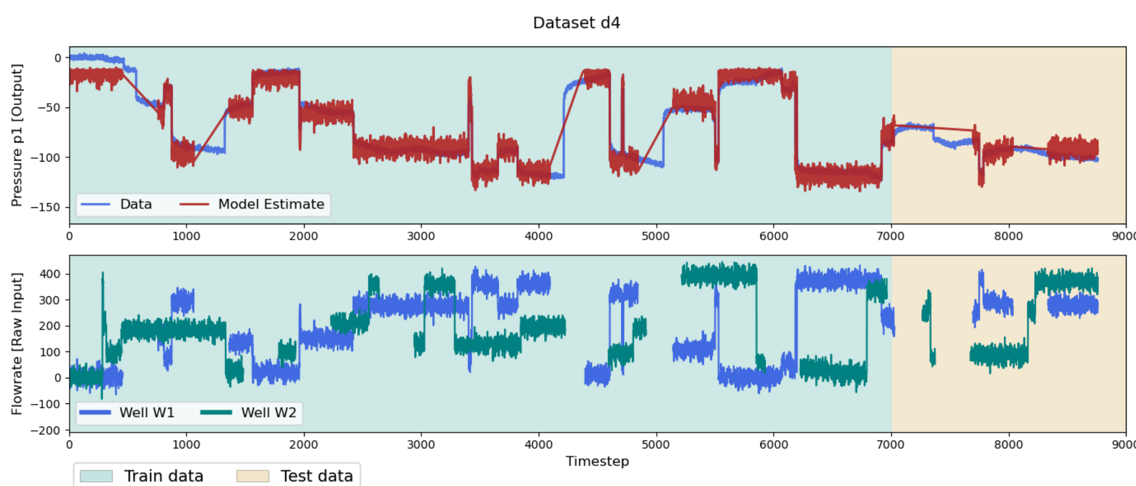

Figure 3.30: Model B: LassoNet with missing flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate schedule.

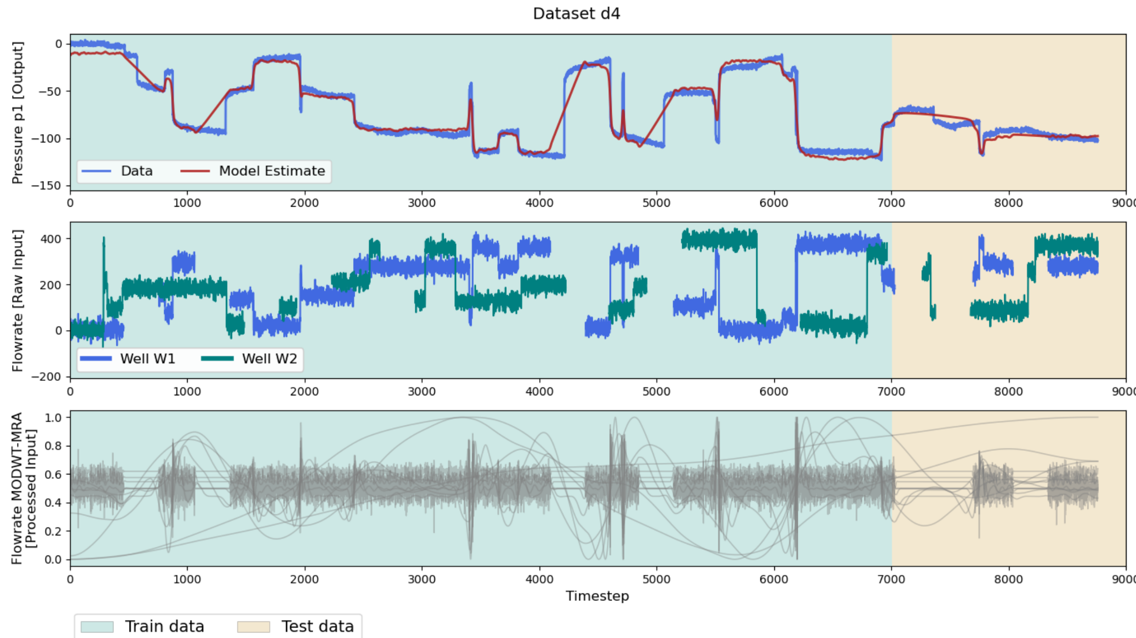

Figure 3.31: Model C: Lasso regression with missing flow rate data. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

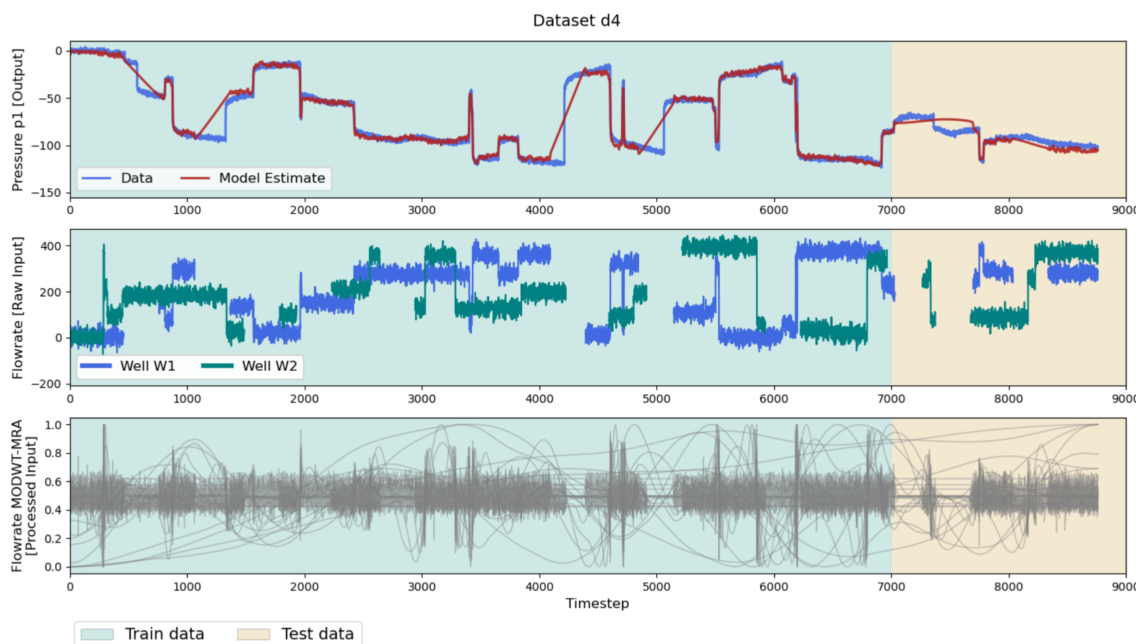

Figure 3.32: Model D: Lasso regression with missing flow rate data. Top panel: Pressure estimates. Center panel: Unprocessed flow rate schedule. Bottom panel: Flow rate MODWT-MRA.

Table 3.10: Model descriptions for the two-well scenario with missing flow rate data

Model Name	Regression Method	Features	Target
A	Lasso	Flow rate (Linear interpolation only)	Pressure Well W1
B	LassoNet	Well W1 and W2	
C	Lasso	Flow rate Wells W2 and W2	
D	LassoNet	Lin. Interp. + MODWT-MRA	

Interference Filtering with Missing Flow Rate Data

In a similar way to the two-well full data scenario, the performance of the four models can be tested using the self response of the target well W1. This is the response of the well W1 with the same flow rate schedule but assuming the interfering well W2 is shut. Figures 3.33 to 3.36 show the results of interference filtering for models A to D. Overall it is noticeable that models A, B and C do not successful capture the self-response of the target well. The estimated pressure when the interfering well is shut closely resembles that of the scenario with both wells flowing. Models A and B display higher noise that can mask the results due to the fact that the interference effect is of similar magnitude to the noise levels. Model C has less noisy output but its pressure estimates still resemble those of the case with both wells flowing. Only model D, the LassoNet with flow rate MODWT-MRA inputs successfully captures the self response of the target well W1. In figure 3.36 it can be seen that the pressure estimate matches the theoretical pressure self-response and it also displays a lower amount of noise when compared with models A and B. These results are similar to those in the full-data scenario, proving that model D is still capable of fully capturing the self response even with high data fragmentation.

However, the increase in data fragmentation does come with a downgrade in overall performance for model D. In Figure 3.37, the gamma deviance distributions for all four models are shown for train, test and interference filtering scenarios. Across the 20 datasets, model D has the lowest median gamma deviance. However, there are some datasets where performance is not as good as it was with full data. Nonetheless, model D is still the only one able to successfully filter out interference and capture the self-response for the target well.

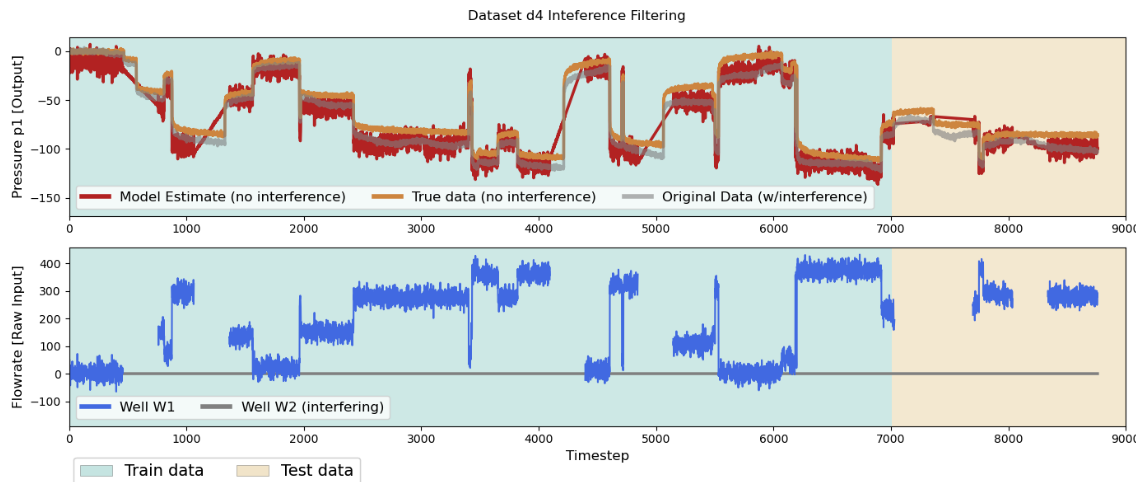

Figure 3.33: Model A: Lasso regression with missing flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

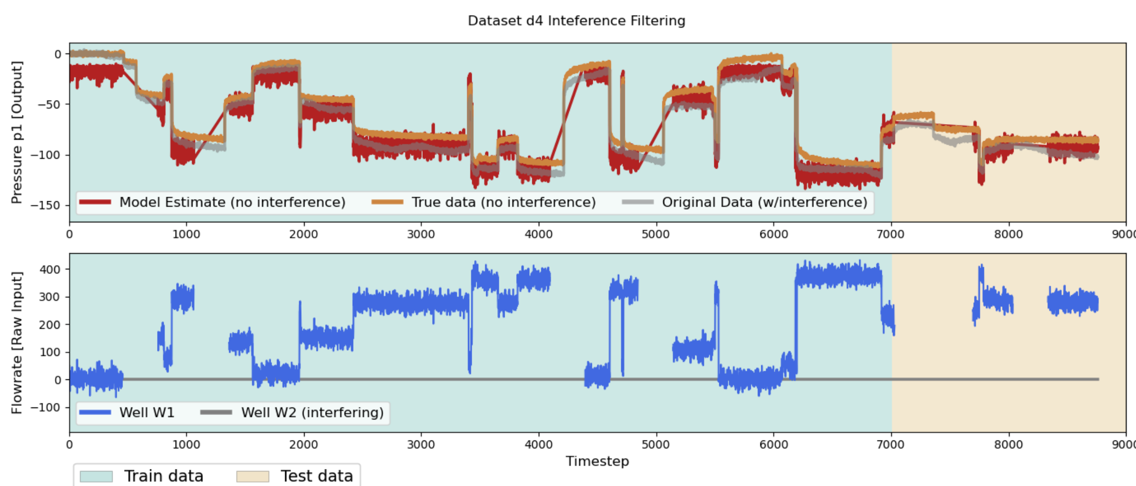

Figure 3.34: Model B: LassoNet with missing flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

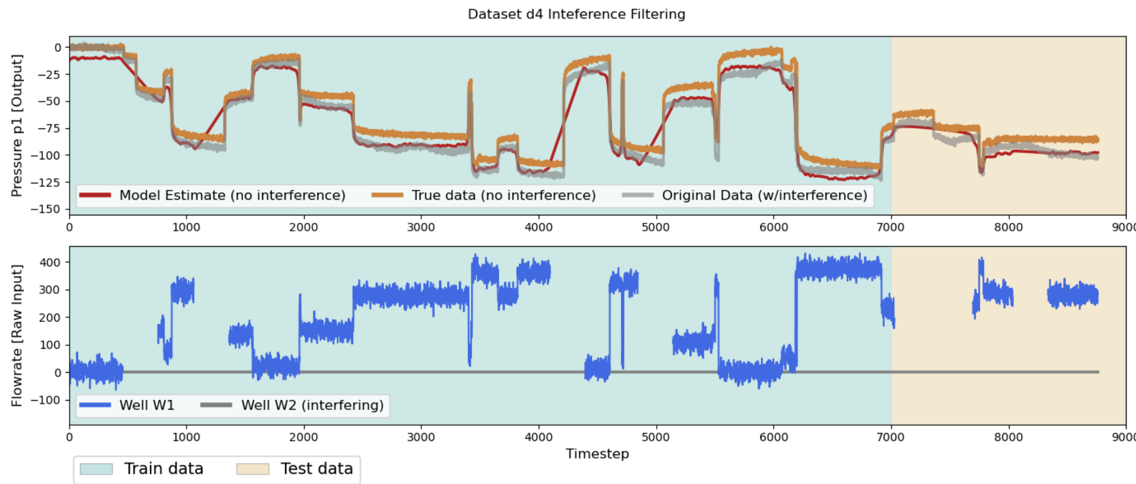

Figure 3.35: Model C: Lasso regression with missing flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

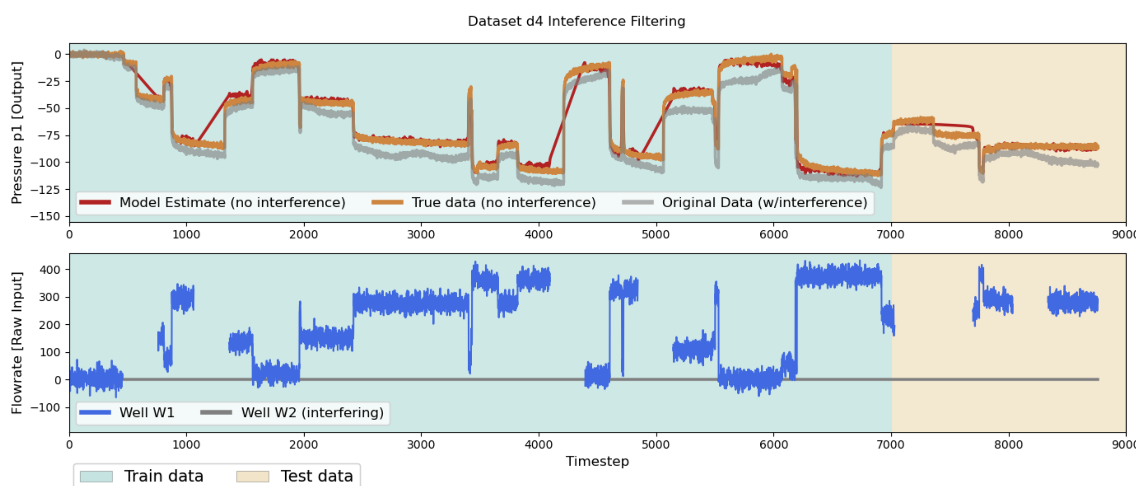

Figure 3.36: Model D: LassoNet with missing flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

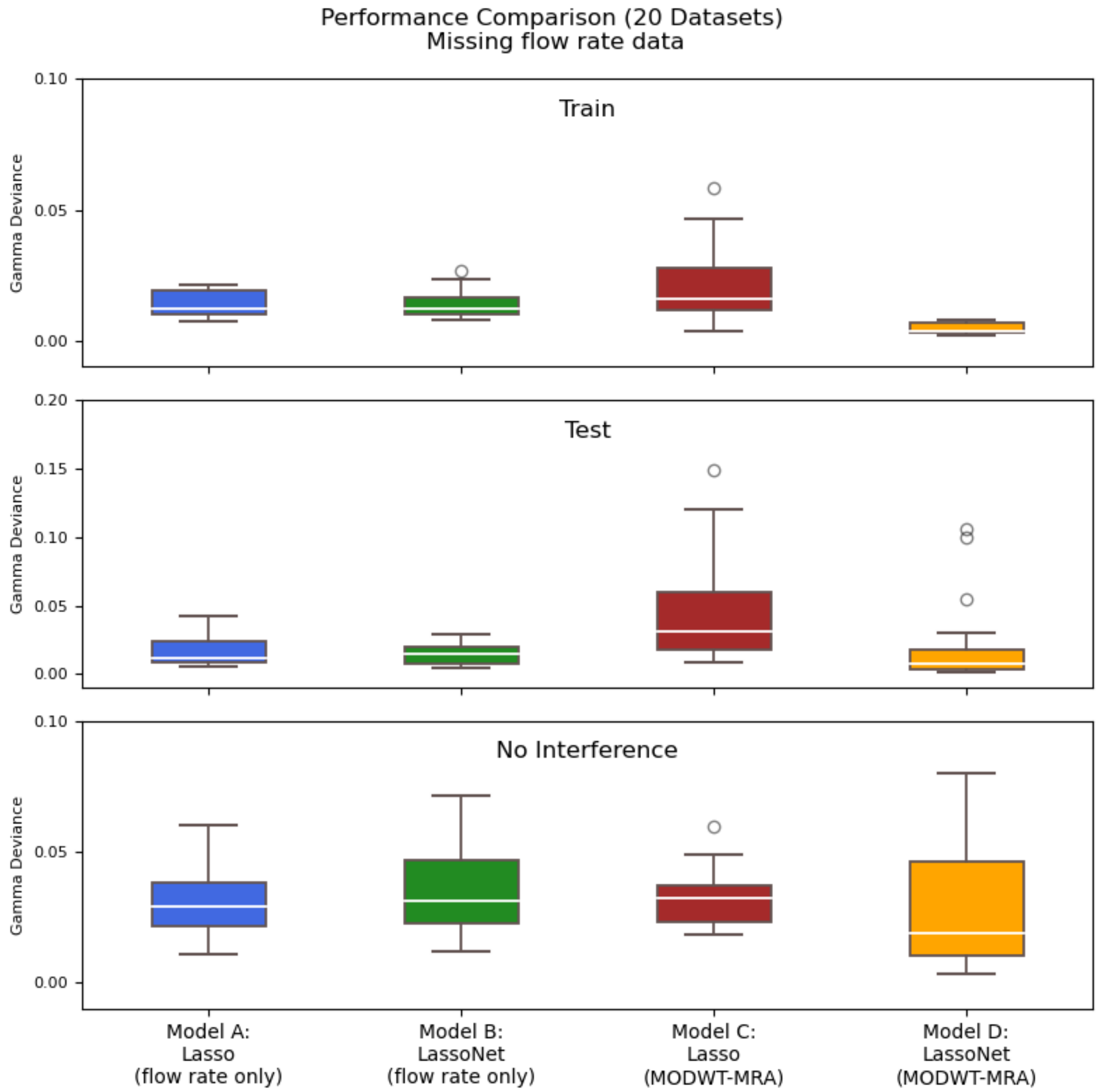

Figure 3.37: Models comparison for the two well scenario with missing flow rate. The top panel corresponds to training data performance, medium panel shows the test data performance and bottom panel shows the interference filtering results. A lower gamma deviance denotes better performance

3

3.5.7 Missing Flow Rate and Pressure Data

The last case for the two-well scenario is to have missing data in both flow rate inputs as well as the pressure training data. To create this case, the 20 datasets used in Section 3.5.1 were modified to include ten missing data gaps in pressure and flow rate for both wells. In a similar fashion to the previous section, the missing data gaps were of 300 hours of duration introduced at randomized points in the data. Table 3.11 shows the missing data statistics for the 20 datasets. Between 19% to 38% of the flow rate data is missing per well in each dataset and the missing pressure data ranges between 16% to 38%. Because of data fragmentation, the fractions of full data for each dataset are only between 23% to 40%.

Table 3.11: Fraction of total missing flow rate and pressure data for each of the synthetic data sets.

Dataset	Missing Flow Rate W1	Missing Flow Rate W2	Missing Pressure W1	Full Data
d1	0.27	0.30	0.26	0.24
d2	0.28	0.35	0.26	0.38
d3	0.25	0.36	0.16	0.38
d4	0.25	0.19	0.34	0.39
d5	0.32	0.26	0.29	0.32
d6	0.25	0.33	0.34	0.35
d7	0.27	0.34	0.30	0.33
d8	0.26	0.37	0.21	0.34
d9	0.27	0.27	0.31	0.40
d10	0.30	0.35	0.29	0.35
d11	0.29	0.33	0.29	0.32
d12	0.31	0.35	0.34	0.25
d13	0.30	0.38	0.36	0.34
d14	0.37	0.38	0.24	0.31
d15	0.33	0.30	0.35	0.38
d16	0.33	0.30	0.22	0.38
d17	0.35	0.31	0.28	0.34
d18	0.38	0.34	0.32	0.23
d19	0.36	0.38	0.21	0.32
d20	0.27	0.29	0.38	0.34

Figures 3.38 to 3.41 show the results of the two-well scenario with missing flow rate and pressure.

Overall, a similar trend appears for all four models. Models A and B show a noisy but unbiased prediction for times where the flow rate is available for both wells. Models C and D which use the flow rate MODWT-MRA as input have less noisy estimates. Model C however is noticeably more noisy than model D in this example, with model D having the closest fit and less noise of all four models.

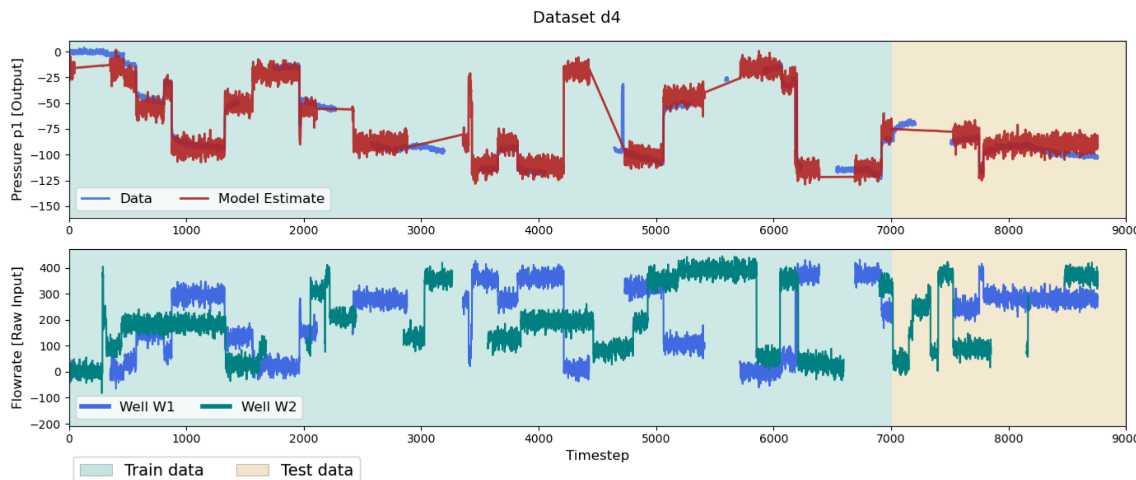

Figure 3.38: Model A: Lasso regression with missing gaps in both pressure and flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate.

