ESTIMATION OF DRILL-STRING TORSIONAL VIBRATION SEVERITY USING FIELD DATA AND MACHINE LEARNING

Matheus Vera Di Vaio

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Mecânica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Mecânica.

Orientador: Fernando Augusto de Noronha Castro Pinto

Rio de Janeiro
Novembro de 2019

ESTIMATION OF DRILL-STRING TORSIONAL VIBRATION SEVERITY USING FIELD DATA AND MACHINE LEARNING

Matheus Vera Di Vaio

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA MECÂNICA.

Examinada por:

Prof. Fernando Augusto de Noronha Castro Pinto, Dr.-Ing.

Prof. Thiago Gamboa Ritto, D.Sc.

Prof. José Manoel de Seixas, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2019

Di Vaio, Matheus Vera

Estimation of drill-string torsional vibration severity using field data and machine learning/Matheus Vera Di Vaio. – Rio de Janeiro: UFRJ/COPPE, 2019.

XII, 82 p.: il.; 29, 7cm.

Orientador: Fernando Augusto de Noronha Castro Pinto

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Mecânica, 2019.

Referências Bibliográficas: p. 70 – 76.

1. Drilling. 2. Field Data. 3. Torsional vibration. 4. Machine learning. I. Pinto, Fernando Augusto de Noronha Castro. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Mecânica. III. Título.

"Fear tends to come from ignorance. Once I knew what the problem was, it was just a problem, nothing to fear."
Patrick Rothfuss

Acknowledgments

I would like to thank the following people, without whom I would not have been able to complete this research, and without whom I would not have come this far!

To the low-level BRDrilling team, Daniel, Fabían, Lucas, Raphael and the outsider Rodrigo, which made every day to-dos enjoyable and challenging.

To Prof. Thiago Ritto, who was in charge of the team, and constantly kept up with the work and discussions.

To my mentor, Prof. Fernando Castro Pinto, who provided important insights, and guide lines.

To Petrobras, specially to Emílio, which provided all the means to make this research possible.

To all the special friends made in my experience through this university.

And to the most important ones, my parents, who provided full support in the whole process, gave important advice, and had a lot of patience.

You all were awesome companions.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ESTIMAÇÃO DA SEVERIDADE DE VIBRAÇÃO TORCIONAL DE COLUNA DE PERFURAÇÃO VIA DADOS DE CAMPO E APRENDIZADO DE MAQUINA

Matheus Vera Di Vaio

Novembro/2019

Orientador: Fernando Augusto de Noronha Castro Pinto

Programa: Engenharia Mecânica

Este trabalho tem como objetivo desenvolver um método para estimar em tempo real a vibração torcional da coluna de perfuração. Essa estimativa durante a operação de perfuração fornece informação importante ao operador para que ele possa controlar os parâmetros de perfuração de forma assertiva. Para isso, é feita uma apresentação dos poços e dos dados possuídos. Uma adaptação do PCA é proposta para fazer o pré-processamento dos dados que alimentam uma rede neural profunda proposta. Por fim, o método é testado em quatro casos distintos, cada um com suas características singulares, com ou sem extrapolação de domínio. A ferramenta de pré-processamento proposta e o uso dos dados brutos têm seus resultados comparados e avaliados. A conclusão fornece um resumo e algumas discussões sobre os resultados, suas limitações e características.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ESTIMATION OF DRILL-STRING TORSIONAL VIBRATION SEVERITY USING FIELD DATA AND MACHINE LEARNING

Matheus Vera Di Vaio

November/2019

Advisor: Fernando Augusto de Noronha Castro Pinto

Department: Mechanical Engineering

This work aims to develop a method for real-time estimation of the drill string torsional vibration. This estimation during the drilling operation gives important information to the operator so that he can control the drilling parameters assertively. For that, it is made a presentation of the wells, and the possessed data. An adaptation of the PCA is proposed to make the preprocessing of the data that feeds a proposed deep neural network. Finally, the method is tested through four distinct cases, each one with its singular characteristics, with or not domain extrapolation. The proposed preprocessing tool and the use of the raw data have its results compared and evaluated. The conclusion provides a resume and some discussions of the results, its limitations, and its characteristics.