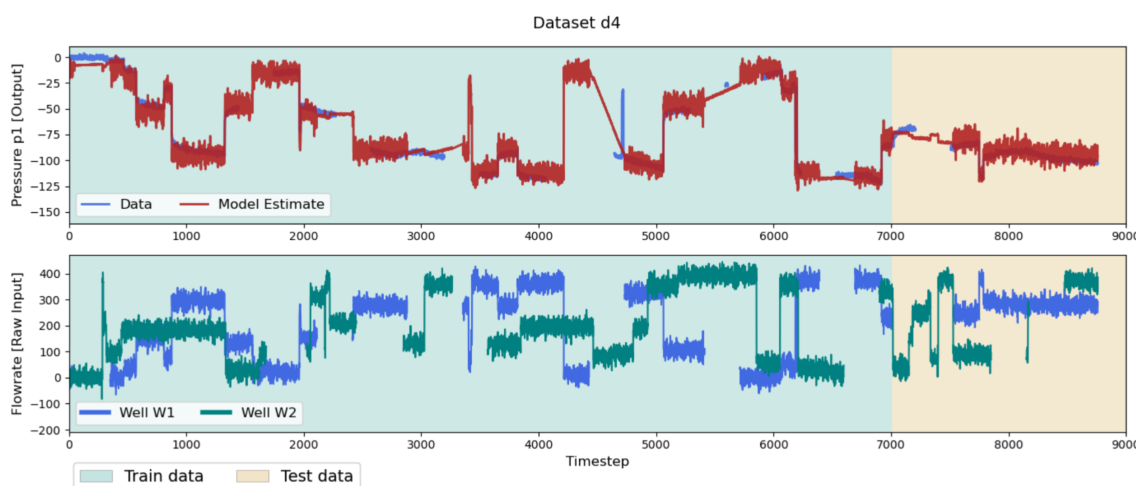

Figure 3.39: Model B: LassoNet with missing gaps in both pressure and flow rate data. Top panel: Pressure estimates. Bottom panel: Unprocessed flow rate.

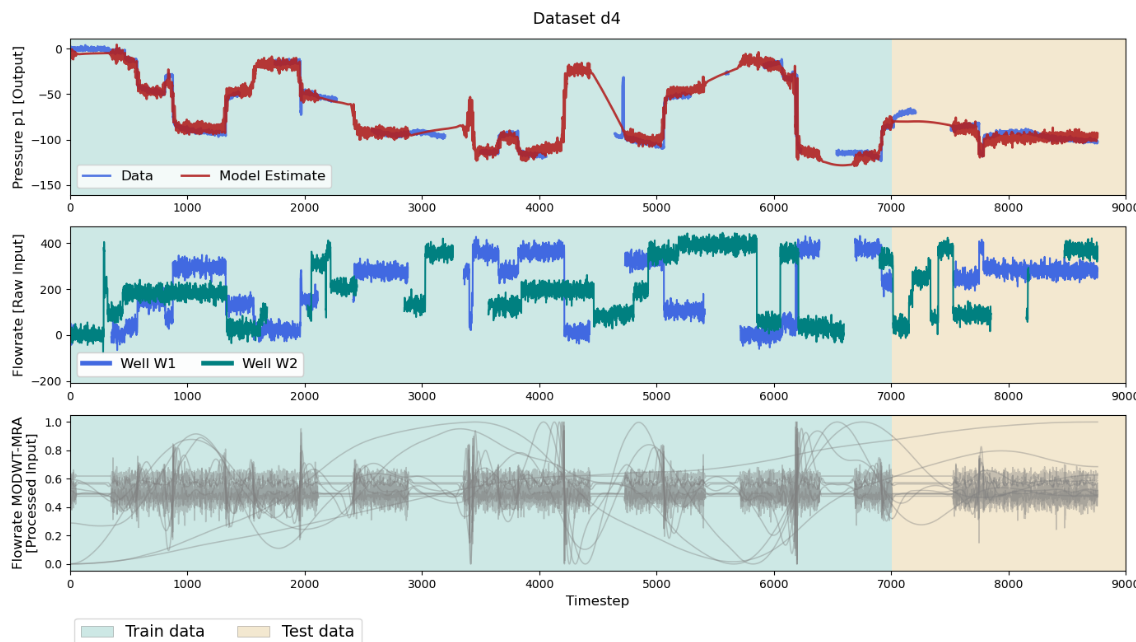

Figure 3.40: Model C: Lasso regression with missing gaps in both pressure and flow rate data. Top panel: Pressure estimates. Center panel: Unprocessed flow rate. Bottom panel: Flow rate MODWT-MRA.

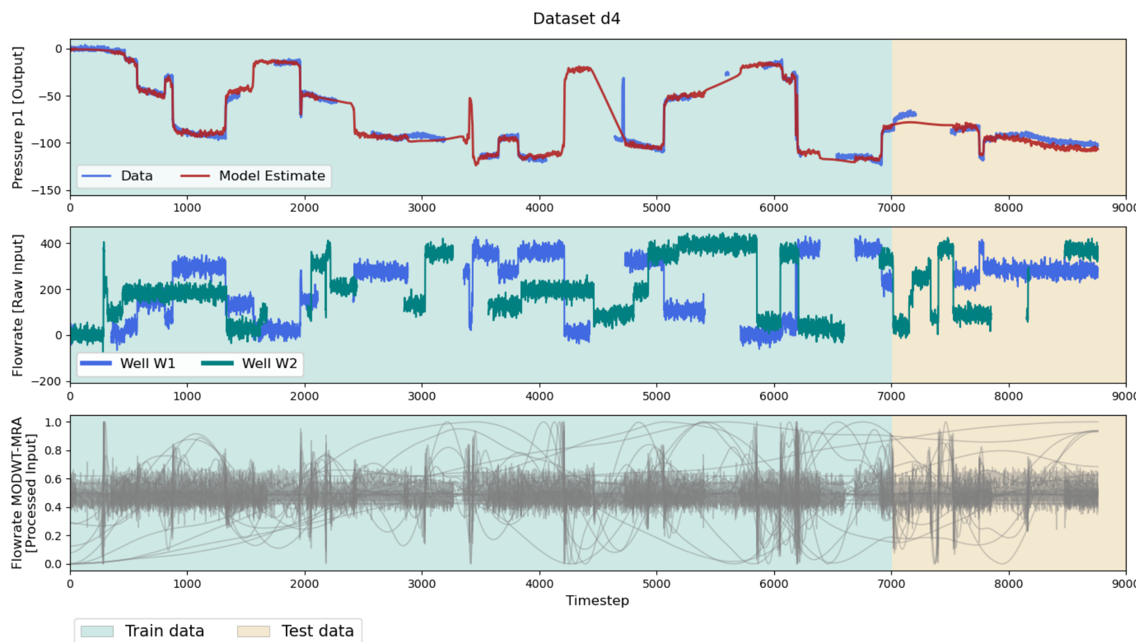

Figure 3.41: Model D: LassoNet with missing gaps in both pressure and flow rate data. Top panel: Pressure estimates. Center panel: Unprocessed flow rate. Bottom panel: Flow rate MODWT-MRA.

Interference Filtering with Missing Flow Rate and Pressure Data

In a similar fashion to Section 3.5.6, the four models performance was tested against in the no interference scenario. Figures 3.42 to 3.45 show the results for all four models A-D. Models A, B and C failed to successfully capture the self response of the target well when the interfering well W2 was shut. Both models A and B display a high noise estimate that masks the interference effect and model C even if less noisy than models A and B still shows large amounts of noise, especially when compared with the scenario with no missing data. Only model D, the LassoNet with flow rate MODWT-MRA inputs was successfully in capturing the target well's self response. Not only did model D was able to capture the self-response of the target well W1 but also achieved a minimum amount of noise. It must be noticed that the correct pressure estimates were achieved when flow rate data was available from the target well with straight continuous lines where no flow rate data was available.

As in Section 3.5.6, similar results appear throughout the 20 datasets. In this scenario, model D still achieved the smallest median gamma deviance throughout all 20 datasets. However, the distribution of gamma deviance became wider with the increase of missing data as shown in Figure 3.46

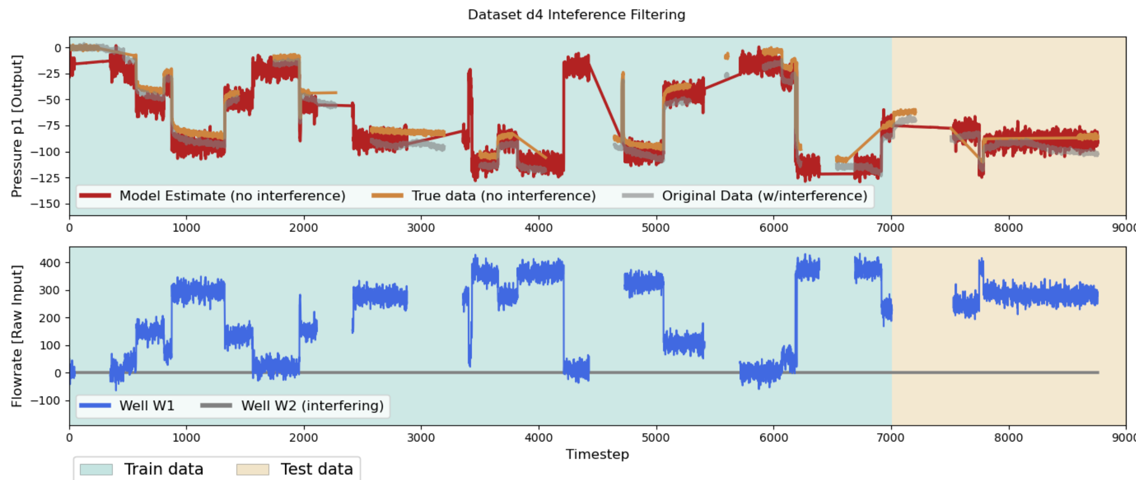

Figure 3.42: Model A: Lasso regression with missing gaps in both pressure and flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

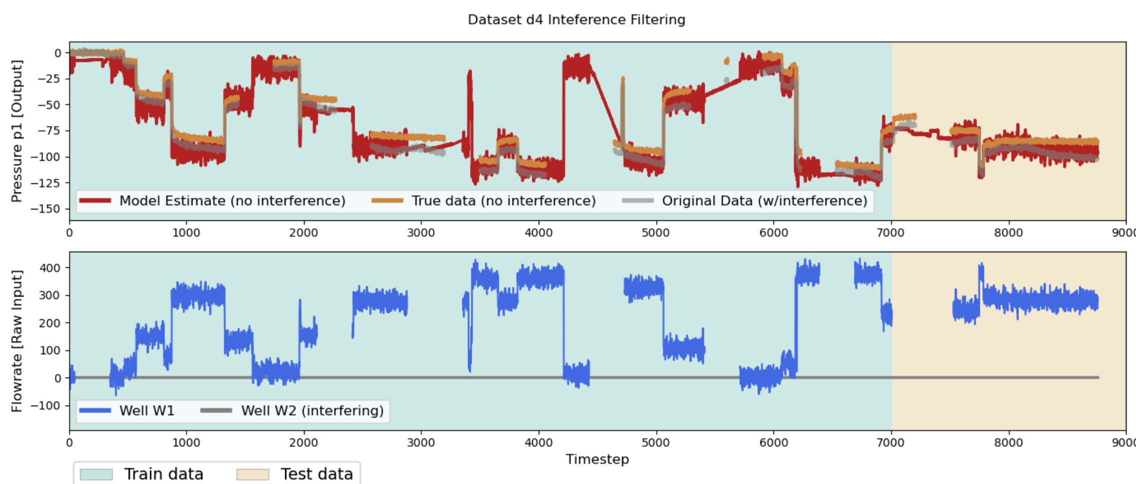

Figure 3.43: Model B: LassoNet with missing gaps in both pressure and flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

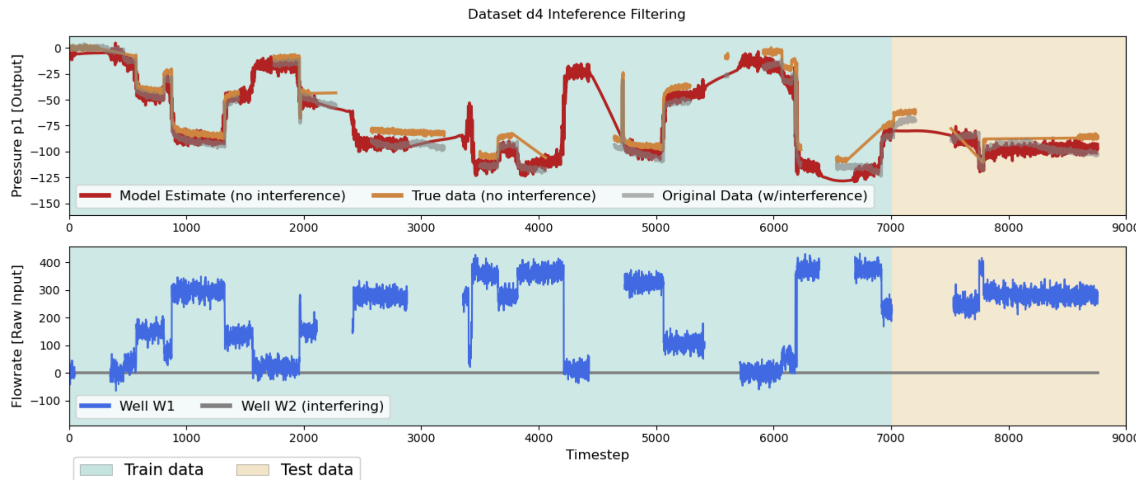

Figure 3.44: Model C: Lasso regression with missing gaps in both pressure and flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

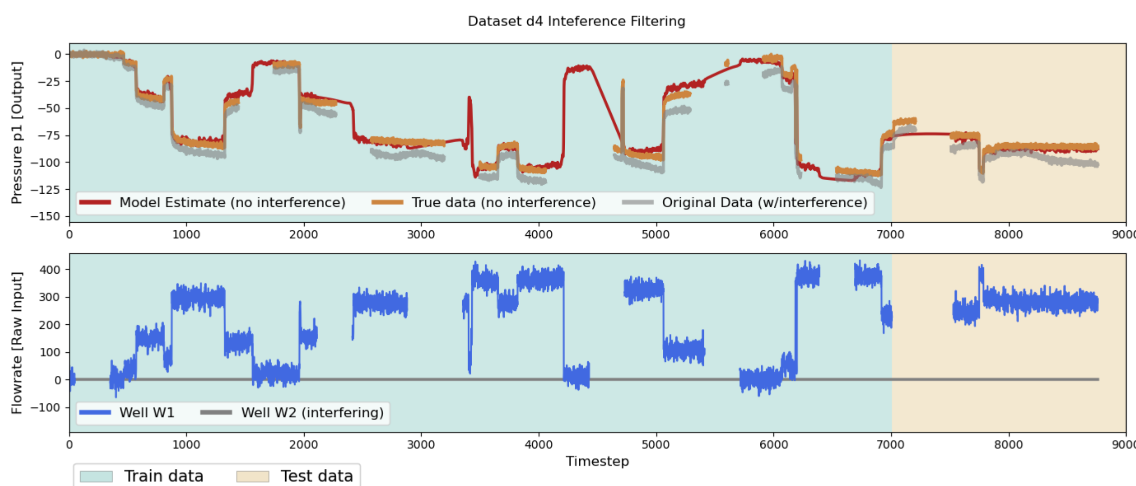

Figure 3.45: Model D: LassoNet with missing gaps in both pressure and flow rate data. The top panel displays the pressure estimates, the bottom panel shows the unprocessed flow rate.

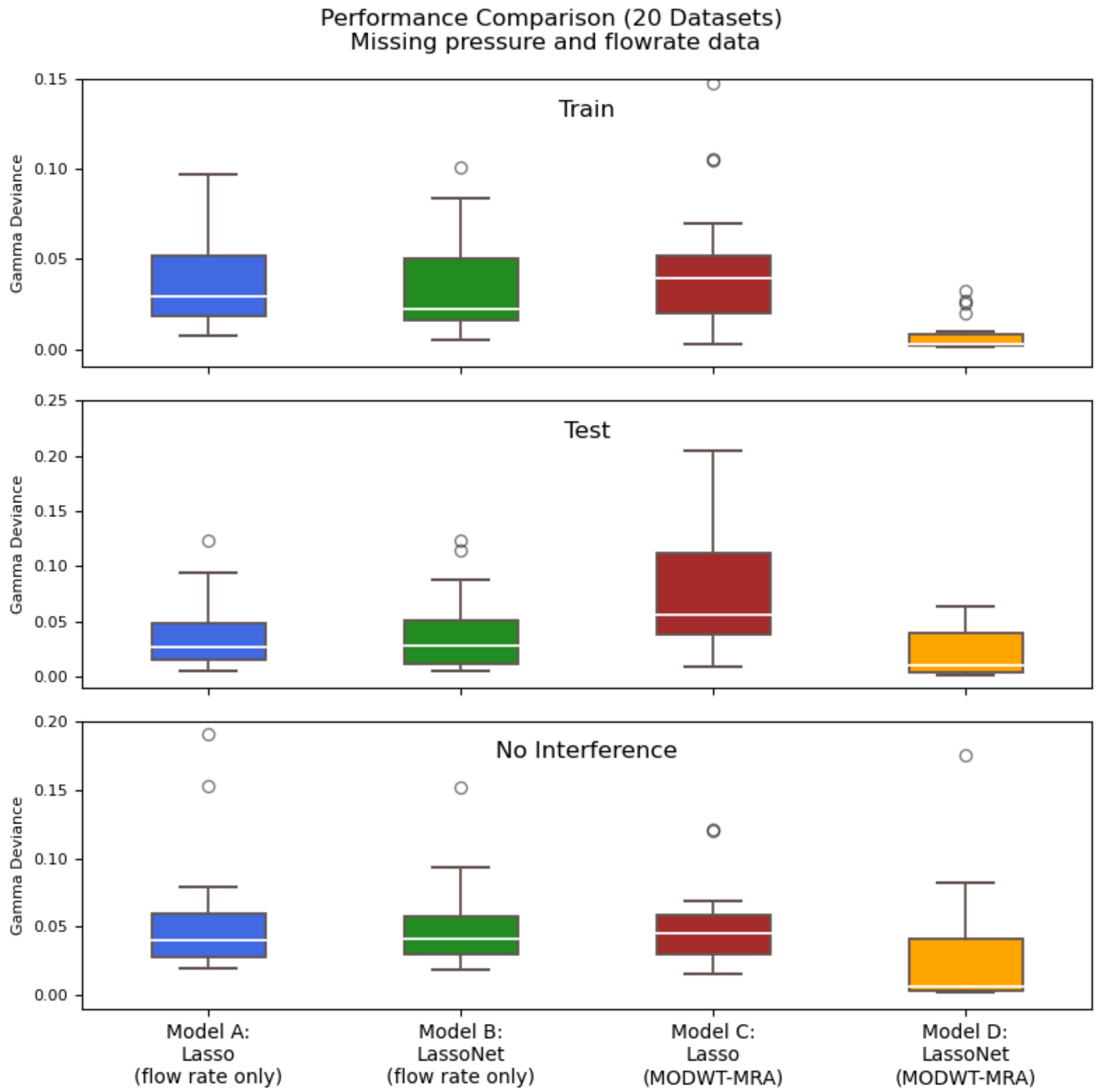

Figure 3.46: Models comparison for the two well scenario with missing flow rate and pressure data. The top panel corresponds to training data performance, medium panel shows the test data performance and bottom panel shows the interference filtering results. A lower gamma deviance denotes better performance

3.6 Summary

This chapter explored the use of a variety of machine learning and deep learning models for modeling pressure response of wells both in isolation and in the presence of others. Overall, the approach for all models was to use the MODWT-MRA decomposition framework to create useful features that simplify the learning task for the data-driven algorithms. As shown in Chapter 2, the MODWT-MRA decomposes data with complex behavior into a set of simpler datasets that in the particular case of flow rate and pressure pairs are analog to virtual wells. The Lasso and LassoNet methodologies were tested in two overall scenarios, single well and two-well. For each scenario, the regression methodologies were tested with and without the use of the wavelet decomposition as inputs. Finally, all scenarios were tested with varying amounts of missing data.

Coupling the flow-rate MODWT-MRA as an input with shrinkage methods such as the Lasso and LassoNet resulted the ability to capture pressure response even with imperfect data. For the single well scenario, the main advantage of using the MODWT-MRA decomposition resulted in a significantly lower noisy estimate and higher accuracy even when using linear regression and in the presence of missing data. The MODWT-MRA is able to contain the uncertainty caused by missing data within a few levels of the decomposition.

Moveover, it is in the two-well scenario where the main benefits of using the wavelet decomposition came to light. A model that successfully captured an individual well's pressure response in the presence of interference was developed by pairing the MODWT-MRA flow rate features with the LassoNet. Using linear methods such as the Lasso or not including the MODWT-MRA flow rate inputs did not give successful results. Moreover, the model was still successful when dealing with high data fragmentation of up to 60%.

Chapter 4

Modeling Flow Rate

4.1 Introduction

Flow rate reconstruction is the process of estimating a flow rate schedule from the available pressure history. This is often a goal in cases where equipment such as flow meters have failed or operational constraints caused unrecorded data periods. Building a data-driven model to model flow rate using pressure history as input is the opposite problem to that of Chapter 3. From a methodological point of view it is similar in the sense that a function mapping is generated using data, in this case using pressure as the input and flow rate as the output. However, from a physical standpoint, the problem is very different as it is an inverse problem and the pressure data used as input for the model has gone through a process of information loss caused by diffusion.

This chapter explores the applicability of the MODWT-MRA decomposition for the purpose of modeling flow using data-driven models. First, a single well scenario was built and tested with varying amounts of missing data and two data modeling methodologies, the Lasso and LassoNet. This first scenario served as the building block for the two well scenario, where data from two connected wells were used to reconstruct the flow rate history of one of them. The additional challenge of this scenario is caused by interference effects present in the pressure histories. In this case, a data-driven model needs not only to map the reverse relationship of flow rate and pressure but also to identify and filter out the external interactions present in the input data. The two well scenario also was tested with varying amounts of missing data, making the learning problem even more challenging.

4.2 Single Well Scenario

A single well scenario is the simplest case for testing a flow rate recovery algorithm. In this scenario, the pressure history of a well in isolation is used to generate a flow rate history estimate. Because the objective of this exercise is to evaluate the usefulness of the MODWT-MRA decomposition when designing a flow rate model, two sets of input features are compared: a base case of using the unprocessed pressure history and the MODWT-MRA decomposition of flow rate. Similar to Chapter 3, two modeling methodologies were tested with each set of pressure features, the Lasso and LassoNet.

4.2.1 Data Generation

For the single well scenario, the set of 20 data sets used in Section 3.4 were used. Each dataset consists of a single well flow rate and pressure data. The flow rate data for the well was generated as a randomized step function with up to 35 step changes and a maximum value of 400 STB/day. The sampling frequency of the data was set to one observation per hour and a total duration of one year. The pressure solutions were obtained analytically in Laplace space and subsequently subjected to numerical inversion using the Gaver-Stehfest algorithm [Jacquot et al., 1983]. A total of 20 data sets were constructed with different flow rate sequences each to test the methodology in a variety of production scenarios. The 20 data sets shared the same reservoir model, which includes wellbore storage, skin effect, infinite-acting radial flow and a constant pressure boundary. Moreover, Gaussian noise of 5% and 1% was added to both flow rate and pressure datasets respectively. The reservoir was assumed homogeneous with parameters summarized in Table 4.1.

Table 4.1: Reservoir parameters for the synthetic data sets

Parameter	**Value**
k	100
S	0
C (STB/psi)	0.001
μ (cp)	1
h (feet)	50
ϕ	0.2
c_t (/psi)	5e-6
r_w (feet)	0.208
r_e (feet)	10,000
B	1

4.2.2 Model Input and Features Development

In an analogous way to Chapter 3, to create the input features of the flow rate reconstruction models, the MODWT-MRA was applied to the unprocessed pressure data. The feature generation process consists of the following steps:

1. Select pressure data to be used for modeling.
2. Split datasets into training, validation and testing if applicable.
3. Determine the maximum decomposition level J_0. The depth of the decomposition must be applicable to all training, validation and testing datasets.
4. Apply the J_0th level MODWT-MRA separately to each of the datasets.

In this exercise, for all scenarios the decomposition depth parameter J_0 was set to 12, resulting 13 total features from each individual pressure history. An example of the resulting pressure decomposition can be seen in Figure 4.1.

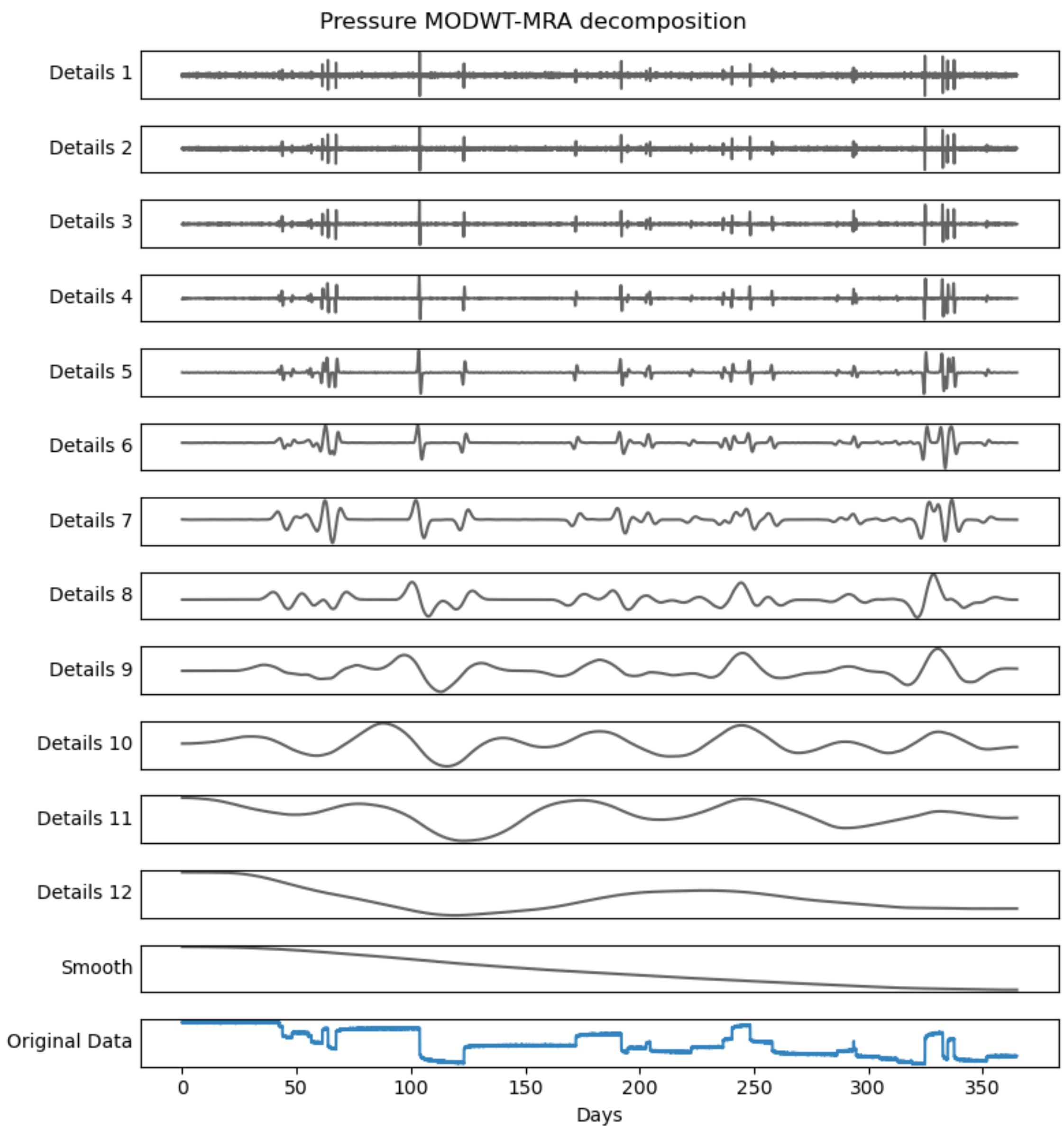

Figure 4.1: MODWT-MRA decomposition of pressure history

4.2.3 Model Specifications

Four model specifications were fitted for the 20 synthetic datasets. Two of the models were based on the Lasso and two used the LassoNet for the fitting. In all cases, the train and validation sets used 80% of the data with the last 20% reserved for the test set. The baseline scenarios used pressure data as the only model feature. For the MODWT-MRA models, the wavelet filter was the Haar filter and the depth parameter J was set to 12 for a total of 13 features. Table 4.2 contains the different names and specifications for all four models, referred as models A through D.

Table 4.2: Model descriptions for the two-well scenario with missing flow rate data

Model Name	Regression Method	Features	Target
AF	Lasso	Pressure History	Flow Rate
BF	LassoNet	Pressure History	Flow Rate
CF	Lasso	MODWT-MRA of Pressure History	Flow Rate
DF	LassoNet	MODWT-MRA of Pressure History	Flow Rate

LassoNet Specifications

Both models BF and DF which use the LassoNet for fitting the data have a shared set of training hyperparameters and hidden layer architecture. For both models, a hidden layer of size 1x100 was defined for the fully connected part of the LassoNet. The input layer varied in size due to the difference in number of features and the output of both models was a vector containing the flow rate estimates. The regularization hyperparameter λ was selected through cross validation for each dataset, however the initial λ and regularization path (search space) was shared among the two models. The search space for λ was defined as a geometric progression of 150 points spanning from $\lambda = 400$ to $\lambda = 7,000$.

Both models BF and DF also share the hierarchy hyperparameter M that controls the relative importance of the linear and nonlinear components in the LassoNet. A value of $M = 0$ results in a Lasso regression while $M \to +\infty$ results in an unregularized fully connected neural network. For both models BF and DF the value was kept constant as $M = 200$. The experience gained using the LassoNet revealed similar results at values of $200 < M < 300$ with the larger parameter sensitivity being λ. For this reason and to save computational time, M was not part of the cross-validation parameter optimization. The optimizer used for the neural network training was Adam with a learning rate of 0.01. An early stopping strategy was used for training, halting the training when a least 10 training epochs had passed without a minimum improvement of 1% in the validation set.

Performance Metric

To compare model performance across different datasets, gamma deviance was chosen as the performance metric. This is not to be confused with the training loss function, which for the Lasso is the L1 norm and for LassoNet is least squares. Because the synthetic datasets contain a variety of ranges in flow rate histories , a metric that could be compared across scales becomes necessary. Gamma deviance is scaling-invariant derived from the more general Tweedie deviance. In fact, both mean squared error and gamma deviance are both special cases of the Tweedie deviance [Dunn and Smyth, 2008], [Dunn and Smyth, 2005]. Equation 4.1 shows the definition for the gamma deviance:

$$D(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 2 \left(\log(\hat{y}_i/y_i) \ + \ y_i/\hat{y}_i \ - \ 1 \right) \tag{4.1}$$

4.2.4 Performance Results

After training all four models for each of the 20 datasets, similar results were obtained for all four models AF to AD in the single well scenario with no missing data. Figures 4.2 to 4.5 show the results for all the four models for dataset d2, chosen for its representative results. Model AF displays results centered in the true flow rate data and with less noise than the original data due to the lower noise ratio in the input pressure data. However, due to the linear nature of the model, the sharp step-like transitions in flow rate cannot be achieved completely and the estimates display a slight but noticeable error at each flow rate step change. Model BF (Figure 4.3) shows a flow rate estimate centered in the correct values and also with a less noisy output than the original pressure data. The step transitions in the flow rate estimates are sharper than those of model AF but still display residual effects of the pressure diffusion. Figure 4.4 shows the pressure estimates for Model CF, the Lasso with the pressure MODWT-MRA decomposition as input. Overall, it has similar results to other models showing an accurate prediction but more sharp transitions in the flow rate estimates. Model DF (Figure 4.5), the LassoNet with pressure MODWT-MRA as input, has the least noisy estimate and the sharpest transitions in the flow rate estimates. Analyzing the results from a single datasets it is difficult to identify a model that consistenly shows better performance. However, observing the gamma deviance distribution for all 20 datasets in Figure 4.6, it can be seen that models CF and DF, which incorporate the MODWT-MRA decomposition as input have the lowest deviance and thus best performance. Model CF, the Lasso with MODWT-MRA pressure features has the smallest median gamma deviance for the test sets as well as the least dispersion across datasets. However, all models seem to have adequate performance for this simple one well scenario.

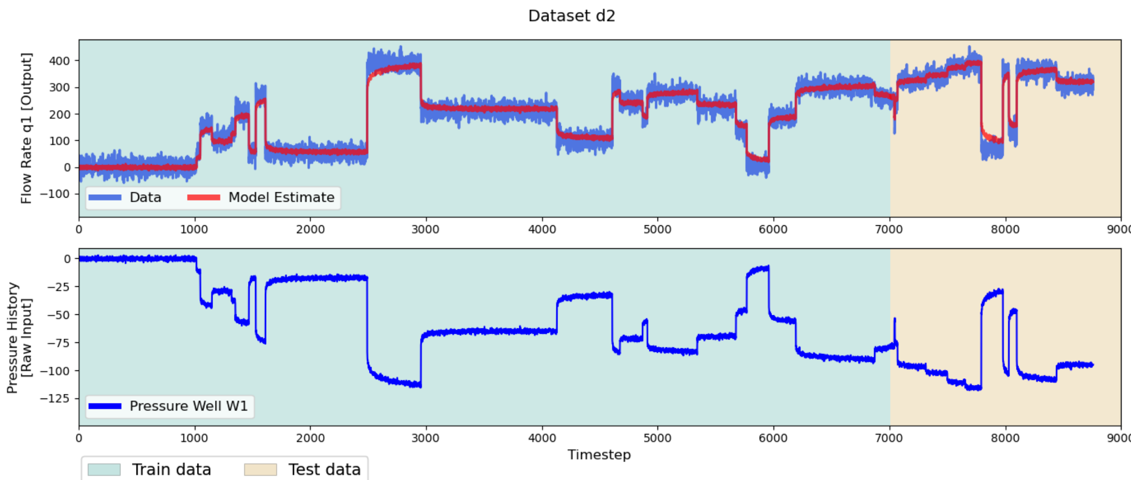

Figure 4.2: Model AF: Lasso regression with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

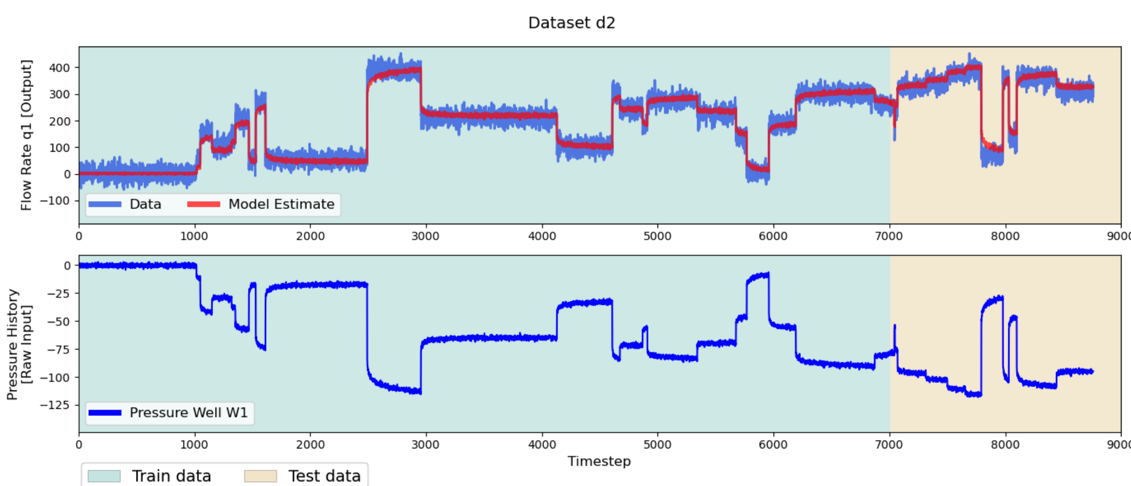

Figure 4.3: Model BF: LassoNet with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

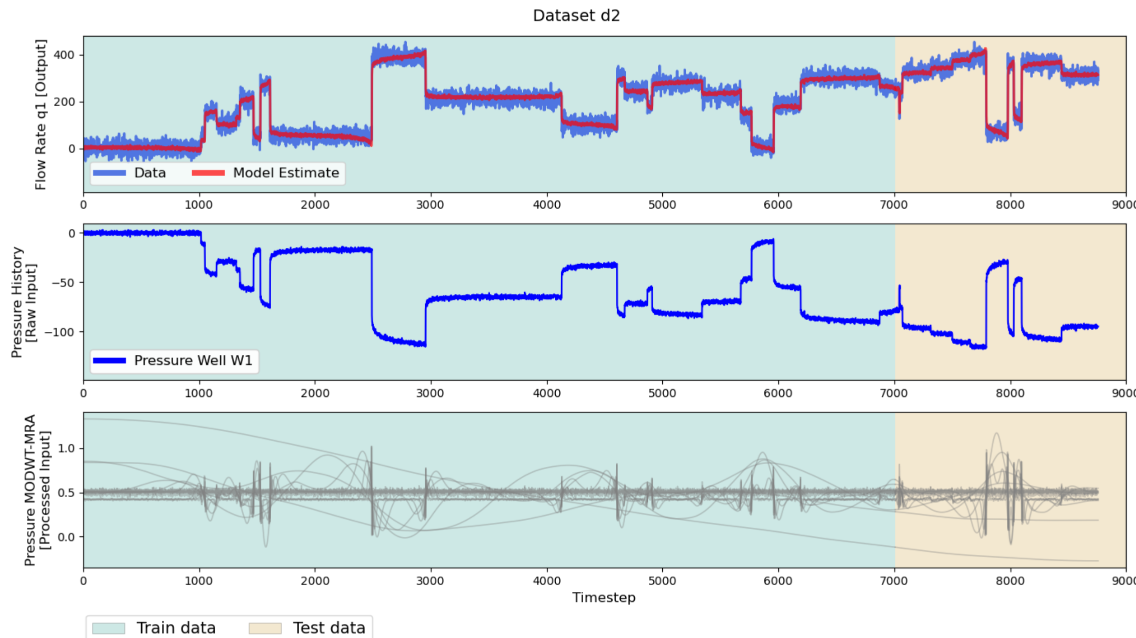

Figure 4.4: Model CF: Lasso regression with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

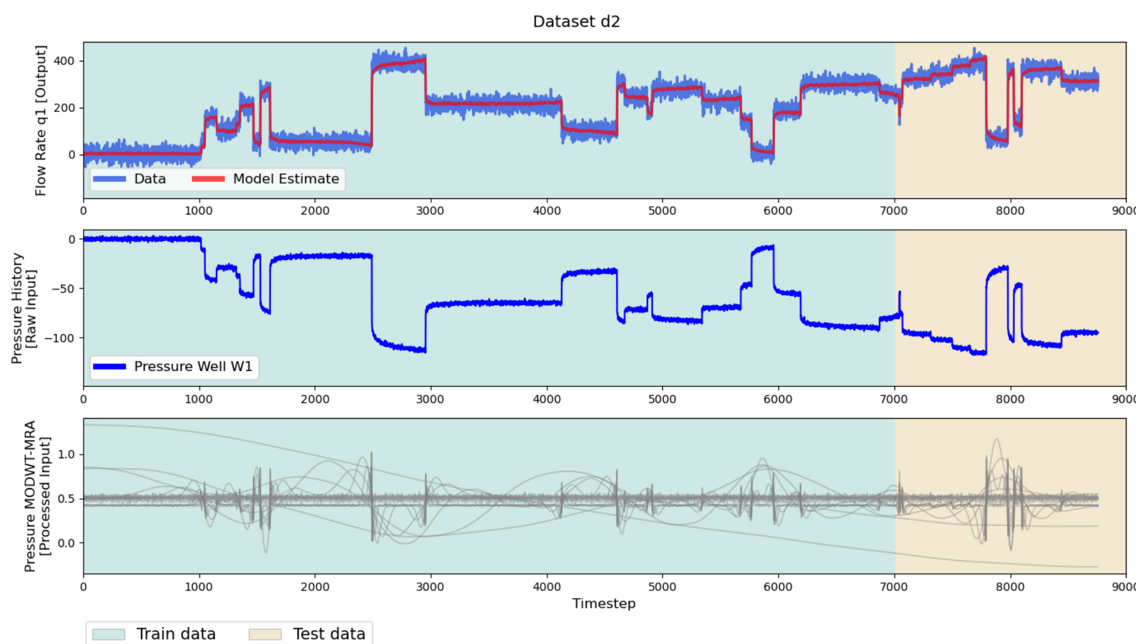

Figure 4.5: Model DF: LassoNet with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

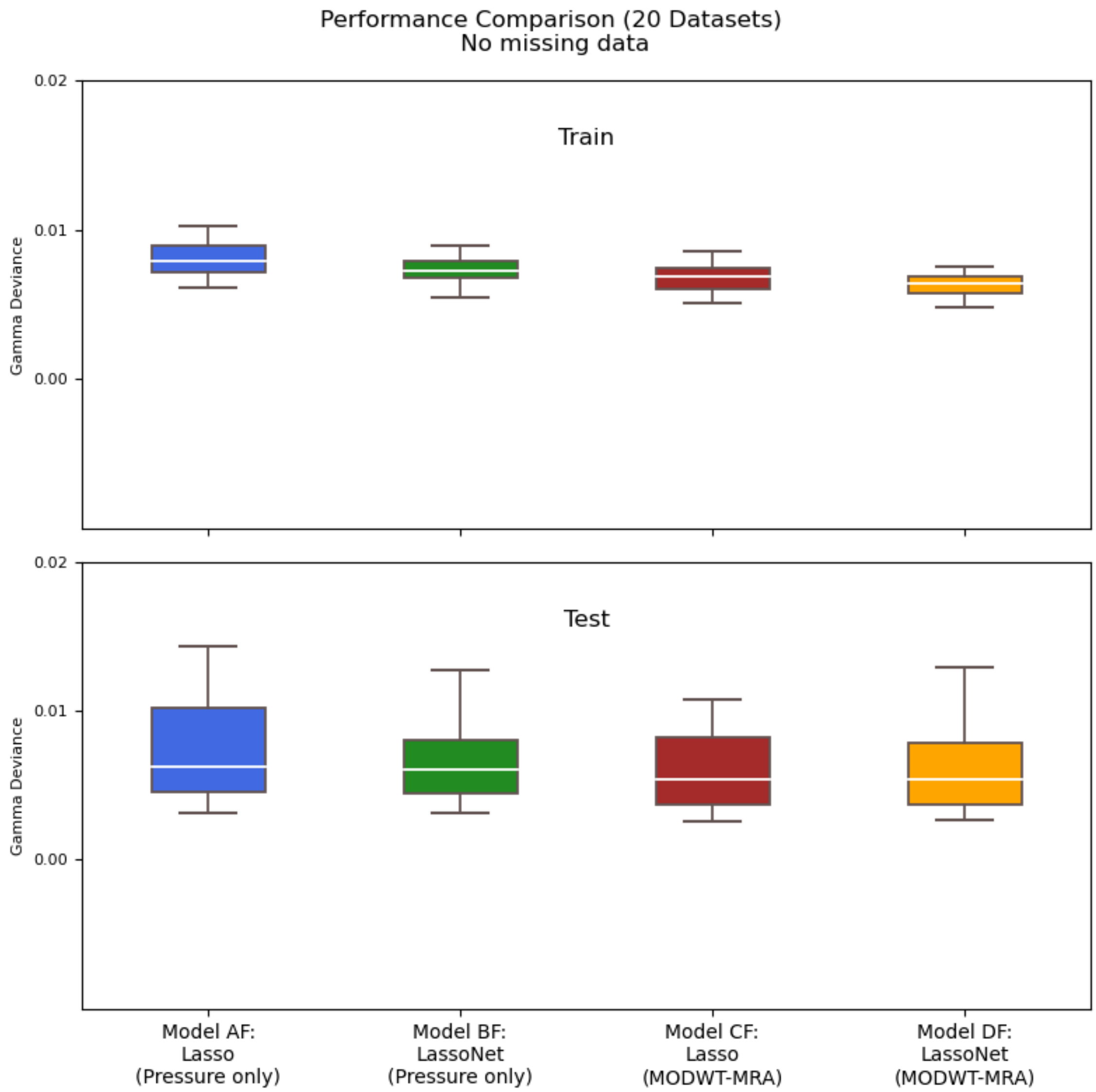

Figure 4.6: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

4.2.5 Missing Flow Rate Data

The scenario with missing flow rate data is a more challenging and realistic application of flow-rate reconstruction. In this scenario, the pressure history is assumed to be complete and the goal is to fill in the missing gaps in flow rate. For this scenario, the same set of 20 datasets was used as in Section 4.2.1 but with missing data gaps. To create the missing gaps in the flow rate schedules of the 20 datasets, 10 contiguous periods of 300 hours in duration were deleted at random points in the time series. This results in no single dataset containing the same missing data points. The inserted missing data gaps can overlap with each other creating longer duration gaps. The total percentage of missing flow rate data per dataset ranged between 15% to 35% as noted in Table 4.3. Pressure data were not modified for any of the datasets.

The same set of four models AF to AD was tested as in the previous section. To successfully use the Lasso or LassoNet with missing gaps and to be able to apply the MODWT-MRA, a linear

Table 4.3: Fraction of total missing flow rate data for each of the synthetic data sets.

Dataset	Missing Flow Rate %
d1	0.25
d2	0.30
d3	0.25
d4	0.23
d5	0.16
d6	0.21
d7	0.34
d8	0.29
d9	0.29
d10	0.34
d11	0.33
d12	0.35
d13	0.30
d14	0.29
d15	0.20
d16	0.30
d17	0.30
d18	0.32
d19	0.29
d20	0.35

interpolation was used to fill in the missing flow rate data. Figure 4.3 shows five of the 20 datasets where the missing data gaps were introduced for flow rate.

For models AF and AC which use the Lasso, the linear interpolation in flow rate data introduces a large bias for the training. To accommodate for this, a weighted Lasso was used where only the complete flow rate periods were used for the fitting process. Specifically, the following transformation was applied to the data for model AF:

$$\begin{aligned} y_{trans} &= \sqrt{w} * y \\ x_{trans} &= \sqrt{w} * x \end{aligned} \tag{4.2}$$

where w is the weight vector, and $w_i = 1$ when flow rate data is available and $w_i = 0$ otherwise. For LassoNet models AB and AC the weight vector w was passed directly to the loss function without

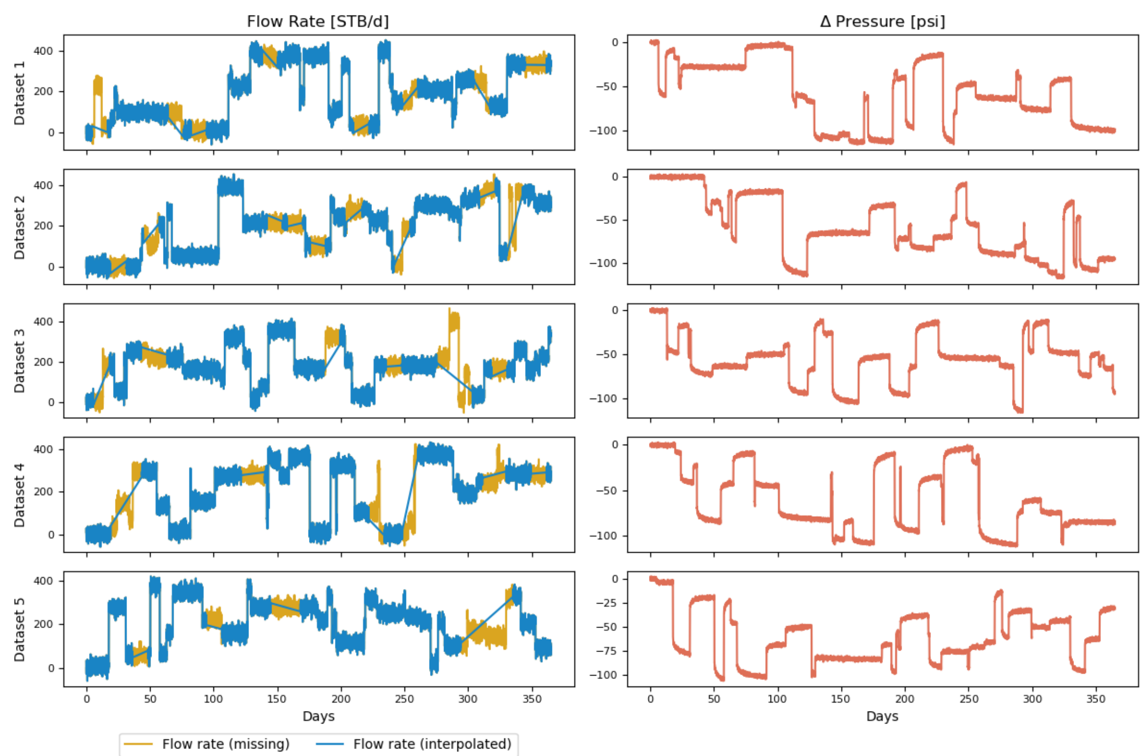

Figure 4.7: Datasets with missing flow rate data.

the need for any data transform.