Contents

List of Figures **x**

List of Tables **xii**

1 Introduction **1**

1.1 Motivation 1
1.2 Dissertation Objective and Organization 2
1.3 Overview of a drilling rig 3
1.4 Vibrations in drill-string 4
1.5 Basic principles of machine learning 5
1.6 Artificial neural networks 7
1.6.1 Long-Short-Term-Memory (LSTM) Neural Networks 11
1.6.2 Extreme learning machine (ELM) 11
1.6.3 Deep Learning 12
1.7 Bibliographic Review 12
1.7.1 Machine learning applied to sequential data 13
1.7.2 Machine learning applied to the fossil fuel exploration industry 16
1.7.3 Considerations 17

2 Data preparation **19**

2.1 Overview of the field drilling data 19
2.2 Introducing the wells 21
2.3 Data processing 23
2.3.1 Data synchronization and cutting 23
2.3.2 Creation of indexes 25
2.3.3 Analysis in frequency domain 28
2.3.4 Torsional vibration severity map 31

3 Deep Neural Network Model **39**

3.1 The architecture 39
3.1.1 Input layer 39
3.1.2 Hidden layers 40

3.1.3 Dropout layers . . . 41
3.1.4 Output layer . . . 42
3.1.5 Weights initialization . . . 43
3.1.6 Loss function . . . 44
3.1.7 Optimizer . . . 44
3.2 Training data equalization: . . . 45
3.3 Batch size . . . 45

4 Results **47**
4.1 Case 1: Domain extrapolation at Rock B . . . 50
4.2 Case 2: No domain extrapolation, not using Well Br2 . . . 57
4.3 Case 3: Domain extrapolation at Well Br2 . . . 60
4.4 Case 4: No domain extrapolation, using Well Br2 . . . 64

5 Conclusions **68**
5.1 Future Steps . . . 69

Bibliography **70**

List of Figures

1.1 A schematic view of a drilling rig. Reproduced from [1]. 3
1.2 Drill-string vibration modes, adapted from [2]. 5
1.3 Machine learning types. 6
1.4 ANN scheme. 8
1.5 Neuron scheme. 8
1.6 Traditional activation functions. 9
1.7 Examples of underfitting, overfitting and proper fitting. Source: Adapted from [3]. 10
1.8 Common error curve. 10
1.9 Traditional deep learning scheme. 12
1.10 The prediction of traffic flow with different regression analysis. Source: [4]. 13
1.11 Main Framework of Real-Time Prediction of Solar Radiation. Source: [5]. 14
1.12 RMSE of DO with different time steps. Source: [6]. 14
1.13 Accuracy of different methods in 10 trails. Source [7] 15
1.14 Scheme of the data processing tool created by [8] to optimize drilling. 17

2.1 Transition from Rock A to Rock B at Well A. 22
2.2 Drilling start. 24
2.3 One drilling interval characterization. 25
2.4 Calculated values of S_S, $STOR_{var}$ and ROP. 27
2.5 Calculated values of SS, $STOR_{var}$ and ROP. 27
2.6 Wavelet Transform of the surface torque at the intervals A, B and C. 29
2.7 Wavelet Transform of the RPM on bit at the intervals A, B and C. . 30
2.8 Torsional vibration severity map. 31
2.9 Torsional vibration severity map. Adapted from [2] 32
2.10 Map of the Global PCA of the Well 36
2.11 Adapted PCA map. 38

3.1 ReLU vs PReLU. 41

3.2 Example of Neural Network with dropout. Neurons randomly dropped. 42

4.1 Default training error curve: train vs test error datasets. 48
4.2 Calculated S_S for all the Wells. 49
4.3 Case 1: Raw data. 52
4.4 Case 1: Adapted PCA. 53
4.5 Case 1 error evaluation. 56
4.6 Case 2: Raw data. 57
4.7 Case 2: Adapted PCA. 58
4.8 Case 2 error evaluation. 59
4.9 Case 3: Raw data. 60
4.10 Case 3: Adapted PCA. 61
4.11 Case 3 error evaluation. 63
4.12 Case 4: Raw data. 64
4.13 Case 4: Adapted PCA. 65
4.14 Case 4 error evaluation. 67

5.1 Histogram with 5000 simulation of sample 1000 (Rock A) of the test dataset of Case 1 80
5.2 Histogram with 5000 simulation of sample 7000 (Rock B) of the test dataset of Case 1 81
5.3 Histogram with 300 simulation of sample 1000 (Rock A) of the test dataset of Case 1 81
5.4 Histogram with 300 simulation of sample 7000 (Rock B) of the test dataset of Case 1 82