Table 4.4: Model descriptions for the single-well scenario with missing flow rate data

Model Name	**Regression Method**	**Features**	**Target**
AF	Lasso (weighted)	Pressure	Flow Rate (interpolated)
BF	LassoNet		
CF	Lasso (weighted)	Pressure MODWT-MRA	
DF	LassoNet		

Figures 4.8 and 4.9 show the results for models AF and BF corresponding to the Lasso and LassoNet with unprocessed pressure features. Both models display accurate flow rate reconstruction with less noise than the original data. As in Section 4.2.4, the step changes in flow rate are not as sharp as the true data, reflecting a curved response that resembles the pressure transient. The use of weighted Lasso in model AF was fundamental in achieving reasonable flow rate reconstruction. Without weights in model AF, the obtained estimate is a constant line with the average flow rate throughout the period.

Model CF, shown in Figure 4.10 is the Lasso with pressure MODWT-MRA inputs. As expected, the estimate is less noisy because of the Lasso's ability to filter out less relevant features such as the high frequency noise levels in the MODWT-MRA. Other than noise reduction the results do not show a significant improvement in the model performance compared with model AF.

Model DF shown in Figure 4.11 shows the LassoNet with pressure MODWT-MRA inputs. The results are similar to model CF in terms of noise reduction and the model is able to accurately reconstruct the flow rate signal.

Comparing models AF to DF across all 20 datasets gives a different perspective. Figure 4.12 shows the gamma deviance distribution for all 20 datasets and four models. It is clear that models AF and CF which use the Lasso have consistently lower performance compared to the models that use LassoNet. For this scenario, model BF has the lowest median gamma deviance for the test set and also the lowest spread throughout the 20 datasets. Model CF, the Lasso with MODWT-MRA pressure features has the worst median deviance and also the largest dispersion.

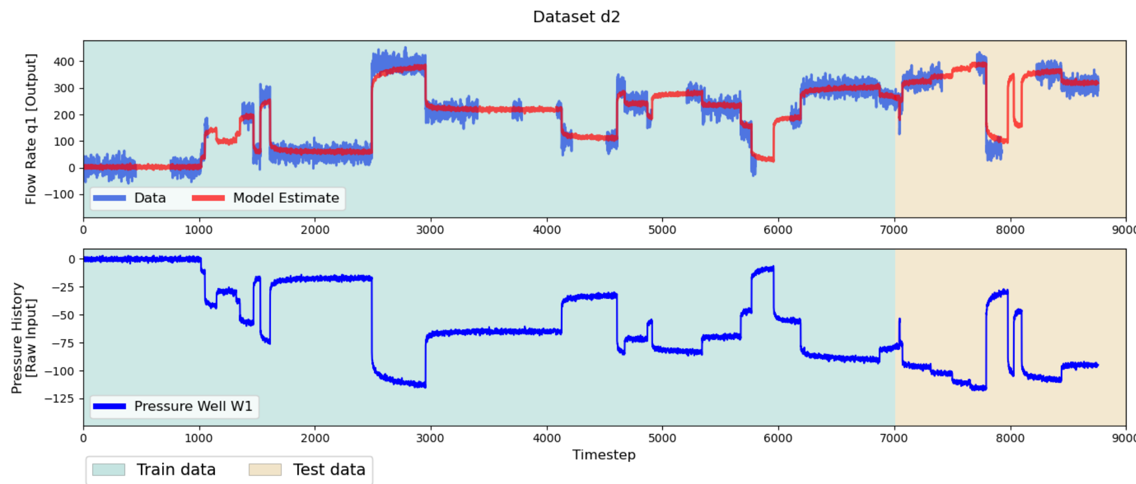

Figure 4.8: Model AF: Lasso regression with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

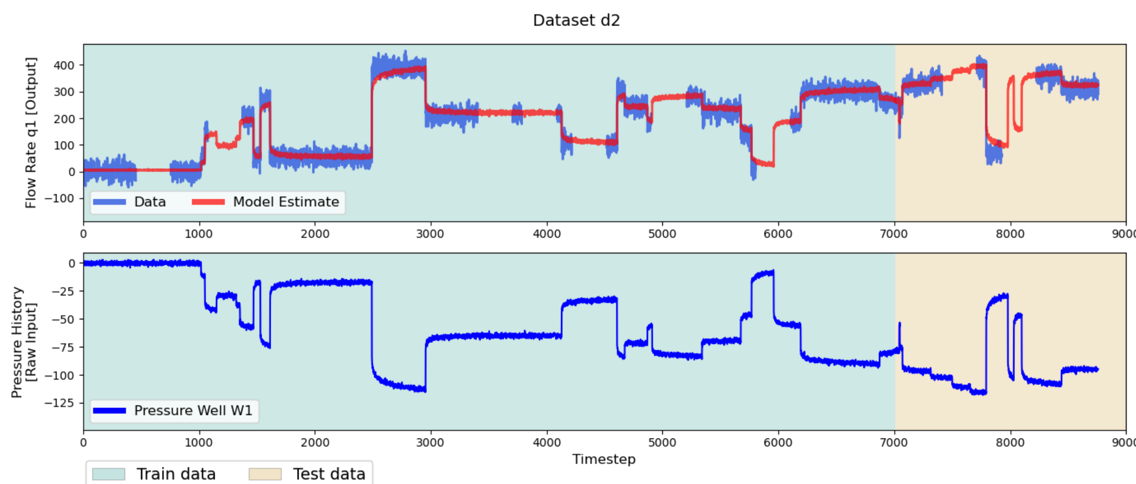

Figure 4.9: Model BF: LassoNet with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

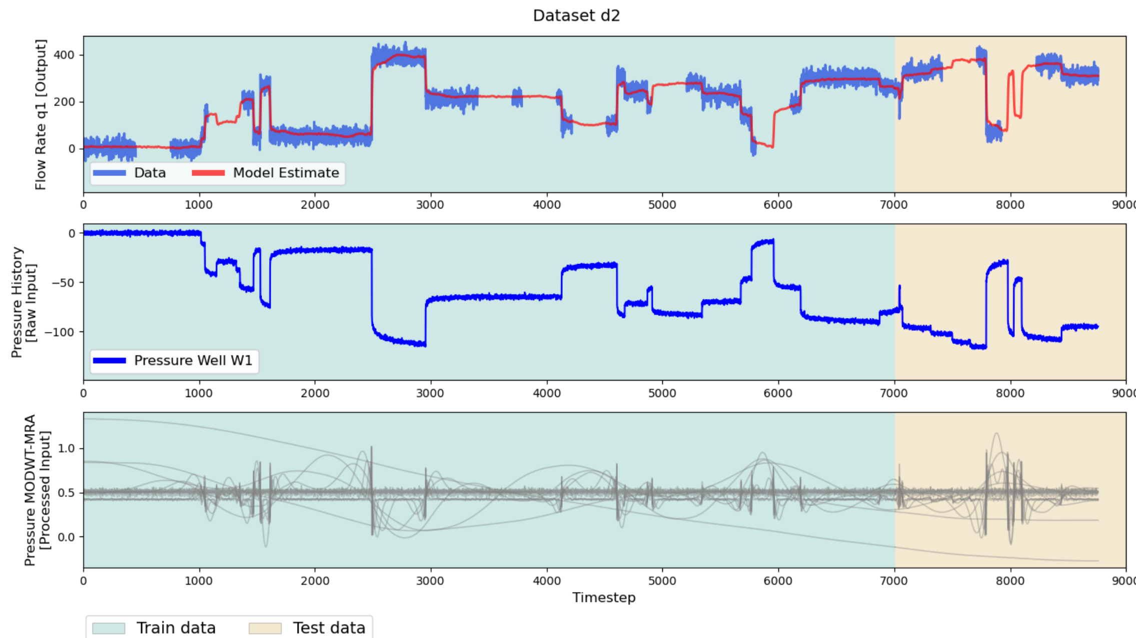

Figure 4.10: Model CF: Lasso regression with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

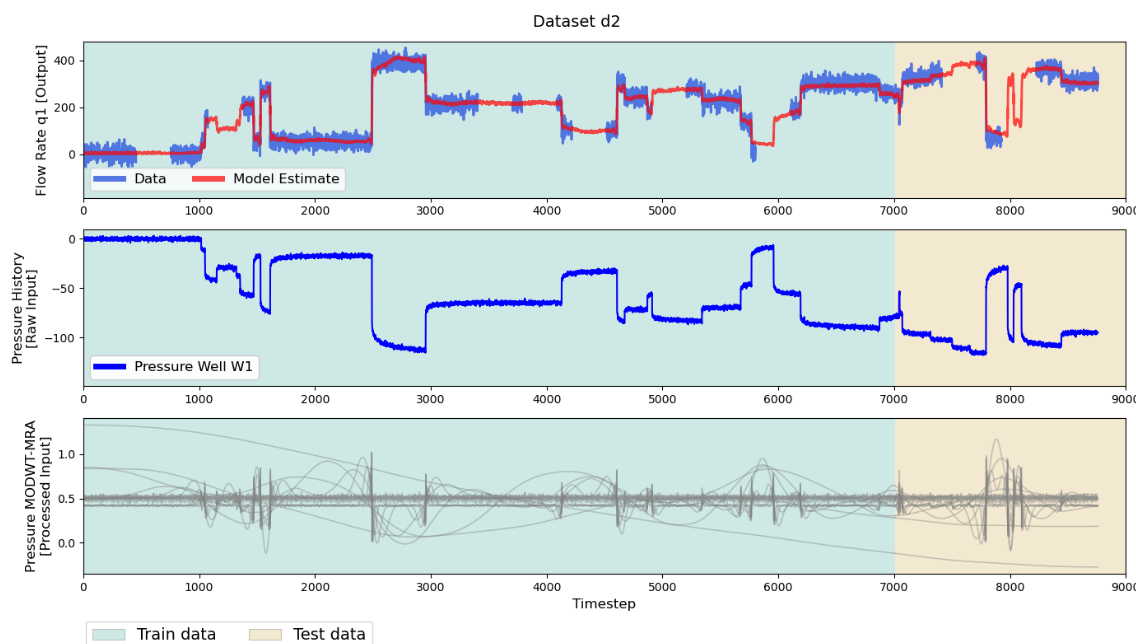

Figure 4.11: Model DF: LassoNet with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

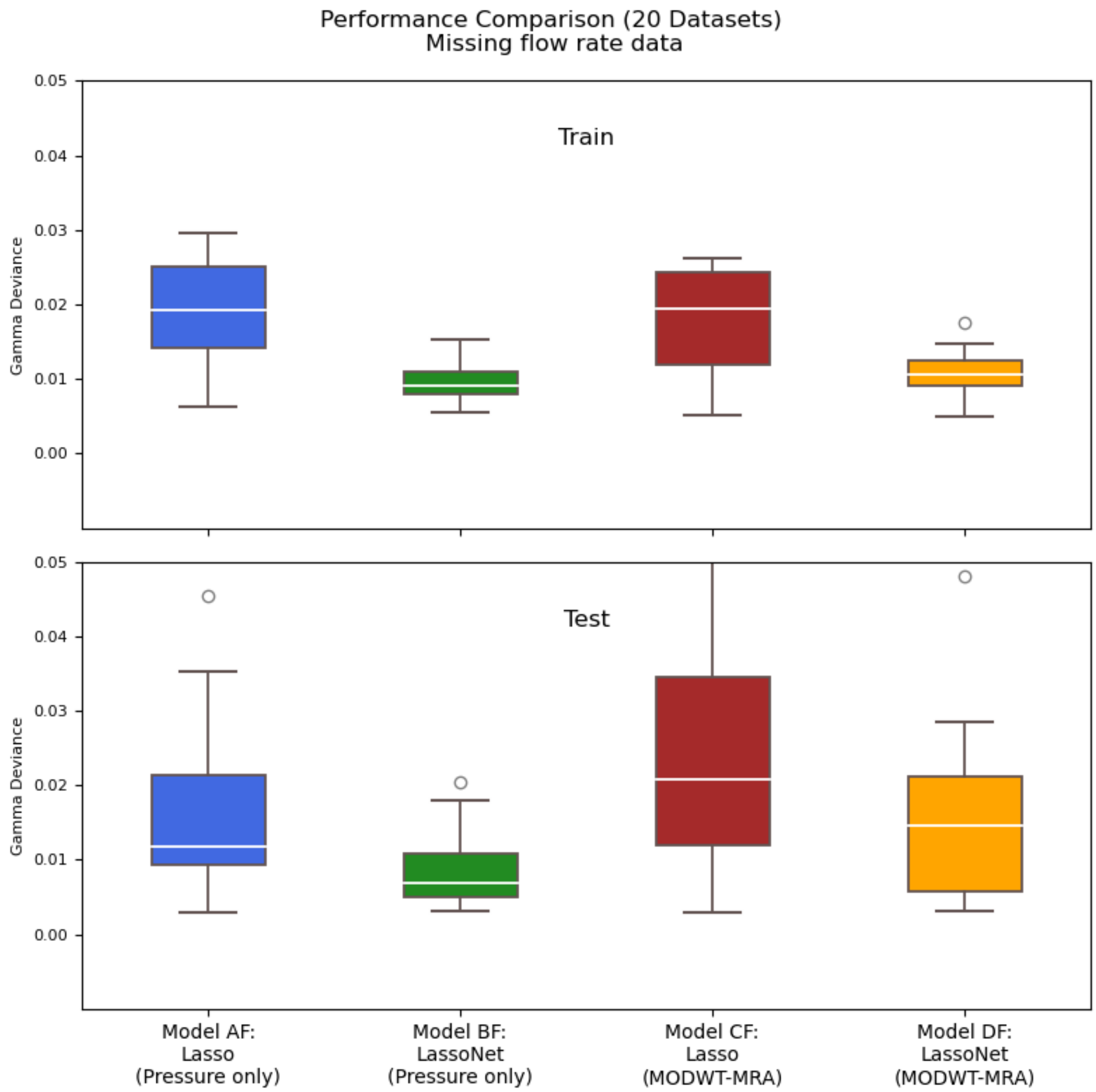

Figure 4.12: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

4.2.6 Missing Flow Rate and Pressure Data

The most complex case in the flow rate recovery single well scenario is that of missing data in both pressure inputs and flow rate outputs. The increased complexity of the scenario comes from the fragmentation of the data, as both features and target variables are incomplete. This fragmentation decreases the amount of effective data from which models can learn.

For this scenario, the strategy of introducing missing data gaps consisted of deleting periods of 300 consecutive hours from both flow rate and pressure data. Both variables were handled independently, but overlapping missing periods in both pressure and flow rate appeared as well. All 20 datasets were processed in this manner and no single dataset or variable within a dataset has the same missing data gaps. Table 4.5 shows relevant completeness statistics for the resulting datasets. For each variable (flow rate or pressure) the total missing fraction is between 17% to 38% per variable. However, when looking at the fraction of the data that is fully complete, the percentage varies between 39% and 61%.

As in Section 4.2.5, the missing gaps were processed using linear interpolation to be able to apply the MODWT-MRA as well as to fit the Lasso and LassoNet. In this case, both flow rate and pressure gaps were interpolated. After the interpolation, the MODWT-MRA could be applied to the pressure data accordingly. As seen in Chapter 3, the uncertainty caused by missing data is contained within a few levels of the MODWT-MRA (Figure 4.13).

Table 4.6 shows the definitions for models AF to DF for the scenario with missing flow rate and pressure. As in Section 4.2.5, models AF and CF use weighted Lasso via the transformation defined in Equation 4.2. The LassoNet models BF and DF also incorporate the weight vector w to make all models comparable. No data transformation is used for models BF and DF. As in Section 4.2.5, the weight vector w is defined as $w_i = 1$ when flow rate data is available and $w_i = 0$ otherwise.

Figure 4.14 shows the results for model AF. Evidently, the Lasso was unable to create an accurate

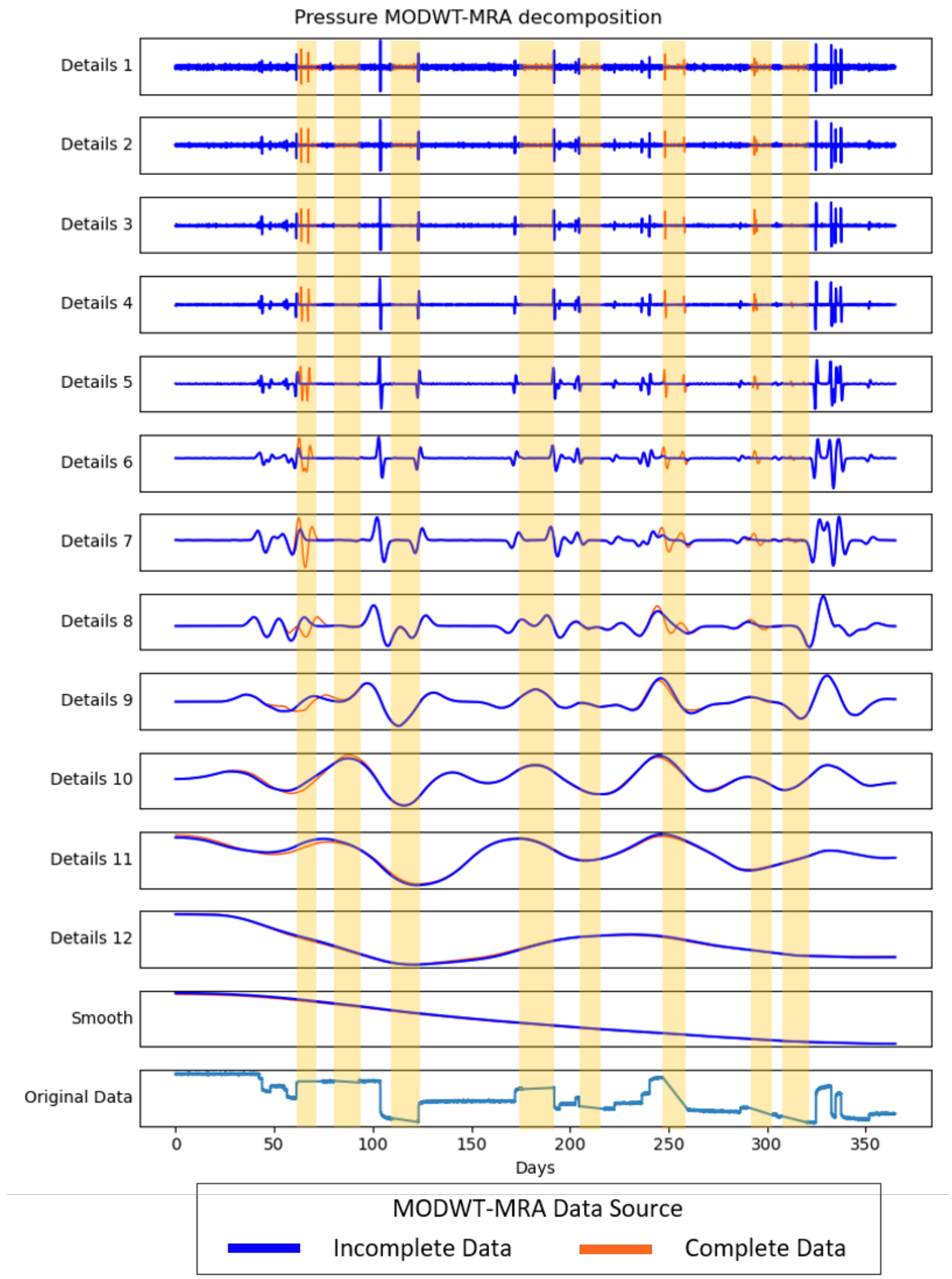

Figure 4.13: Pressure MODWT-MRA with missing data. The effect of missing data is mostly contained in the high frequency levels of the MODWT-MRA.

Table 4.5: Missing data fractions for each of the synthetic data sets.

Dataset	Missing Flow Rate	Missing Pressure	Full Data	No data (any variable)
d1	0.32	0.32	0.48	0.11
d2	0.17	0.27	0.61	0.06
d3	0.36	0.25	0.52	0.12
d4	0.34	0.22	0.53	0.09
d5	0.30	0.38	0.41	0.09
d6	0.29	0.27	0.45	0.01
d7	0.23	0.33	0.47	0.03
d8	0.38	0.35	0.39	0.12
d9	0.29	0.30	0.46	0.05
d10	0.29	0.30	0.48	0.07
d11	0.36	0.28	0.45	0.08
d12	0.28	0.26	0.50	0.05
d13	0.27	0.31	0.47	0.06
d14	0.33	0.31	0.46	0.11
d15	0.34	0.34	0.42	0.10
d16	0.24	0.35	0.47	0.06
d17	0.24	0.26	0.57	0.07
d18	0.24	0.28	0.52	0.04
d19	0.29	0.29	0.50	0.08
d20	0.31	0.22	0.57	0.11

Table 4.6: Model descriptions for the single-well scenario with missing flow rate data

Model Name	Regression Method	Features	Target
AF	Lasso (weighted)	Pressure (interpolated)	Flow Rate (interpolated)
BF	LassoNet		
CF	Lasso (weighted)	Pressure (interpolated) + MODWT-MRA	
DF	LassoNet		

flow rate estimate and the model shows extremely high bias. Even with the use of a weighted vector, the Lasso cannot overcome the high data fragmentation problem. A similar result was obtained with model BF (Figure 4.15). The model has a very high bias and is unable to get an accurate flow rate estimate, only reflecting values close to the average flow rate.

A different result was obtained when the MODWT-MRA pressure features are introduced. Model CF, the Lasso with pressure MODWT-MRA features achieved a much more unbiased estimate and it is able to recover the flow rate in the regions where pressure is available. Similar results are shown in Figure 4.16 for model DF, where the use of MODWT-MRA pressure features and the LassoNet clearly captured the flow rate when data is available and an unbiased model can be obtained even in a highly data fragmentation environment.

Throughout the 20 datasets, similar results were obtained for all models AF to DF. Figure 4.18 shows the performance distributions for all datasets in the single well scenario with missing flow rate and pressure data. As seen in the examples above, the models that do not incorporate the MODWT-MRA features are unable to estimate flow rate. Both model AF and BF have significantly larger train and test errors compared to models CF and DF, which incorporate the MODWT-MRA features. Overall, model CF showed the best performance both in train and test scenarios. These are similar results to those in Section 4.2.5. It is clear that using the MODWT-MRA features highly simplifies the learning problem and allows for the use of highly fragmented data in a way that a simple model such as the Lasso can estimate accurately the flow rate history from imperfect pressure data.

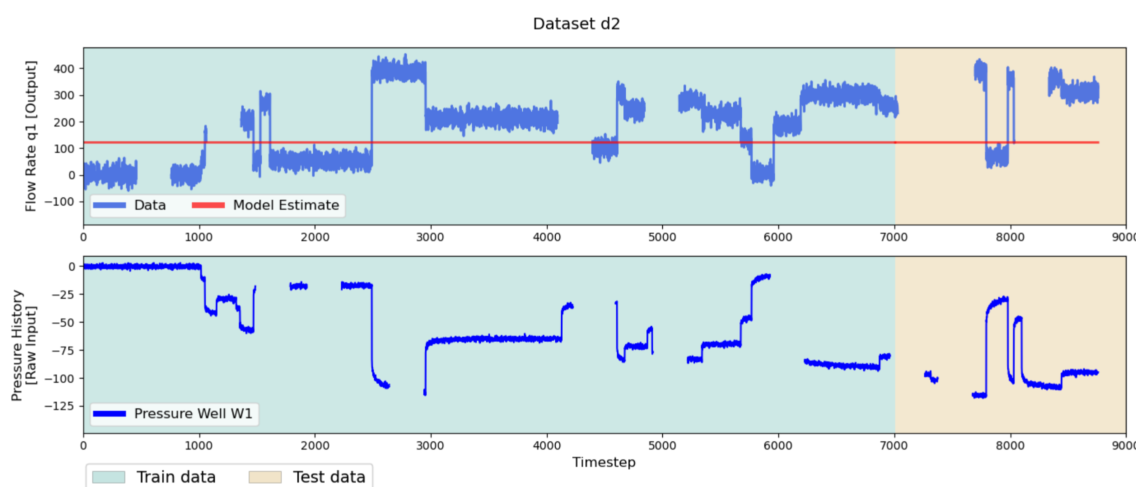

Figure 4.14: Model AF: Lasso regression with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

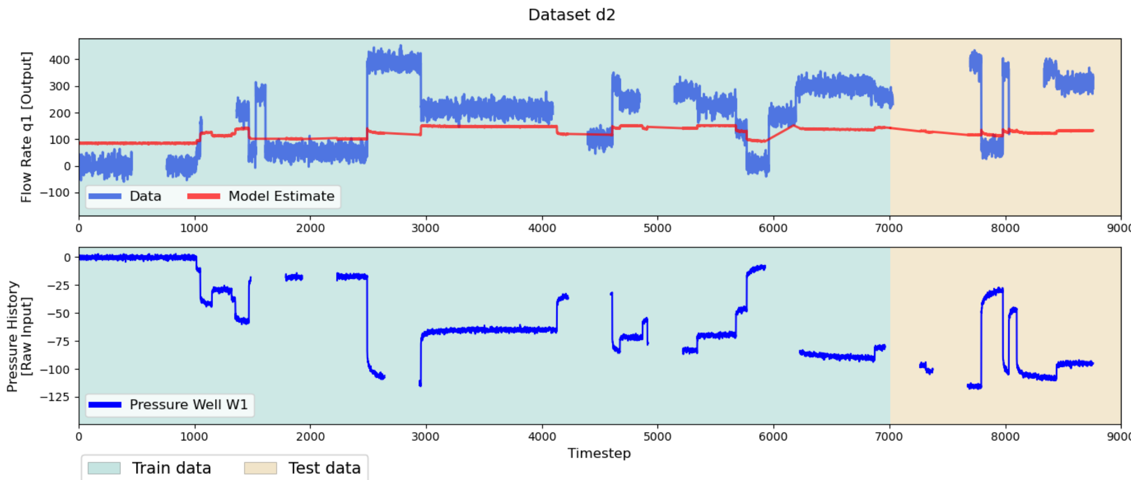

Figure 4.15: Model BF: LassoNet with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

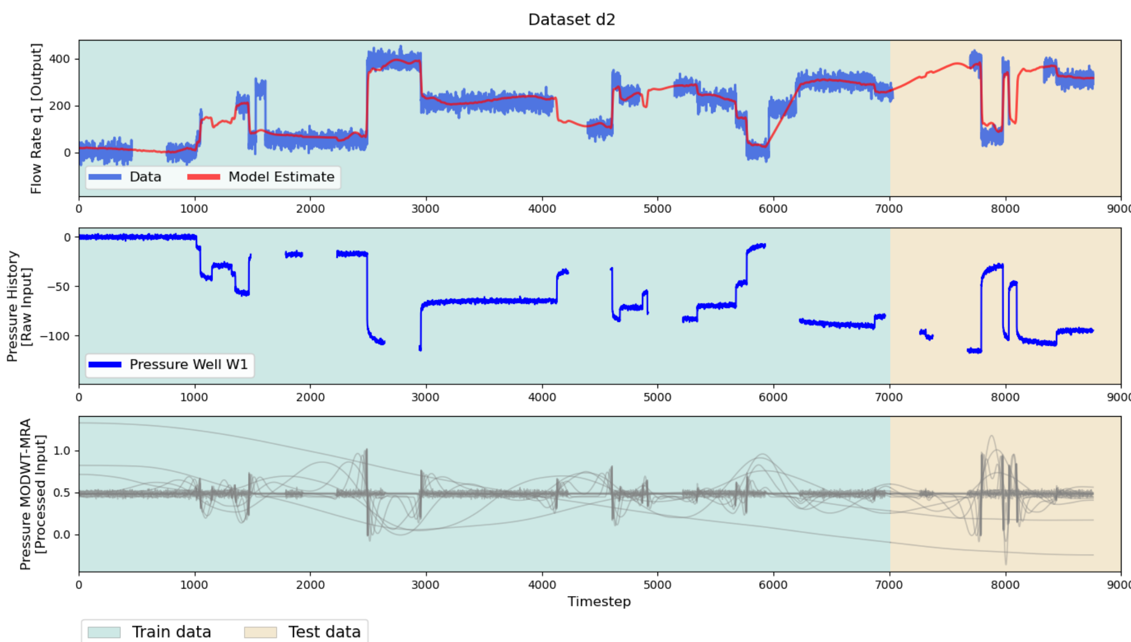

Figure 4.16: Model CF: Lasso regression with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

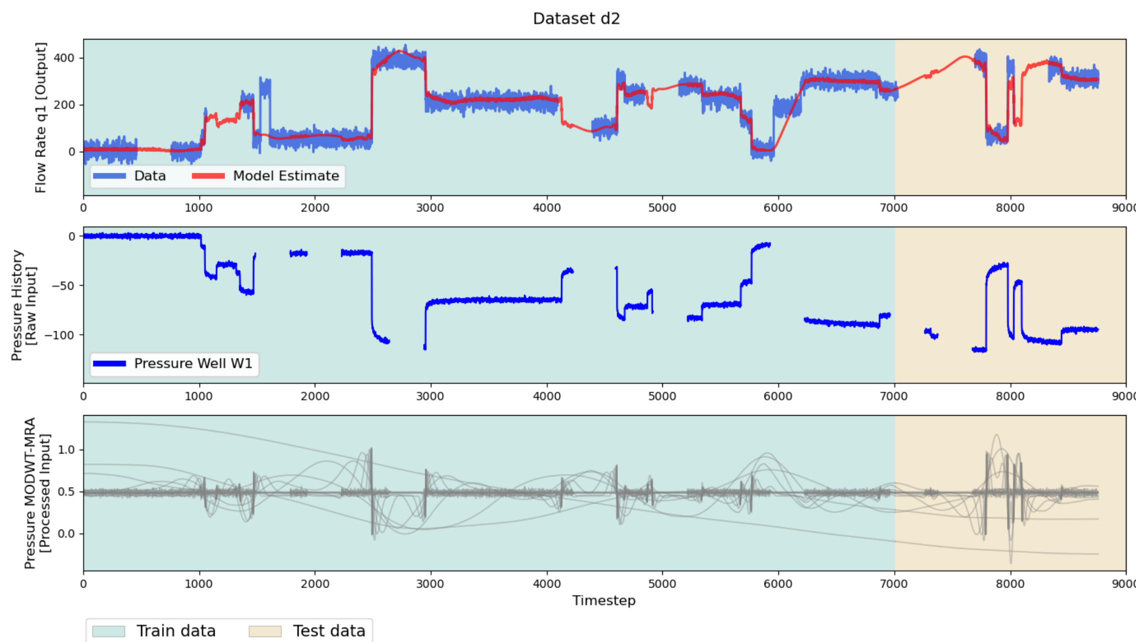

Figure 4.17: Model DF: LassoNet with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

4.3 Two Well Scenario

The two well scenario for flow rate recovery is an extension of the single well scenario, where pressure data from two connected and interacting wells are used for estimating the flow rate of one of the wells, called the target well. The increased complexity of the two well scenario for a data-driven model are the interference effects present in the pressure data of the target well. Using only pressure data from the target well is not sufficient to recover the flow rate history so the learning challenge from a data-driven perspective is to simultaneously separate the interference effects between both wells in the pressure data and create a function map from pressure to flow rate for the target well.

Formally, the two-well scenario is composed by a pair of wells called W1 and W2. Well W1, is the target well, whose flow rate response will be estimated. Well W2 is the interfering well. Both wells can produce at the same time with different flow rates and are connected hydraulically through

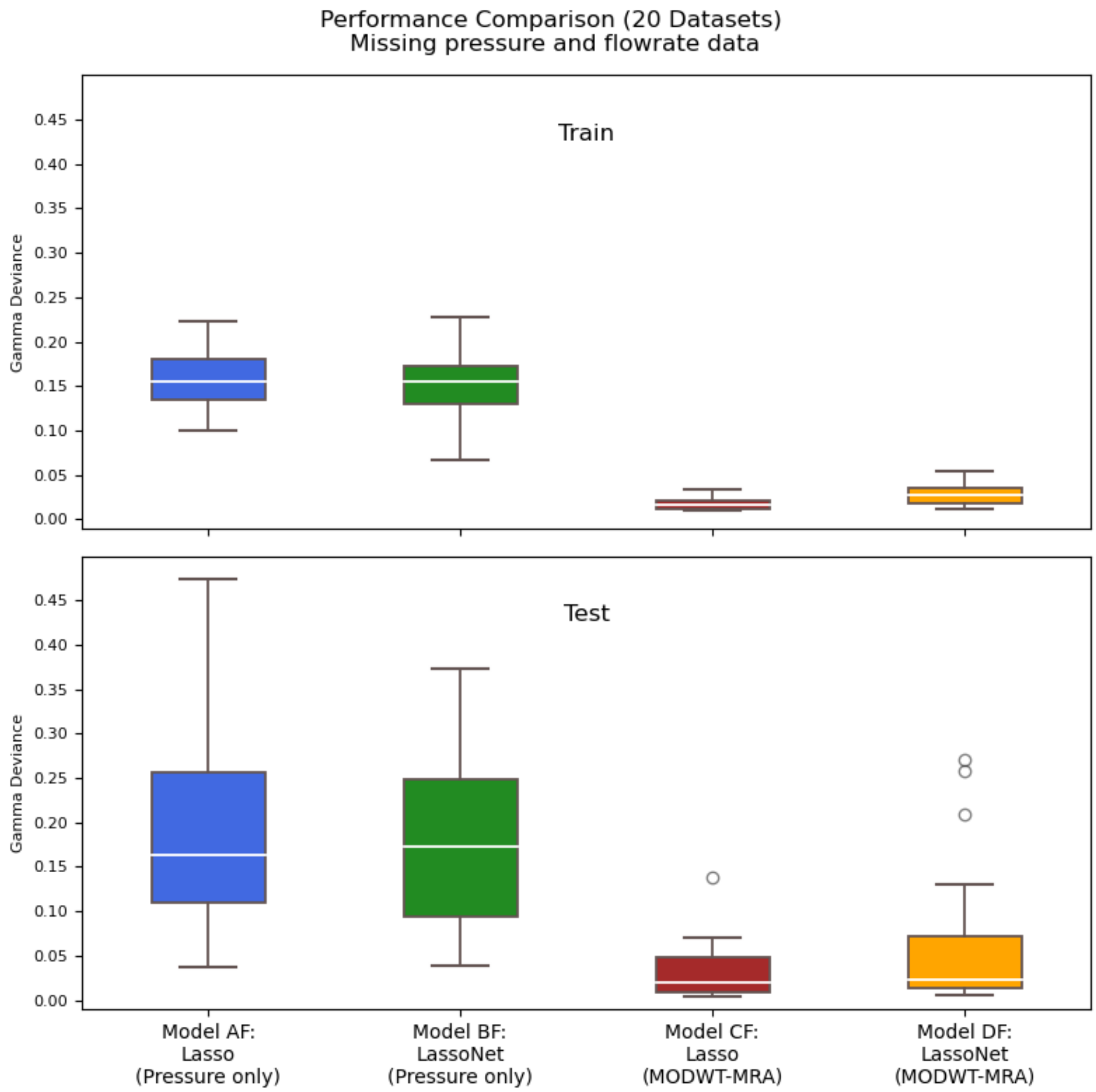

Figure 4.18: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

the reservoir, although that might be an unknown fact for the modeler. The objective is to obtain a data-driven model that estimates the flow rate of well W1 using as input the pressure histories of wells W1 and W2.

The scenario assumes that at least the pressure rate schedules of the two wells are known and the flow rate history of at least one of the wells is available. Similarly to the single well scenario, the methodology was tested in synthetic data to be able to identify its potential and measure its performance when the true answer is known.

4.3.1 Data Generation

The data for the two well scenario were generated using a modified version of the data generating function for the single-well scenario. The reservoir parameters were kept constant as shown in Table 4.7. For each well, a flow rate schedule was created using a randomized step function of up to 35 step changes and a maximum rate value of 400 STB/day. The sampling frequency for each well is one observation per hour and the total duration of the data is 365 days. The pressure solutions were obtained analytically in Laplace space, including superposition of the two wells and then subjected to numerical inversion. A total of 20 datasets consisting of two wells were created, each with different flow rate schedules. The same reservoir model and parameters were used as in the single well scenario and Gaussian noise of 5% was added to flow rate data and 1% noise to pressure data.

4.3.2 Model Specifications

As in the single well scenario, four model specifications (AF, BF, CF and DF) were fitted for the 20 synthetic datasets d1 to d20. Models AF and CF are based on the Lasso and models BF and DF use LassoNet for the fitting. There is a baseline model for both the Lasso and LassoNet which uses as features the unprocessed flow rate history of both wells W1 and W2. Models BF and DF use the pressure MODWT-MRA of both wells W1 and W2 as features. In models involving the

Table 4.7: Reservoir parameters for the synthetic data sets

Parameter	**Value**
k	100
S	0
C (STB/psi)	0.001
μ (cp)	1
h (feet)	50
ϕ	0.2
c_t (/psi)	5e-6
r_w (feet)	0.208
r_e (feet)	10,000
B	1

MODWT-MRA the depth parameter J_0 was set to 12, resulting in 13 features per well or a total of 26 features. Table 4.8 shows the details of all four models. In all cases, the train and validation sets used 80% of the data with the last 20% reserved for the test set.

Table 4.8: Model descriptions for the single-well scenario with missing flow rate data

Model Name	**Regression Method**	**Features**	**Target**
AF	Lasso	Pressure W1 & W2	Flow Rate W1
BF	LassoNet	(unprocessed)	
CF	Lasso	Pressure MODWT-MRA	
DF	LassoNet	W1 & W2	

LassoNet Specifications

As in the single well scenario, models BF and DF which use the LassoNet for fitting the data share training hyperparameters and hidden layer architecture. For both models, a hidden layer of size 1x100 was defined for the fully connected part of the LassoNet. The regularization hyperparameter λ was selected through cross validation for each dataset, however the initial λ and regularization path (search space) is shared among the two models. The search space for λ was defined as a geometric progression spanning from $\lambda = 400$ to $\lambda = 6,600$.

Both models BF and DF also share the hierarchy hyperparameter M that controls the relative importance of the linear and nonlinear components in the LassoNet. A value of $M = 0$ results in a Lasso regression while $M \to +\infty$ results in an unregularized fully connected neural network. For both models BF and DF the value was kept constant as $M = 200$. The experience gained using the LassoNet revealed similar results at values of $200 < M < 300$ with the larger parameter sensitivity being λ. For this reason and to save computational time, M was not part of the cross-validation parameter optimization. The optimizer used for the neural network training was Adam with a learning rate of 0.01. An early stopping strategy was used for training, halting the training when a least 10 training epochs had passed without a minimum improvement of 1% in the validation set.

4.3.3 Performance Results

Figures 4.19 and 4.20 show the results for the baseline models AF and BF. Both the Lasso and LassoNet models were able to capture the flow rate response for the target well W1 from the unprocessed pressure history of the two wells. Models CF and DF, which incorporate the MODWT-MRA features also successfully capture the flow rate history of the target well with a less noisy signal than the original data. To further explore the differences between the four models, the gamma deviance distribution plots are shown in Figure 4.23. Overall, the four models have very similar median values for the test set even if in training model DF appears more accurate. Both models BF and DF which use LassoNet for fitting are slightly more accurate than the Lasso based models regardless of the use of MODWT-MRA features. Model BF has the best performance overall.

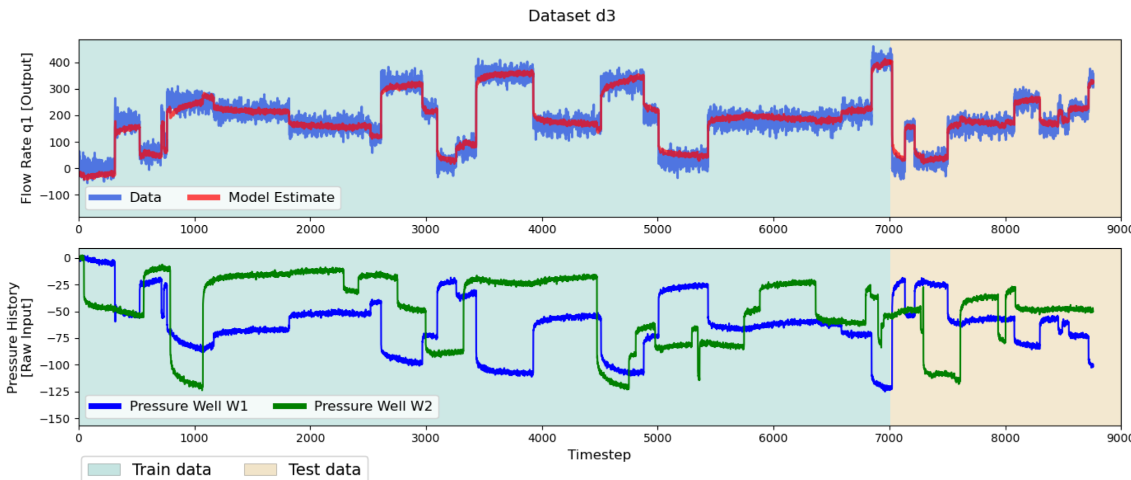

Figure 4.19: Model AF: Lasso regression with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

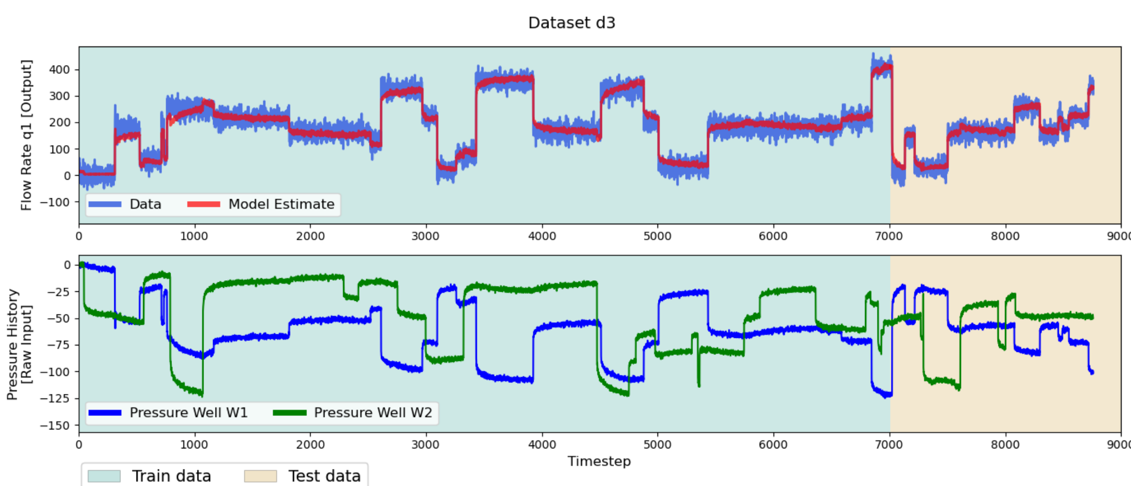

Figure 4.20: Model BF: LassoNet with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

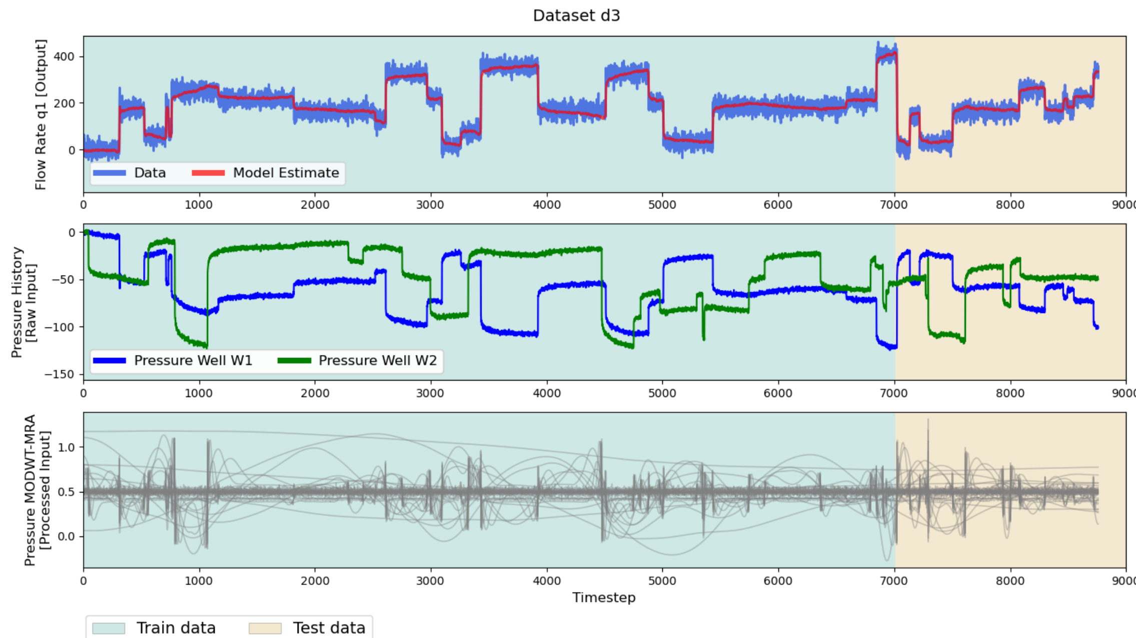

Figure 4.21: Model CF: Lasso regression with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

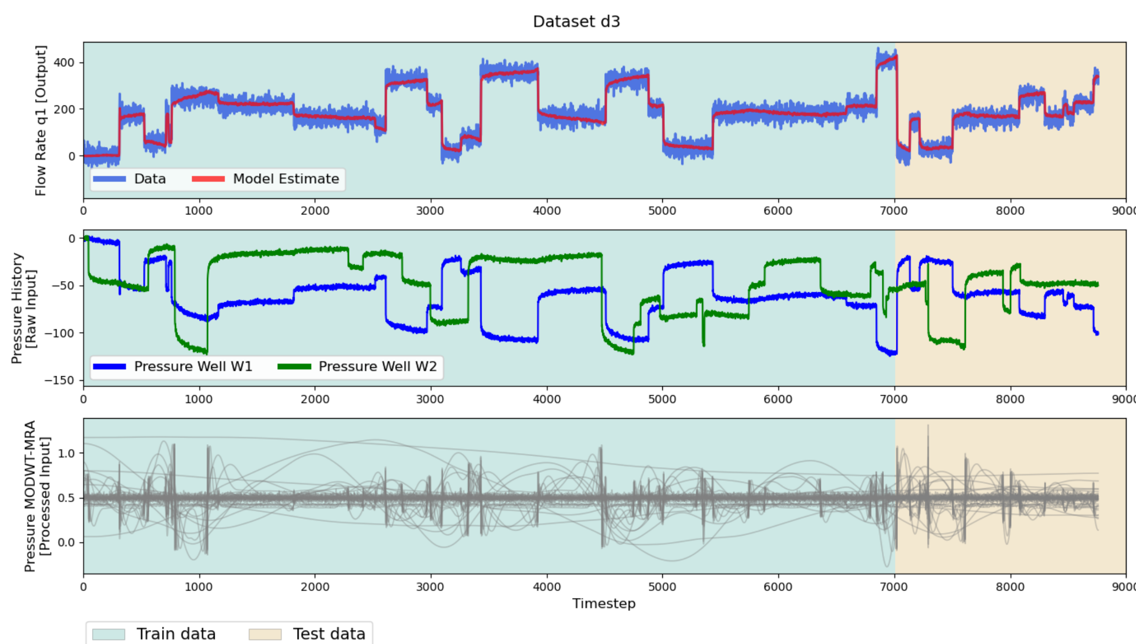

Figure 4.22: Model DF: LassoNet with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

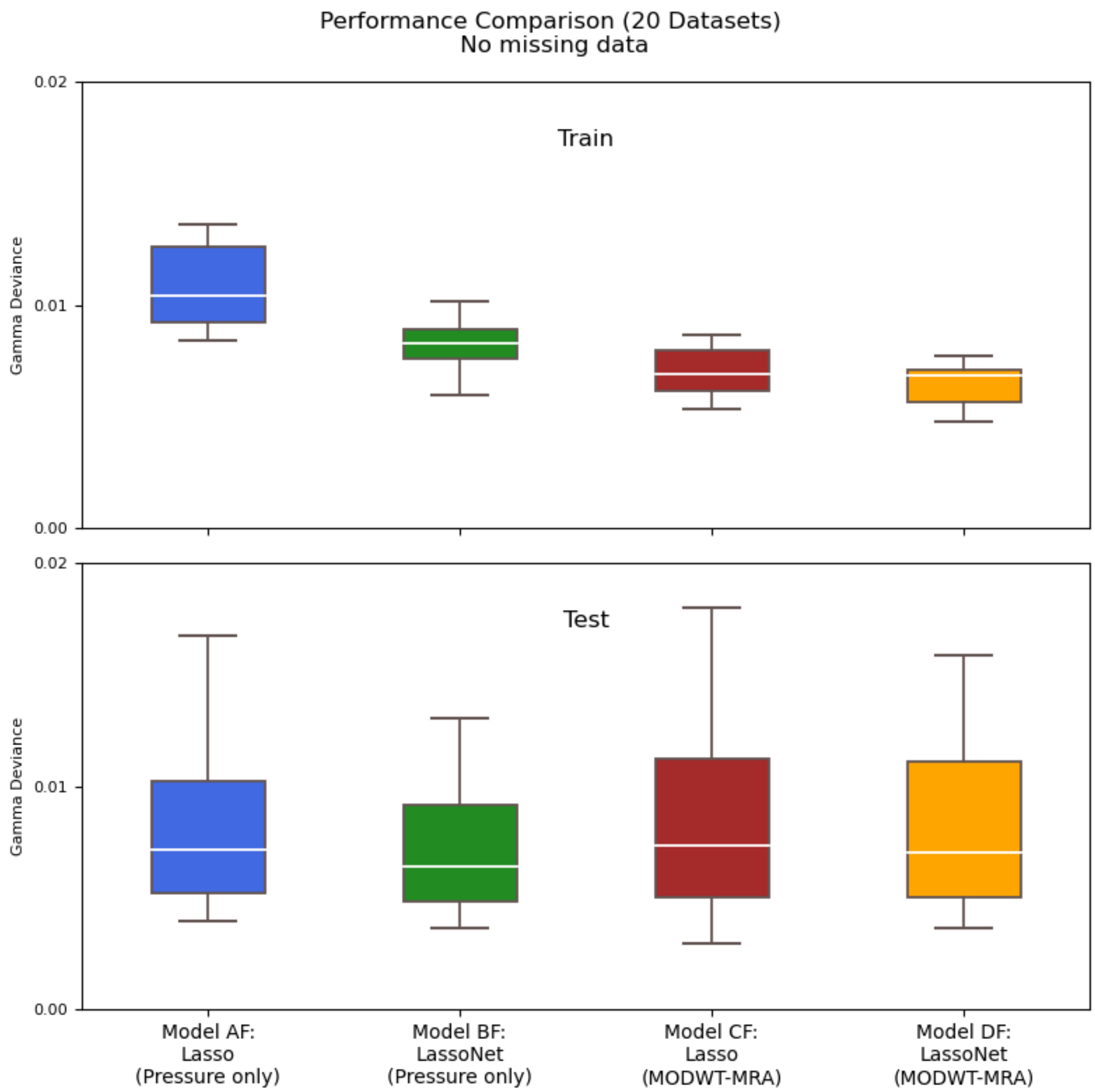

Figure 4.23: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

4.3.4 Missing Flow Rate Data

In this scenario both wells W1 and W2 are flowing and the goal is to build a model for the flow rate of well W1 using the pressure histories of both wells. The flow rate data is incomplete with gaps in the flow rate history of the target well W1. The 20 datasets used in this scenario are based in those of Section 4.3.1 with the added difference of missing gaps in flow rate. A total of ten missing gaps were introduced to the flow rate of well W1 at randomized points in time throughout each time series with each gap being 300 hours long. No modifications were done for pressure data of either wells W1 or W2. Table 4.9 shows the missing data statistics for all 20 datasets used in this scenario.

Four models were built using the Lasso and LassoNet methodologies. Both methodologies were tested with the unprocessed pressure data as input as well as the MODWT-MRA decomposition of pressure. Table 4.10 shows the definitions of all four models AF to DF. In an analogous way to the single well scenario, the missing flow rate data was interpolated linearly before the modeling. For the MODWT-MRA features, the interpolation was applied before applying the decomposition.

As described in Section 4.2.5, the linear interpolation in flow rate data introduces a large bias for the training. To solve this issue in models AF and CF, a weighted Lasso was used. The weights allowed only the complete flow rate periods to be used for the fitting process. Specifically, the following transformation was applied to the data for model AF:

$$
\begin{aligned}
y_{trans} &= \sqrt{w} * y \\
x_{trans} &= \sqrt{w} * x
\end{aligned}
\tag{4.3}
$$

where w is the weight vector, and $w_i = 1$ when flow rate data is available and $w_i = 0$ otherwise. Without using the data transformation, models AF and CF result in a constant flow rate estimate through the entire time period. For LassoNet models AB and AC the weight vector w was passed directly to the loss function without the need for any data transform.

Figures 4.25 to 4.27 show the results of all four models for one of the 20 datasets. Models AF and BF, which use the unprocessed pressure inputs display accurate and unbiased results in both training

Table 4.9: Fraction of total missing flow rate data in well W1 for each of the synthetic data sets.

Dataset	**Missing Flow Rate %**
d1	0.25
d2	0.30
d3	0.25
d4	0.23
d5	0.16
d6	0.21
d7	0.34
d8	0.29
d9	0.29
d10	0.34
d11	0.33
d12	0.35
d13	0.30
d14	0.29
d15	0.20
d16	0.30
d17	0.30
d18	0.32
d19	0.29
d20	0.35

Table 4.10: Model descriptions for the two-well scenario with missing flow rate data

Model Name	**Regression Method**	**Features**	**Target**
AF	Lasso (Weighted)	Pressure W1 and W2 (Variable Transform)	Flow Rate (Interpolated + Variable Transform)
BF	LassoNet	Pressure W1 and W2	Flow Rate (Interpolated)
CF	Lasso (Weighted)	Pressure MODWT-MRA W1 and W2 (Variable Transform)	Flow Rate (Interpolated + Variable Transform)
DF	LassoNet	Pressure MODWT-MRA W1 and W2	Flow Rate (Interpolated)

and test sets. As the noise level of pressure data is less than that of flow rate, the estimates are less noisy than the true data, even for the linear Lasso model AF shown in Figure 4.24. Both models AF and BF successfully recover the missing data gaps. Models CF and DF, which use the pressure MODWT-MRA features show an unbiased and relatively accurate flow rate estimate (Figures 4.26, 4.27). However the flow rate changes are less sharp than the true data, a problem less present in the models that do not use the MODWT-MRA features.

The results for all 20 datasets reveal a pattern where models AF and BF outperform significantly the rest of the models (Figure 4.28). Both models AF and BF use the unprocessed pressure data as input. These results suggest that using the MODWT-MRA decomposition for modeling flow rate does not result in better accuracy as it does when modeling pressure. In this scenario, the results point to model AF, the simplest of all four models, as the one with the lowest median gamma deviance and the tightest distribution across datasets. Compared to the scenario with no missing data, all models suffered from a decrease in performance throughout the 20 datasets. However models CF and DF suffered the most, with a threefold increase in median gamma deviance for the test set. In comparison, models AF and DF only had an increase of 7% and 29% increase respectively in median gamma deviance for the test set.

4.3.5 Missing Flow Rate and Pressure Data

The last case for the two wells scenario is that of missing data in both pressure and flow rate. As explained in Chapter 3, having missing data in both inputs and training outputs heavily increases data fragmentation due to misaligned data gaps. For this case, the missing data gaps were introduced at random points in time in both pressure data for wells W1 and W2 and flow rate data of well W1. The missing data gaps had a 300 hour duration and the process was done independently for each well and variable. This creates time series where the gaps are not aligned in time, increasing

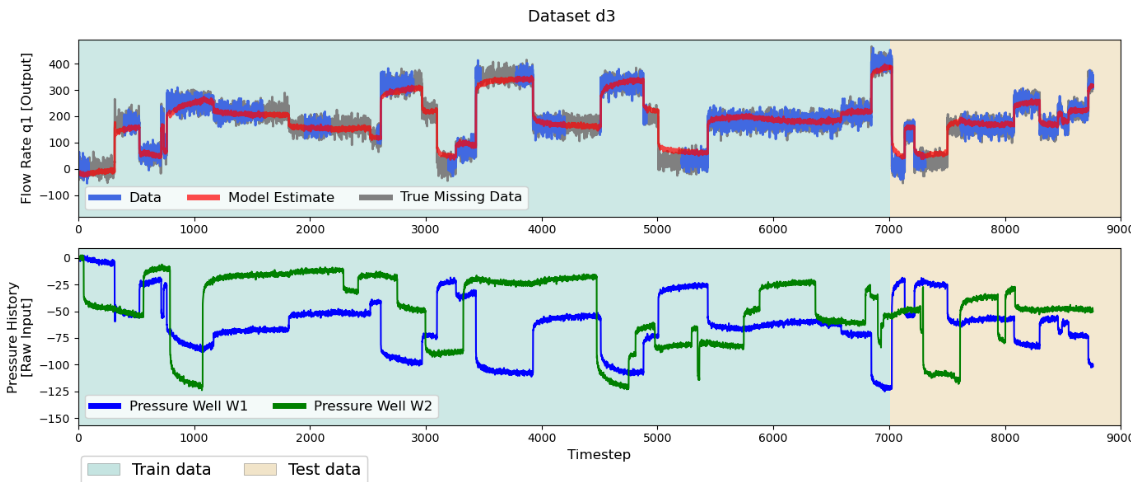

Figure 4.24: Model AF: Lasso regression with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

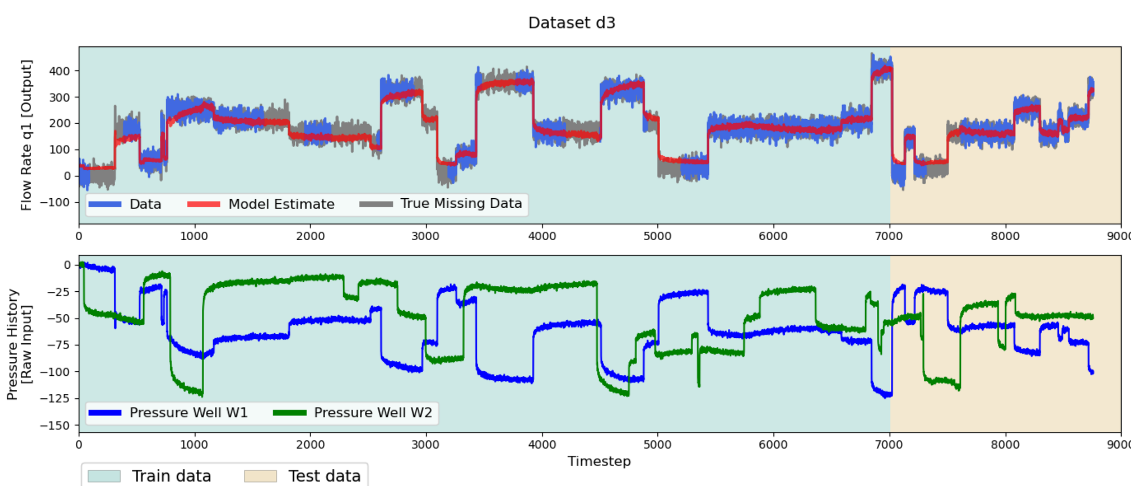

Figure 4.25: Model BF: LassoNet with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

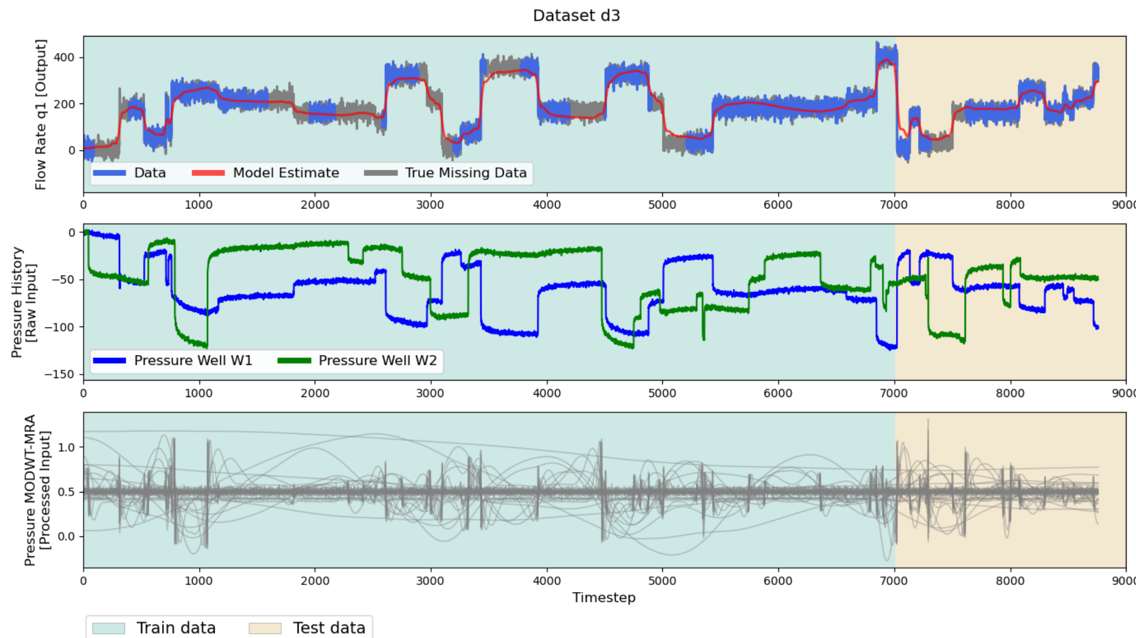

Figure 4.26: Model CF: Lasso regression with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

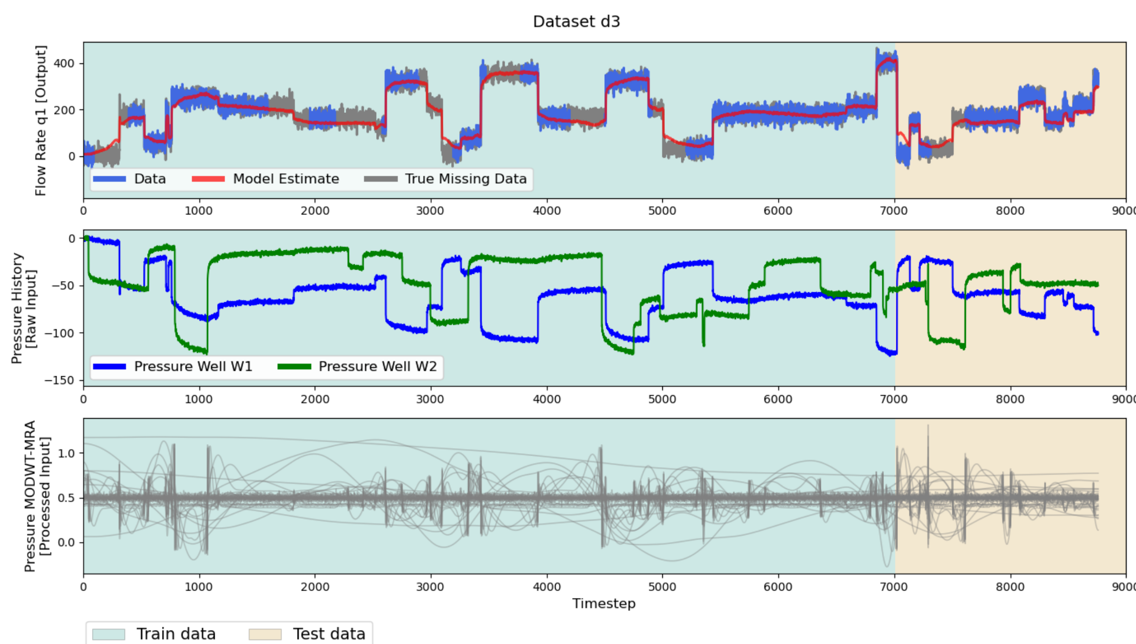

Figure 4.27: Model DF: LassoNet with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

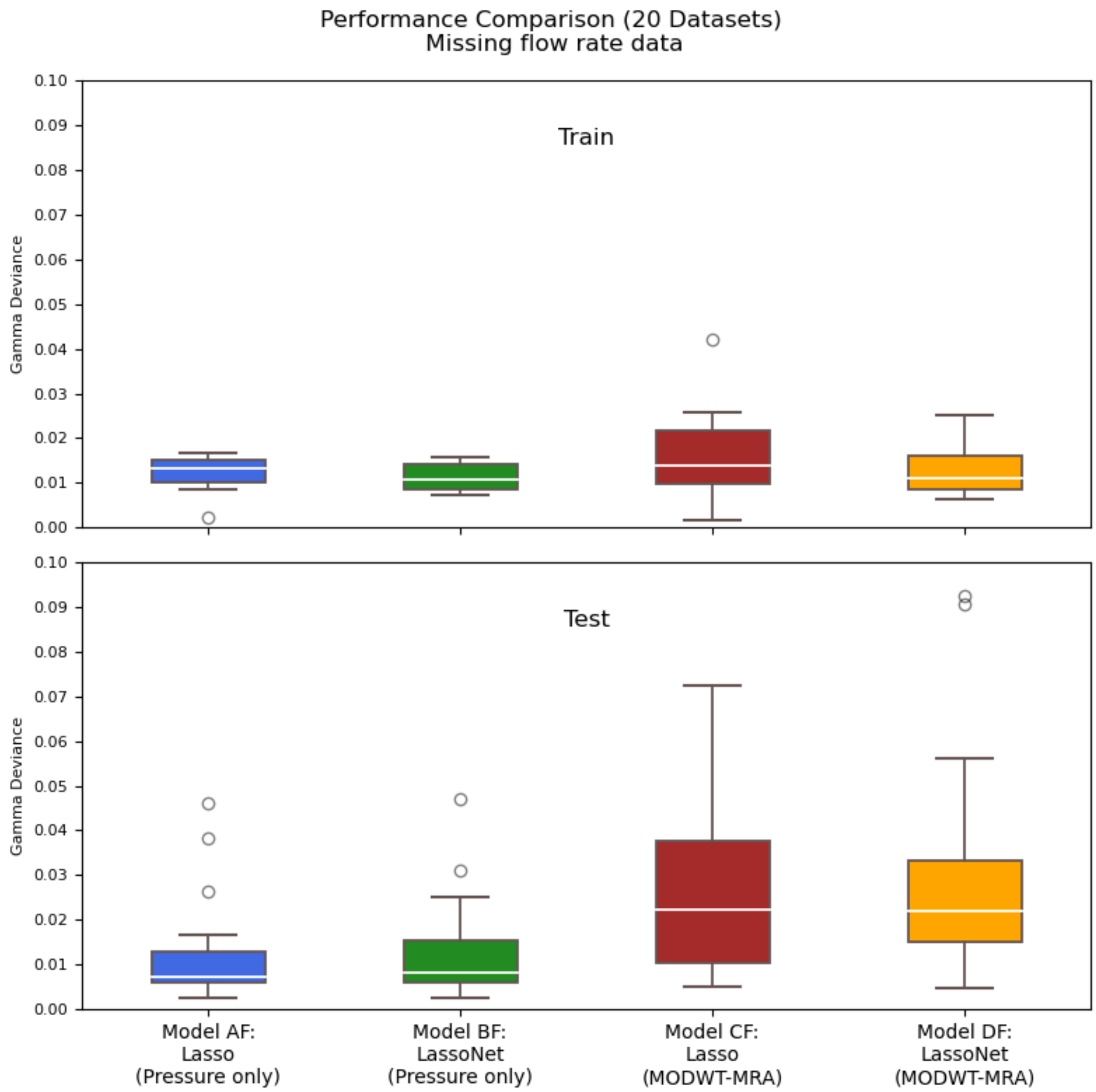

Figure 4.28: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

the data fragmentation even more. Table 4.11 shows the missing data statistics for each of the 20 datasets used in this scenario.

Table 4.11: Fraction of total missing flow rate and pressure data for each of the synthetic data sets.

Dataset	**Missing Pressure W1**	**Missing Pressure W2**	**Missing Flow Rate W1**	**Full Data**
d1	0.25	0.34	0.26	0.24
d2	0.25	0.34	0.28	0.38
d3	0.16	0.35	0.25	0.38
d4	0.34	0.29	0.28	0.39
d5	0.29	0.37	0.31	0.32
d6	0.33	0.27	0.25	0.35
d7	0.30	0.36	0.26	0.33
d8	0.20	0.26	0.25	0.34
d9	0.30	0.37	0.27	0.40
d10	0.29	0.17	0.30	0.35
d11	0.28	0.21	0.28	0.32
d12	0.34	0.18	0.30	0.25
d13	0.35	0.35	0.29	0.34
d14	0.23	0.23	0.37	0.31
d15	0.34	0.34	0.33	0.38
d16	0.22	0.22	0.33	0.38
d17	0.28	0.28	0.34	0.34
d18	0.31	0.35	0.37	0.23
d19	0.20	0.33	0.35	0.32
d20	0.37	0.29	0.27	0.34

The same combination of four models AF to DF was tested, with two Lasso based models and two LassoNet based models. Both pressure and the pressure MODWT-MRA features were tested as well. Table 4.12 shows the modeling methodologies and input definitions of all four models. In a similar way to Section 4.3.4, for models AF and CF, which use the Lasso, a data transformation was done to accomplish a weighted fitting to handle the bias introduced by the interpolation. The weight transform was defined as:

$$y_{trans} = \sqrt{w} * y$$
$$x_{trans} = \sqrt{w} * x \tag{4.4}$$

where w is the weight vector, and $w_i = 1$ when flow rate data is available and $w_i = 0$ otherwise. It must be noted that the weights are not related with the pressure input but only to the flow rate output.

Table 4.12: Model descriptions for the two-well scenario with missing flow rate data

Model Name	**Regression Method**	**Features**	**Target**
AF	Lasso (Weighted)	Pressure W1 and W2 (Interpolated + Variable Transform)	Flow Rate (Interpolated + Variable Transform)
BF	LassoNet	Pressure W1 and W2 (interpolated)	Flow Rate (Interpolated)
CF	Lasso (Weighted)	Pressure MODWT-MRA W1 and W2 (Interpolated + Variable Transform)	Flow Rate (Interpolated + Variable Transform)
DF	LassoNet	Pressure MODWT-MRA W1 and W2 (Interpolated)	Flow Rate (Interpolated)

Figures 4.29 to 4.30 show the results of all models for a single dataset. Similar to the results obtained in Section 4.3.4, the two models that use the pressure data without the MODWT-MRA have a good performance with less noise than the original data. Both models AF and BF display sharp changes in flow rate, even if a truly constant flow rate estimate is not achieved between step changes. In contrast, models CF and DF which use the MODWT-MRA pressure decomposition show less sharp flow rate changes. Model DF also displays increased noise around the flow rate changes.

The results for the 20 datasets follow the same described pattern. Figure 4.33 shows the gamma deviance for all the 20 datasets. Similar to the scenario with only missing flow rate data, all models suffer from a decrease in performance when compared to the scenario with no missing data. Model AF, has the best overall performance with model BF showing similar results. Both models that include the MODWT-MRA decomposition as inputs have worse performance than the baseline models.

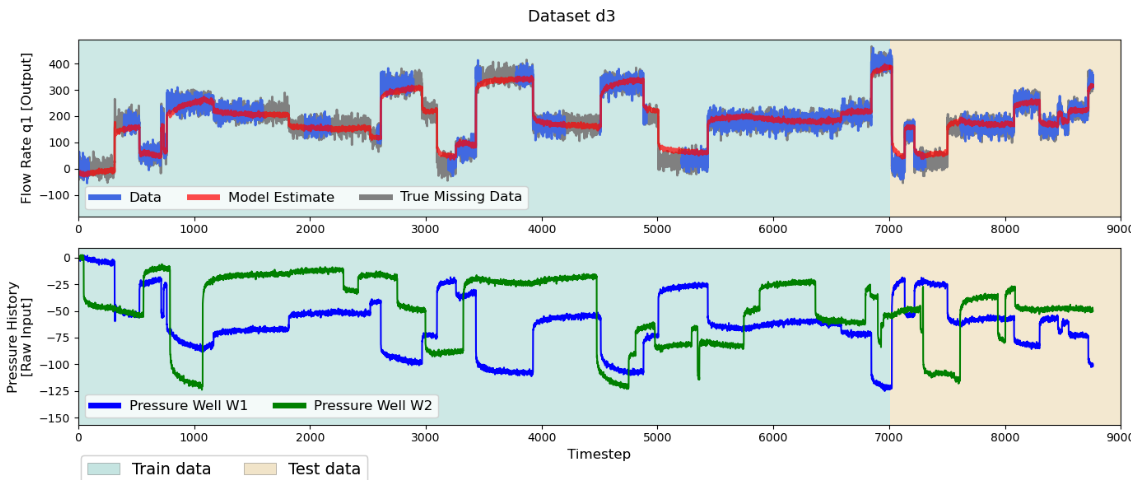

Figure 4.29: Model AF: Lasso regression with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

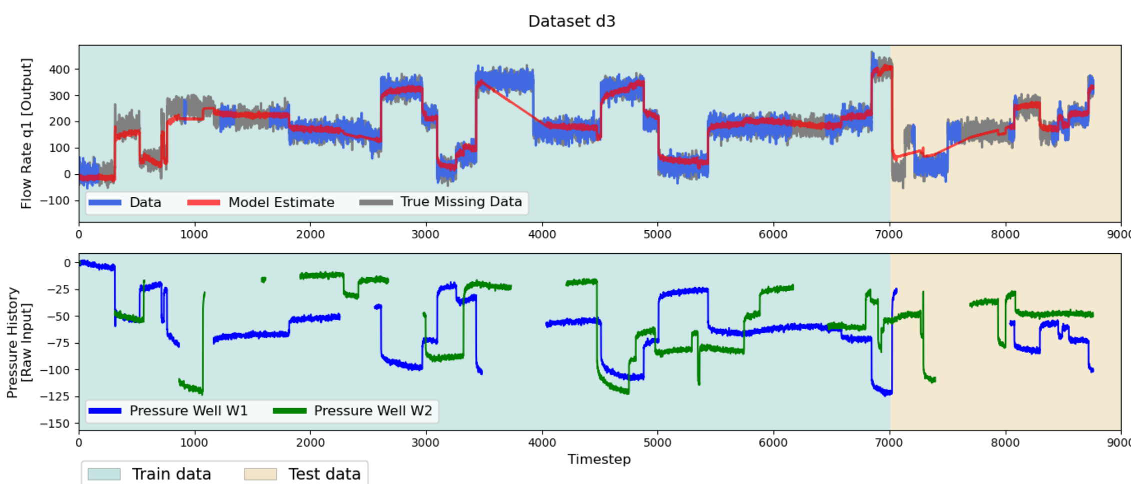

Figure 4.30: Model BF: LassoNet with unprocessed pressure as input. Top panel: Flow rate estimates. Bottom panel: Unprocessed pressure history.

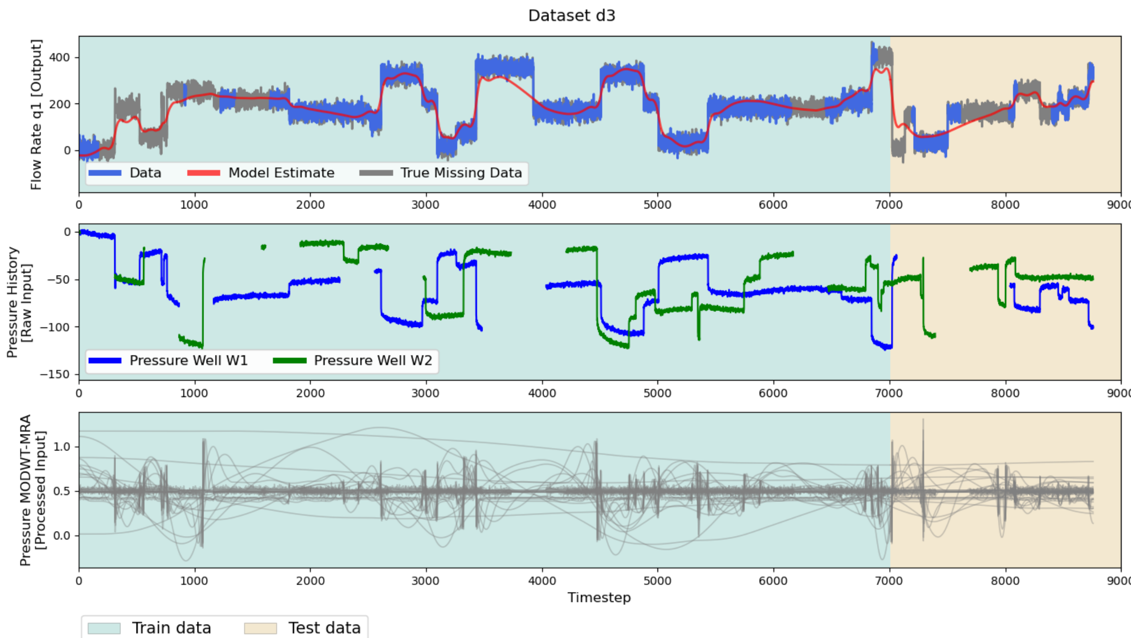

Figure 4.31: Model CF: Lasso regression with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

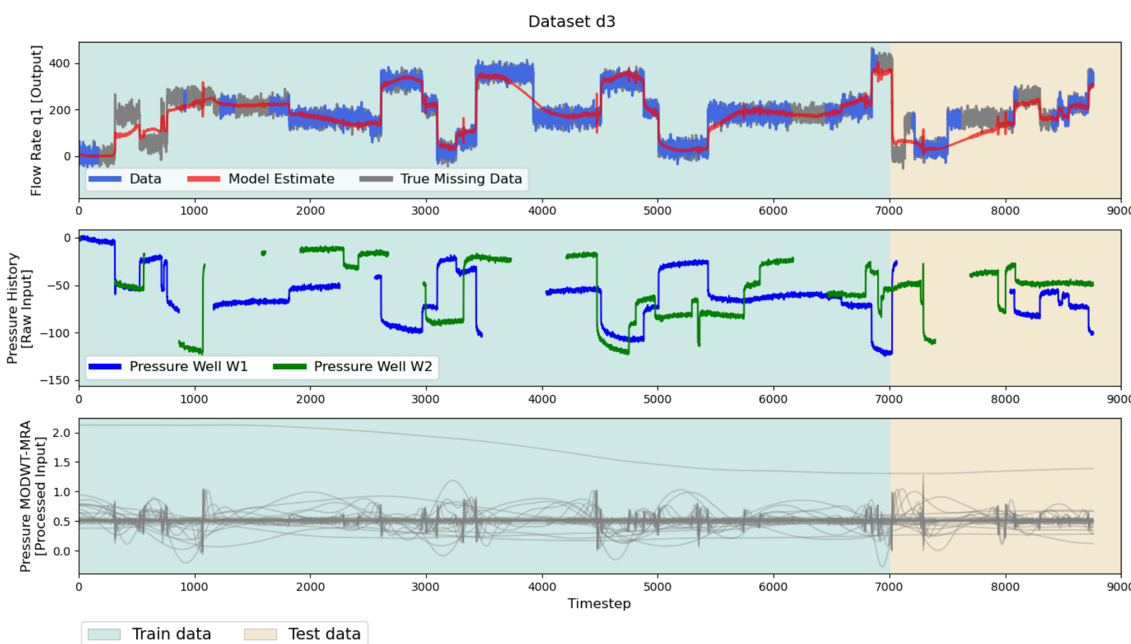

Figure 4.32: Model DF: LassoNet with pressure MODWT-MRA as input. Top panel: Flow rate estimates. Center panel: Unprocessed pressure history. Bottom panel: Pressure MODWT-MRA.

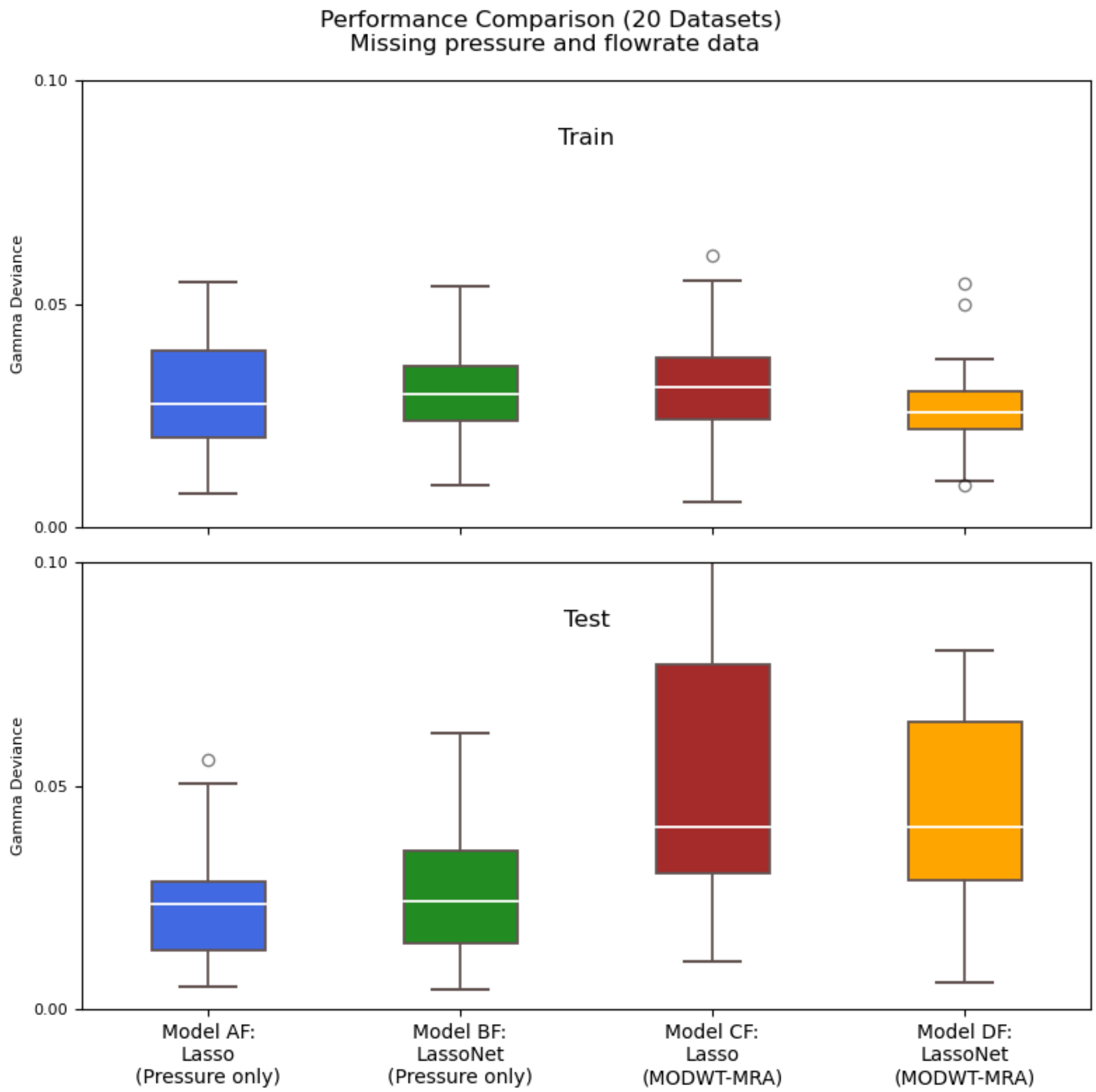

Figure 4.33: Single well scenario models comparison. The top panel corresponds to training data and bottom panel shows the test data. A lower gamma deviance denotes better performance

4.4 Summary

In this chapter, a series of machine learning and deep learning models for modeling flow rate were introduced. Two general scenarios were explored including single well and double well with increasing amounts of missing data. The results obtained show that using pressure to recover flow rate is possible even when large amounts of data are missing. However, the most successful models required using weighted regression to successfully deal with missing data. Also, the results of this chapter showed that applying the MODWT-MRA to pressure data used as model input was not as beneficial when modeling flow rate as it was when modeling pressure as in Chapter 3.

The models that used the MODWT-MRA decomposition tended to produce an excessively smooth flow rate estimate. While the MODWT-MRA decomposition can preserve sharp edges in data, those edges end up being contained in the high frequency levels of the decomposition, which are also the levels where noise is contained. This makes it complicated for models that apply shrinkage, as the noise features is dropped thus decreasing the ability of the models to recreate sharp features. This has consequences for flow rate estimation compared to pressure estimation, because flow rate histories have sharp features whereas the physical relationship between flow rate and pressure causes the pressure response to be smoother.

Chapter 5

Conclusions

This work introduced a framework to use time series production data to create a full data-driven model of well response. The overall approach taken in this work was to allow for the use of imperfect data without the need of tailored data cleaning and selection processes. Two applications were studied for the designed methodology, modeling pressure from flow rate data and reconstructing flow rate history using pressure data. For each application, single well and two well scenarios were tested with increasing amounts of missing data.

The Maximum Overlap Discrete Wavelet Transform Multiresolution Analysis (MODWT-MRA) was introduced as it proved have useful properties for building data-driven models. Among those properties are its applicability to datasets of any length, lack of decimation and invariance to data shifting. Moreover, the research showed that by applying the MODWT-MRA to production data time series consisting of flow rate and pressure data, a decomposition into superposed virtual wells is achieved. For pressure, the additive decomposition follows the principle of superposition, with the original pressure signal recovered by adding all the MODWT-MRA components. When applied to flow rate, the MODWT-MRA is mass preserving. These proved to be useful properties when the data contains imperfections such as noise and missing data because the noise gets contained in the high frequency levels and the uncertainty caused by missing data is also contained to a few of the total decomposition levels.

The modeling approach in both applications involved using the MODWT-MRA to create automatic input features. By applying the decomposition, the goal was to split a complex dataset into a set of data inputs with simpler behavior. This required the use of modeling methodologies capable of shrinkage, this is automatically select useful variables and discard those that are not useful. Two shrinkage methods were used for the study: the Lasso and LassoNet. These methods were chosen because of their interpretability, ease of use and capacity for comparison as LassoNet is a nonlinear generalization of the Lasso based on neural networks.

The results for modeling pressure pointed to a substantial benefit of using the MODWT-MRA features. In most tests the produced estimates showed less noise and higher accuracy when compared to models that used the unprocessed data. These advantages proved more significant when missing data was present. In the two well scenario, creating a data-driven model for pressure allowed for detecting connectivity and filtering out interference. In this case, only by using the MODWT-MRA and a complex model like LassoNet could a true interference-free model be constructed from data with interference.

For the flow rate recovery application, the advantages of using the MODWT-MRA were less notable. In the single well scenario, similar results were obtained by using features with or without the decomposition. In the two well scenario, no improvement was achieved when using the MODWT-MRA decomposition.

5.1 Future Work

Additional applications of the designed methodology remain to be tested. The ability of the methodology to seamlessly deal with noise and missing data are two of the properties that might prove useful for other tasks. Some unexplored ideas are listed here:

1. Detecting interference and connectivity among more than two wells. Modeling pressure response when more than two wells are connected is an interesting and challenging scenario, especially if the degree of connectivity is unknown. The problem would be more difficult to model but it is possible that the wavelet decomposition presented here is useful to identify the signature of multiple wells interfering with each other.

2. Flow rate allocation from multiple wells. In this application, the total flow coming from multiple wells is known but the fractional contribution of each well is unknown. In this scenario, the total flow rate from all wells could be decomposed into a set of virtual wells each of which can then be assigned to one of the real wells by using a mixture model.

3. Temperature data incorporation. The effect of the MODWT-MRA decomposition to temperature data is unknown. In this work it was shown that mass preservation and superposition were honored for flow rate and pressure respectively. The additive nature of the decomposition opens the door for using the MODWT-MRA to use temperature time series to also identify connectivity between wells or increase accuracy of existing flow rate models.

Appendix A

LassoNet Tuning and Parameter Selection

A.1 LassoNet Implementation

As described in Section 3.3.2, LassoNet is an extension of the Lasso that combines the linear component and a neural network component to preserve the feature sparsity properties and interpretability of the Lasso and the flexible function representation of neural networks. The functional form of the LassoNet is [Lemhadri et al., 2021]:

$$f_{\theta,W}(x) = \theta^T x + g_W(x) \tag{A.1}$$

where g_W is a feed-forward neural network with weights W, θ denotes the residual layer parameters and x are the input features. The loss function for LassoNet is:

$$L(\theta, W) = \frac{1}{n}\sum_{i=1}^{n} l\Big(f_{\theta,W}(x_i),\ y_i\Big) \tag{A.2}$$

where n is the number of observations in x, and l denotes the appropriate loss function such as mean squared error. The objective function for LassoNet is defined as:

$$\begin{aligned} &\min_{\theta,W}\ L(\theta, W) + \lambda\ ||\theta||_1 \\ \text{s.t.}\quad &||W_j^{(1)}||_\infty\ \leq M\ |\theta_j|, \quad j = 1, ..., d \end{aligned} \tag{A.3}$$

where $W_j^{(1)}$ are the weights for feature j in the first hidden layer, λ is the regularization parameter which encourages sparsity similar to that of the Lasso.

The parameter M controls the strength of the linear and nonlinear components. When $M = 0$, only the residual layer is active and the formulation is that of the Lasso. Alternatively, when $M \to +\infty$, the unregularized neural network is obtained.

The LassoNet can be implemented in any generic deep-learning framework such as Pytorch or Tensorflow. The authors of LassoNet [Lemhadri et al., 2021] released a publicly available Python package used in this work. This package can be installed by typing `pip install lassonet` or downloaded from https://github.com/lasso-net/lassonet. This implementation is based on Pytorch and contains a set of interfaces for creating LassoNet models for regression and classification as well as utilities for fitting and validating a model. For the applications shown in this work, the base code of the `lassonet` library was customized to allow for the incorporation of a weighted loss function for model selection during training and cross validation.

A.1.1 Cross-Validation Stragegy

In this work, the LassoNet models were built by creating an instance of the class `LassoNetRegressor` through the `LassoNetRegressorCV` function, which implements a cross-validation strategy and returns the best chosen parameters as well as the regularization path for the model. In this work, the chosen cross-validation strategy was a time-series split, which is a specific kind of k-fold cross-validation that maintains the temporal structure of the data. To achieve this, successive training sets are supersets of those that come before them thus preventing a model from learning from future data (Figure A.1)

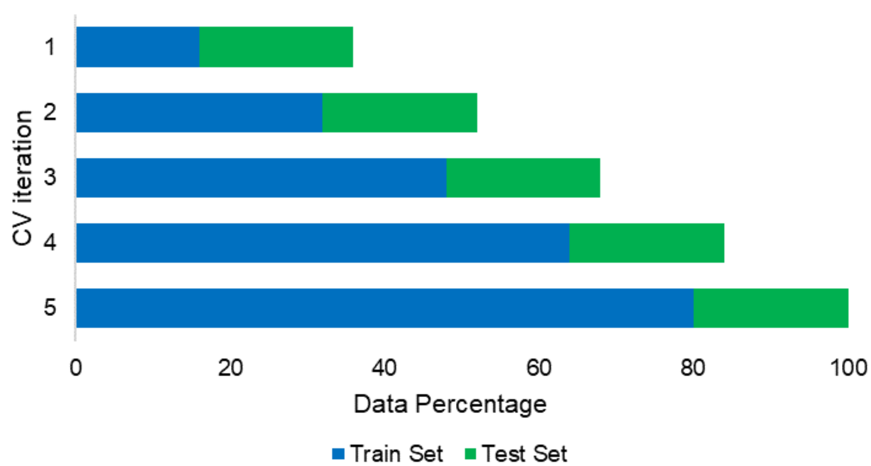

Figure A.1: Time series cross-validation split

For most of the examples shown in this work, 5 cross-validation folds were used. In some instances with missing data, the number of folds was reduced to 3 because the smallest training set would not contain useful data otherwise. However, using less cross-validation folds contributed to instability in the estimate of optimal λ. A minimum of 5 folds is recommended unless missing data makes it impossible to start the training.

A.1.2 Regularization parameters

LassoNet allows for multiple ways of regularization. The first and most important is λ, the Lasso-like regularization parameter that reduces model complexity by shrinking the input coefficients towards zero. In addition to λ, the neural network component of LassoNet can also be regularized through dropout or adding L2 penalties to the skip connections. Dropout is the process of skipping training passes in a random subset of the elements of a layer. This achieves sparsity without an explicit requirement for it. The L2 penalty on the skip connection works in a similar way to the λ regularization parameter by increasing the value of the loss function, thus forcing the selection of smaller weights. For the models presented in this work, the only regularization applied to the models was λ as it proved to be the most relevant for the problem at hand. Both dropout and the L2 penalty on the skip connection were fixed at zero and a range of λ values were tested through the cross-validation strategy defined in A.1.1.

Figure A.2 shows the regularization path of dataset d8 in the two well scenario in Chapter 3. The top panel shows the performance score of each of the cross-validation folds as well as the average performance of all folds plotted against their regularization λ. The value of λ with the best average score across folds is picked and the model is retrained using that λ. The bottom panel of Figure A.2 shows the number of input features that are deemed relevant for each value of λ. In this case as λ increases, the number of features decreases until the optimal set of eight features are chosen by the model.

A.1.3 Hierarchy parameter

As mentioned previously, the parameter M controls the strength of the linear and nonlinear components. When $M = 0$, only the residual layer is active and the formulation is that of the Lasso. Alternatively, when $M \to +\infty$, the unregularized neural network is obtained.

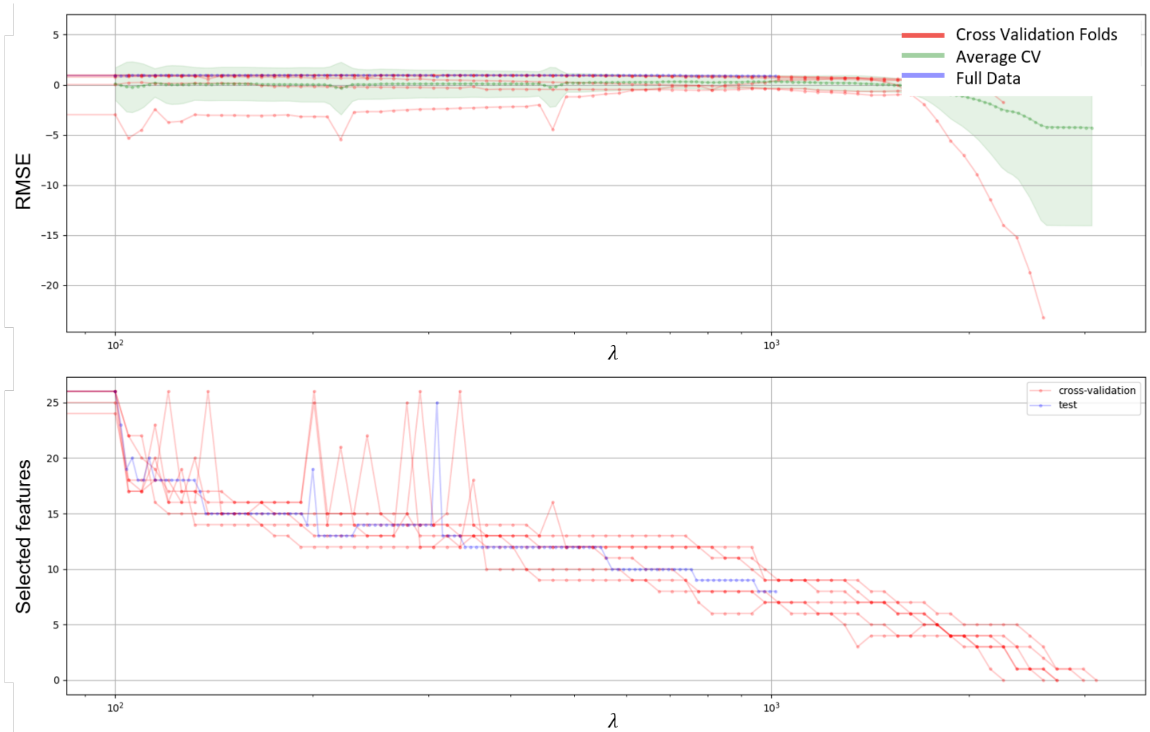

Figure A.2: Regularization path of a dataset for the LassoNet in the two-well scenario for modeling pressure data

The LassoNet authors recommend a value for $M = 10$. However this proved to be a very low value for this application. Using such low values resulted in models with high bias, specially when more features were being used. Values for M between 100-300 proved to be more successful in capturing the relevant behavior. Values larger than 300 usually resulted in models prone to overfitting, even with the use of cross-validation.

Appendix B

Flow Rate Reconstruction in Real Data

B.1 Volve Dataset

A test of the flow rate modeling methodology was done using real production data from the Volve oil field in the North Sea. The Volve field was produced from 2008 to 2016 by Statoil, (now Equinor) and was developed with 27 wells. The field was produced using water injection for pressure support and the production wells recovered a mixture of oil, gas and condensate.

To test the proposed methodology, well 15/9-F-11 was chosen. The well data contained a mixture of oil, gas and water flow rates as well as down hole and well head pressure readings. The dataset covered over three years of data with daily data points. Figure B.1 shows the production history for the well.

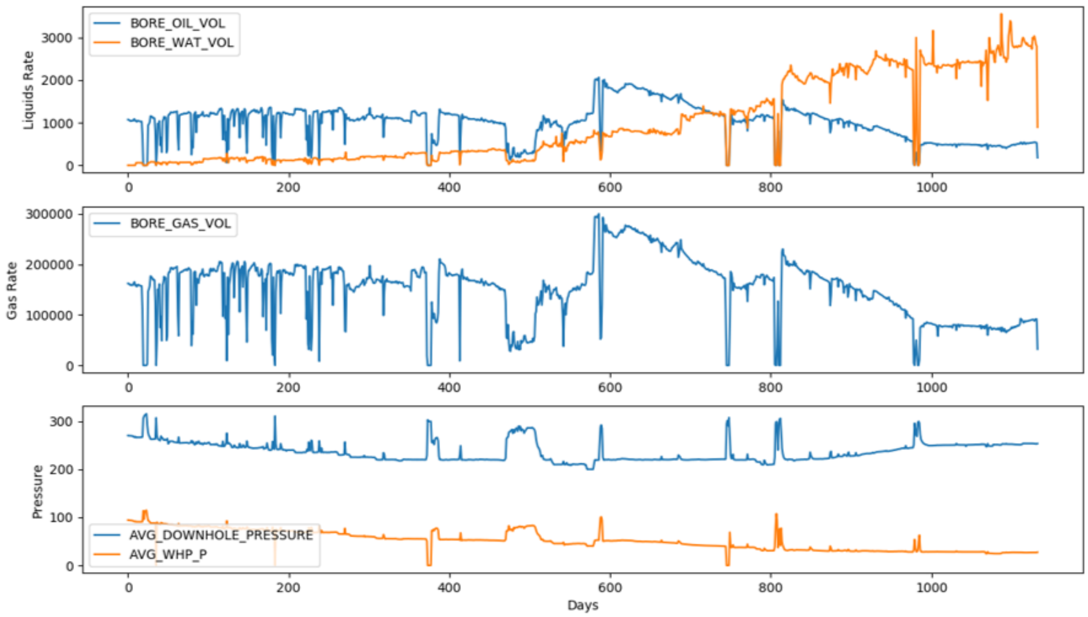

Figure B.1: Production history of Well 15/9-F-11 from the Volve oil field. The top panel shows the liquids production history, the middle panel contains the gas production history and the bottom panel shows pressure down hole and well head data.

B.2 Model Definition and Results

In contrast to the synthetic data scenarios showed in Chapter 4, the data from well 15/9-F-11 include a mixture of liquid and gas phases with both oil and water production histories. For this exercise, oil production rate was chosen as the target variable and down hole pressure data was used to generate input variables. The fist month of production data was ignored to eliminate anomalies in flow rate history not consistent with the rest of the dataset.

A model using the LassoNet and pressure MODWT-MRA features was built (Model D). The specific model inputs were the MODWT-MRA 9-level decomposition of down hole pressure. The choice of 9 MODWT-MRA levels was due to data length constraints as the total length of the data was 1,133 data points. To train the model, 80% of the data was used and the last 20% was set aside as test set. Table B.1 shows the model specifications for this scenario.

Table B.1: Model descriptions for the two-well scenario with missing flow rate data

Model Name	Regression Method	Features	Target
DF	LassoNet	Down hole pressure MODWT-MRA	Oil rate

Figure B.2 shows the model results for well 15/9-F-11. It is noticeable that the model is able to capture most of the variation in oil flow rate. In both train and test sections of the data, the large sudden changes in flow rate are accurately captured by the model. However, the model overestimates the oil rate in the latter section of the test data when the pressure data is slowly varying. It must be noted that this scenario is being trained with less data than the synthetic scenarios as data from well 15/9-F-11 was captured daily as opposed to hourly. Synthetic data scenarios had over 8,000 data points and in this case, well 15/9-F-11 only has 1,133 total data points. Those differences in data length may account partly for the less accurate performance of the model when compared to synthetic data scenarios. Moreover, in the Volve reservoir, water injection was used to provide pressure support, which highlights the added complexities of using real data.

B.3 Missing Data Scenario

A missing data scenario involving well 15/9-F-11 was also tested using the same model configuration. However, in this scenario 25% of the data was removed and the gaps were linearly interpolated. Pressure data was kept untouched and the 9-level MODWT-MRA decomposition was applied to create the model features. Table B.2 shows the specifications of the model.

Table B.2: Model descriptions for the two-well scenario with missing flow rate data

Model Name	Regression Method	Features	Target
DF	LassoNet	Down hole pressure MODWT-MRA (Interpolated)	Oil rate

Figure B.3 shows the result of the modeling for well well 15/9-F-11 with missing data. It is clear

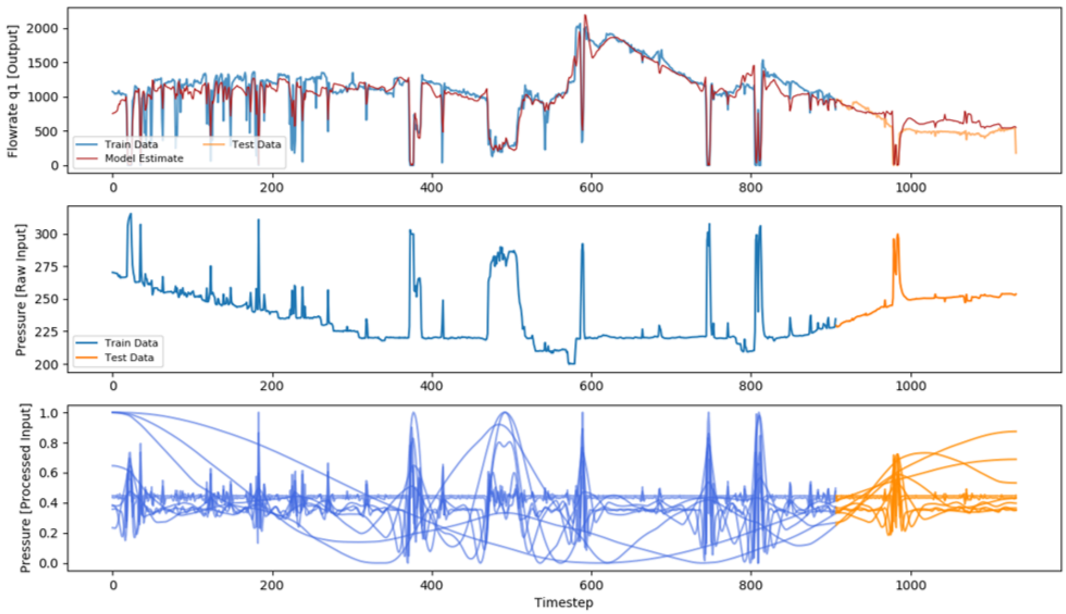

Figure B.2: Model results for well 15/9-F-11 from the Volve oil field. The top panel shows oil flow rate, middle panel shows down hole pressure history and bottom panel shows the pressure MODWT-MRA input features.

that the performance of the model degrades substantially when there is missing oil flow rate data. The interpolation done to the target data introduces additional bias to the data and the LassoNet is unable to fully capture the well behavior with the limited amount of existing data. Moreover, the overestimation in oil flow rate in the test set is much more significant than in the case without missing data.

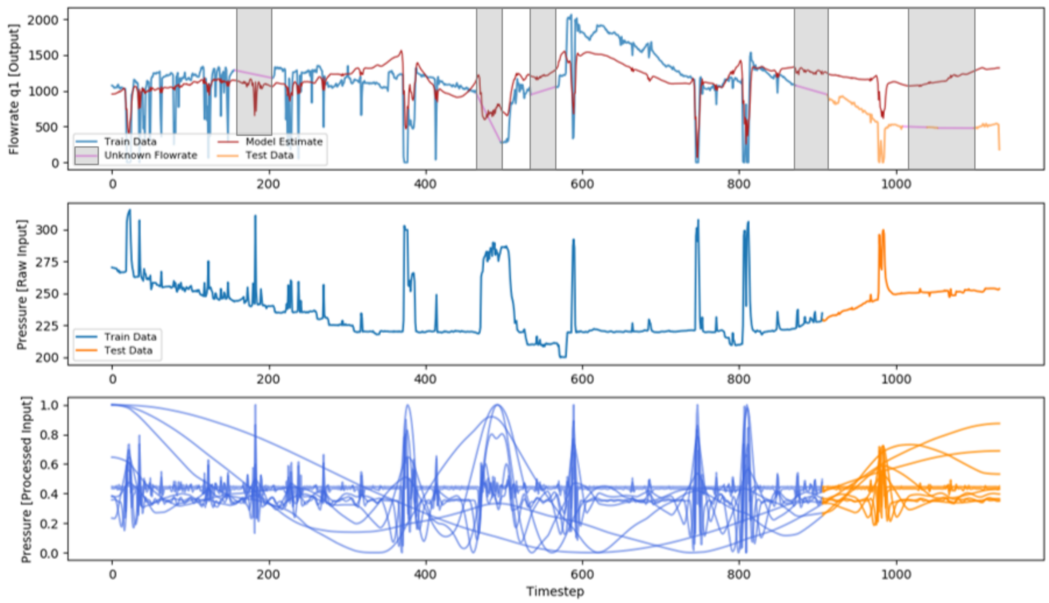

Figure B.3: Model results for well 15/9-F-11 with missing oil flow rate data. The top panel shows oil flow rate, middle panel shows down hole pressure history and bottom panel shows the pressure MODWT-MRA input features.

B.3.1 Including water production rate as input

To try to improve the performance results, another variation of the model was considered in the missing data scenario. In this case, both pressure and water production rate were used as model inputs. The oil flow rate missing data was kept at 25%. The model specifications are shown in Table B.3. Water production rate in this scenario does not contain missing data.

Table B.3: Model descriptions for the two-well scenario with missing flow rate data

Model Name	**Regression Method**	**Features**	**Target**
DF-W	LassoNet	Down hole pressure MODWT-MRA (Interpolated) Water production rate MODWT-MRA	Oil rate

Figure B.4 shows the results of model DF-W which includes both pressure and water rate input features. It is evident that using water rate as input improves the oil rate estimate significantly in the presence of missing data. However, in this case it also creates higher variability in the test estimate, with estimated oil production spikes that do not match the real data.

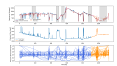

Figure B.4: Model results for well 15/9-F-11 with missing oil data. The top panel shows oil flow rate, middle panel shows down hole pressure history and bottom panel shows the pressure and water rate MODWT-MRA input features.

B.4 Conclusions

The exercise presented in this section shows the added complexities of modeling flow rate using real field production data. In this scenario, the available data was smaller than previous synthetic cases even if it covered a longer real time span. The smaller size of the dataset impacted the capacity of the model to capture complex behavior with missing data, specially with the missing data being in the target variable (oil flow rate). Moreover, real production datasets can contain multiple phases and types of liquids such as oil, gas and water rates. The data can also reflect the effects of the production strategy such as the pressure support caused by water injection.

Nonetheless, we showed that the same methodology of decomposing production time series using the MODWT-MRA can be extended and combined with other kinds of data such as water flow rate. In combination with a shrinkage model such as LassoNet, adding extra variables can help with model performance when there is missing data.

Bibliography

I. Aizenberg, L. Sheremetov, L. Villa-Vargas, and J. Martinez-Muñoz. Multilayer neural network with multi-valued neurons in time series forecasting of oil production. *Neurocomputing*, 175:980–989, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.06.092. URL `https://www.sciencedirect.com/science/article/pii/S0925231215016008`.

A. Alakeely and R. Horne. Application of deep learning methods to estimate multiphase flow rate in producing wells using surface measurements. *Journal of Petroleum Science and Engineering*, 205:108936, 2021. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2021.108936. URL `https://www.sciencedirect.com/science/article/pii/S0920410521005970`.

J. Allen. Short term spectral analysis, synthesis, and modification by discrete fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(3):235–238, 1977. doi: 10.1109/TASSP.1977.1162950.

Athichanagorn et al., 1999. *Processing and Interpretation of Long-term Data from Permanent Downhole Pressure Gauges*, volume All Days of *SPE Annual Technical Conference and Exhibition*, 10 1999. doi: 10.2118/56419-MS. URL `https://doi.org/10.2118/56419-MS`. SPE-56419-MS.

Bernasconi et al., 1999. *Compression of Downhole Data*, volume All Days of *SPE/IADC Drilling*

Conference and Exhibition, 03 1999. doi: 10.2118/52806-MS. URL https://doi.org/10.2118/52806-MS. SPE-52806-MS.

Cao et al., 2016. *Data Driven Production Forecasting Using Machine Learning*, volume Day 2 Thu, June 02, 2016 of *SPE Argentina Exploration and Production of Unconventional Resources Symposium*, 06 2016. doi: 10.2118/180984-MS. URL https://doi.org/10.2118/180984-MS. D021S006R001.

I. A. Chaikine and I. D. Gates. A machine learning model for predicting multi-stage horizontal well production. *Journal of Petroleum Science and Engineering*, 198:108133, 2021. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2020.108133. URL https://www.sciencedirect.com/science/article/pii/S0920410520311876.

H. Chu, X. Liao, P. Dong, Z. Chen, X. Zhao, and J. Zou. An automatic classification method of well testing plot based on convolutional neural network (cnn). *Energies*, 12(15), 2019. ISSN 1996-1073. doi: 10.3390/en12152846. URL https://www.mdpi.com/1996-1073/12/15/2846.

C. R. Cornish, C. S. Bretherton, and D. B. Percival. Maximal overlap wavelet statistical analysis with application to atmospheric turbulence. *Boundary-Layer Meteorology*, 119(2):339–374, May 2006. ISSN 1573-1472. doi: 10.1007/s10546-005-9011-y. URL https://doi.org/10.1007/s10546-005-9011-y.

A. Davtyan, A. Rodin, I. Muchnik, and A. Romashkin. Oil production forecast models based on sliding window regression. *Journal of Petroleum Science and Engineering*, 195:107916, 2020. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2020.107916. URL https://www.sciencedirect.com/science/article/pii/S0920410520309712.

R. de Oliveira Werneck, R. Prates, R. Moura, M. M. Gonçalves, M. Castro, A. Soriano-Vargas, P. Ribeiro Mendes Júnior, M. M. Hossain, M. F. Zampieri, A. Ferreira, A. Davólio, D. Schiozer, and A. Rocha. Data-driven deep-learning forecasting for oil production and pressure. *Journal of Petroleum Science and Engineering*, 210:109937, 2022. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2021.109937. URL `https://www.sciencedirect.com/science/article/pii/S0920410521015515`.

P. Dong, Z. Chen, X. Liao, and W. Yu. Application of deep learning on well-test interpretation for identifying pressure behavior and characterizing reservoirs. *Journal of Petroleum Science and Engineering*, 208:109264, 2022. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2021.109264. URL `https://www.sciencedirect.com/science/article/pii/S0920410521009190`.

P. K. Dunn and G. K. Smyth. Series evaluation of tweedie exponential dispersion model densities. *Statistics and Computing*, 15(4):267–280, Oct 2005. ISSN 1573-1375. doi: 10.1007/s11222-005-4070-y. URL `https://doi.org/10.1007/s11222-005-4070-y`.

P. K. Dunn and G. K. Smyth. Evaluation of tweedie exponential dispersion model densities by fourier inversion. *Statistics and Computing*, 18(1):73–86, Mar 2008. ISSN 1573-1375. doi: 10.1007/s11222-007-9039-6. URL `https://doi.org/10.1007/s11222-007-9039-6`.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `https://www.jstatsoft.org/v33/i01/`.

R. Gençay, F. Selçuk, and B. Whitcher. 4 - discrete wavelet transforms. In R. Gençay, F. Selçuk, and B. Whitcher, editors, *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*, pages 96–160. Academic Press, San Diego, 2002. ISBN 978-0-12-279670-8. doi: https://doi.org/10.1016/B978-012279670-8.50007-0. URL `https://www.sciencedirect.com/science/article/pii/B9780122796708500070`.

A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, Sept. 1910.

R. G. Jacquot, J. W. Steadman, and C. N. Rhodine. The gaver-stehfest algorithm for approximate inversion of laplace transforms. *IEEE Circuits and Systems Magazine*, 5(1):4–8, 1983. doi: 10.1109/MCAS.1983.6323897.

Z. L. Jin, Y. Liu, and L. J. Durlofsky. Deep-learning-based surrogate model for reservoir simulation with time-varying well controls. *Journal of Petroleum Science and Engineering*, 192: 107273, 2020. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2020.107273. URL `https://www.sciencedirect.com/science/article/pii/S0920410520303533`.

Kikani et al., 1998. *Multi-resolution Analysis of Long-Term Pressure Transient Data Using Wavelet Methods*, volume All Days of *SPE Annual Technical Conference and Exhibition*, 09 1998. doi: 10.2118/48966-MS. URL `https://doi.org/10.2118/48966-MS`. SPE-48966-MS.

Y. D. Kim and L. J. Durlofsky. A Recurrent Neural Network–Based Proxy Model for Well-Control Optimization with Nonlinear Output Constraints. *SPE Journal*, 26(04):1837–1857, 08 2021. ISSN 1086-055X. doi: 10.2118/203980-PA. URL `https://doi.org/10.2118/203980-PA`.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. URL `https://arxiv.org/abs/1412.6980`.

Kubota et al., 2019. *Machine Learning Forecasts Oil Rate in Mature Onshore Field Jointly Driven by Water and Steam Injection*, volume Day 2 Tue, October 01, 2019 of *SPE Annual Technical Conference and Exhibition*, 09 2019. doi: 10.2118/196152-MS. URL `https://doi.org/10.2118/196152-MS`. D021S020R003.

F. Kuchuk, M. Onur, and F. Hollaender. *Pressure Transient Formation and Well Testing: Convolution, Deconvolution and Nonlinear Estimation*. Developments in Petroleum Science. Elsevier Science, 2010. ISBN 9780444529534. URL `https://books.google.com/books?id=4xg1lQEACAAJ`.

K. Lee, J. Lim, S. Ahn, and J. Kim. Feature extraction using a deep learning algorithm for uncertainty quantification of channelized reservoirs. *Journal of Petroleum Science and Engineering*, 171:1007–1022, 2018. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2018.07.070. URL `https://www.sciencedirect.com/science/article/pii/S0920410518306454`.

I. Lemhadri, F. Ruan, L. Abraham, and R. Tibshirani. Lassonet: A neural network with feature sparsity. *Journal of Machine Learning Research*, 22(127):1–29, 2021. URL `http://jmlr.org/papers/v22/20-848.html`.

Li et al., 2021. *Reconstruction of Missing Segments in Well Data History Using Data Analytics*, volume Day 3 Wed, November 17, 2021 of *Abu Dhabi International Petroleum Exhibition and Conference*, 11 2021. doi: 10.2118/208137-MS. URL `https://doi.org/10.2118/208137-MS`. D032S232R002.

X. Liu, D. Li, J. Yang, W. Zha, Z. Zhou, L. Gao, and J. Han. Automatic well test interpretation based on convolutional neural network for infinite reservoir. *Journal of Petroleum Science and Engineering*, 195:107618, 2020. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2020.107618. URL `https://www.sciencedirect.com/science/article/pii/S0920410520306860`.

S. Mallat and C. Mallat. *A Wavelet Tour of Signal Processing*. Elsevier Science & Technology, Burlington, UNITED STATES, 1999. ISBN 9780080520834. URL `http://ebookcentral.proquest.com/lib/stanford-ebooks/detail.action?docID=319092`.

J. Morlet, G. Arens, E. Fourgeau, and D. Giard. Wave propagation and sampling theory; Part I, Complex signal and scattering in multilayered media. *Geophysics*, 47(2):203–221, 02 1982a. ISSN 0016-8033. doi: 10.1190/1.1441328. URL `https://doi.org/10.1190/1.1441328`.

J. Morlet, G. Arens, E. Fourgeau, and D. Giard. Wave propagation and sampling theory—part ii: Sampling theory and complex waves. *GEOPHYSICS*, 47(2):222–236, 1982b. doi: 10.1190/1.1441329. URL `https://doi.org/10.1190/1.1441329`.

J. Morlet, G. Arens, E. Fourgeau, and D. Glard. Wave propagation and sampling theory—part i: Complex signal and scattering in multilayered media. *GEOPHYSICS*, 47(2):203–221, 1982c. doi: 10.1190/1.1441328. URL `https://doi.org/10.1190/1.1441328`.

Ouyang et al., 2002. *Improving Permanent Downhole Gauge (PDG) Data Processing via Wavelet Analysis*, volume All Days of *SPE Europec featured at EAGE Conference and Exhibition*, 10 2002. doi: 10.2118/78290-MS. URL `https://doi.org/10.2118/78290-MS`. SPE-78290-MS.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

D. B. Percival. *Analysis of Geophysical Time Series Using Discrete Wavelet Transforms: An Overview*, pages 61–79. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-78938-3. doi: 10.1007/978-3-540-78938-3_4. URL `https://doi.org/10.1007/978-3-540-78938-3_4`.

D. B. Percival and A. T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2000. doi: 10.1017/CBO9780511841040.

Razak et al., 2021. *Transfer Learning with Recurrent Neural Networks for Long-term Production Forecasting in Unconventional Reservoirs*, volume Day 1 Mon, July 26, 2021 of *SPE/AAPG/SEG Unconventional Resources Technology Conference*, 07 2021. doi: 10.15530/urtec-2021-5687. URL https://doi.org/10.15530/urtec-2021-5687. D011S001R002.

P. M. Ribeiro, A. P. Pires, E. A. P. Oliveira, and J. G. Ferroni. Use of Wavelet Transform in Pressure-Data Treatment. *SPE Production & Operations*, 23(01):24–31, 02 2008. ISSN 1930-1855. doi: 10.2118/100719-PA. URL https://doi.org/10.2118/100719-PA.

S. Sun and T. Zhang. Chapter one - introduction. In S. Sun and T. Zhang, editors, *Reservoir Simulations*, pages 1–22. Gulf Professional Publishing, 2020. ISBN 978-0-12-820957-8. doi: https://doi.org/10.1016/B978-0-12-820957-8.00001-0. URL https://www.sciencedirect.com/science/article/pii/B9780128209578000010.

C. Tian and R. N. Horne. Applying Machine-Learning Techniques To Interpret Flow-Rate, Pressure, and Temperature Data From Permanent Downhole Gauges. *SPE Reservoir Evaluation and Engineering*, 22(02):386–401, 01 2019. ISSN 1094-6470. doi: 10.2118/174034-PA. URL https://doi.org/10.2118/174034-PA.

Tian and Horne, 2015. *Machine Learning Applied to Multiwell Test Analysis and Flow Rate Reconstruction*, volume Day 2 Tue, September 29, 2015 of *SPE Annual Technical Conference and Exhibition*, 09 2015. doi: 10.2118/175059-MS. URL https://doi.org/10.2118/175059-MS. D021S013R006.

Tian and Horne, 2016. *Inferring Interwell Connectivity Using Production Data*, volume Day 3 Wed, September 28, 2016 of *SPE Annual Technical Conference and Exhibition*, 09 2016. doi: 10.2118/181556-MS. URL https://doi.org/10.2118/181556-MS. D031S051R004.

Tian and Horne, 2017. *Recurrent Neural Networks for Permanent Downhole Gauge Data Analysis*, volume Day 1 Mon, October 09, 2017 of *SPE Annual Technical Conference and Exhibition*, 10 2017. doi: 10.2118/187181-MS. URL `https://doi.org/10.2118/187181-MS`. D011S008R007.

C. Wei, R. Huang, M. Ding, J. Yang, and L. Xiong. Characterization of saturation and pressure distribution based on deep learning for a typical carbonate reservoir in the middle east. *Journal of Petroleum Science and Engineering*, 213:110442, 2022. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2022.110442. URL `https://www.sciencedirect.com/science/article/pii/S092041052200328X`.

K. E. Willcox, O. Ghattas, and P. Heimbach. The imperative of physics-based modeling and inverse theory in computational science. *Nature Computational Science*, 1(3):166–168, Mar 2021. ISSN 2662-8457. doi: 10.1038/s43588-021-00040-z. URL `https://doi.org/10.1038/s43588-021-00040-z`.

Z. Yuan, H. Huang, Y. Jiang, and J. Li. Hybrid deep neural networks for reservoir production prediction. *Journal of Petroleum Science and Engineering*, 197:108111, 2021. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2020.108111. URL `https://www.sciencedirect.com/science/article/pii/S0920410520311657`.

Zhan et al., 2019. *Application of Machine Learning for Production Forecasting for Unconventional Resources*, volume Day 2 Tue, July 23, 2019 of *SPE/AAPG/SEG Unconventional Resources Technology Conference*, 07 2019. doi: 10.15530/urtec-2019-47. URL `https://doi.org/10.15530/urtec-2019-47`. D023S029R004.