List of Tables

2.1 Data Description. 20
2.2 Overview of the drilling of the Well A and B at the 17,5" phase. . . . 21

5.1 Training and testing errors obtained in each case, considering an interval of 2 and 3 RMS deviations. 68
5.2 Neural Network with Raw Data architecture convergence process . . . 78
5.3 Neural Network with Adapted PCA architecture convergence process 79

Chapter 1

Introduction

1.1 Motivation

The exploration of oil and gas is increasingly being done in ultra-deep waters and ultra-deep wells. As a result of this, the difficulty involved in drilling is increasing and making necessary the use of new technologies and more sophisticated analysis.

One of the principal complications involved in the drilling operations is caused by the vibrations and amplified by the increased length of the drill-pipe, harder formations, bore-hole instabilities and so on. Vibrations steal energy from the system and reduce efficiency, leading to a low rate of penetration and may even cause the failure of downhole equipment. These vibrations are unavoidable because of the non-predictability of the external forces acting on it, mainly the one caused by the drill-bit cutting interaction.

Real-time information of downhole vibration scenario would be of great help for the drilling operator. More assertive decisions about the drilling parameters could be taken and the operation could be led to better states, increasing the rate of penetration (ROP) and diminishing the vibration intensities. The uncertainties and variations in environmental factors, like lithology, for example, make the interactions of the drill-string with the borehole extremely difficult to model and computationally expensive. Another complicating factor is caused by the difficulty in measuring and transmitting data from the drill-bit. All these factors lead to a lack of information in real-time of what is happening downhole for the surface, making it even harder for the drilling operator to take assertive actions.

On the other hand, with the advances in computational power and the storage capacity of great amounts of data, techniques in the machine learning field are currently of great interest in a vast amount of areas. It has shown significant results in complex problems, such as medicine [9, 10], engineering [11], in extreme difficult games like Go [12] and even in oils exploration industry as shown in Chapter 2.

1.2 Dissertation Objective and Organization

The objective of this work is to explore the drilling data from two ultra-deep oil wells provided by Petrobras then apply Machine Learning techniques to estimate the torsional vibration severity factor (SS) only with surface data. Traditional data processing tools were applied to better understand the properties of the data and prepare for the Neural Network implementation. The dissertation aims to create a method to estimate the torsional vibration severity factor (S_S) while drilling based just on surface data. With this information, the drilling operator would have important information about the severity of the S_S to better adjust the drilling parameters during the operation.

The torsional vibration severity estimation is going to be the main focus of this work for two reasons. First, axial and lateral vibrations were very low in magnitude on one well and secondly in other well the provided data from downhole did not contain the tool necessary to measure axial and lateral vibrations.

The S_S is calculated with the recorded data from the downhole measurement tools. It is proposed a method based on a deep neural network (DNN) to estimate the S_S.

The preprocessing of the data that will serve as input to the neural network is of extreme importance. When poorly done may cause the method to give not the optimal results. To be able to better understand the nature of the data and, at first, choose the best preprocessing approach, some traditional tools were applied and the results were commented. Next, two different preprocessing approaches were compared minding the DNN results.

In Chapter 1, is presented an introduction about the drilling operation and vibrations, a brief introduction to machine learning is made and there is the bibliography review. Where the most relevant works in the literature that deals with sequential data and are explored. Works in the oil exploration industry that use machine learning techniques are also reviewed. Chapter 2 presents an introduction to the data, to the data preprocessing, and indexes creation. It also contains the explanation of the developed preprocessing method. In Chapter 3, the deep neural network developed for this work is explained. In Chapter 4, the results obtained by the whole proposed method are discussed. In Chapter 5 is the conclusion of this work with discussions of the results obtained and the proposal of next steps. Appendix A contains the tests made in the definition of the architeture of the ANN. Appendix B contains an evaluation of the ANN's output distribution in two different scenarios.

1.3 Overview of a drilling rig

To make possible the exploration of fossil fuel out of the natural reservoirs, it is necessary to have a rotary machine called drilling rig. A common schematic of a drilling rig is shown in Fig. 1.1 and its principal components are: Drill-string, hoisting system, top rotary system, and the drilling mud. The drilling rig also contains the equipment necessary for more conventional functions, such as power generators and blow out preventers.

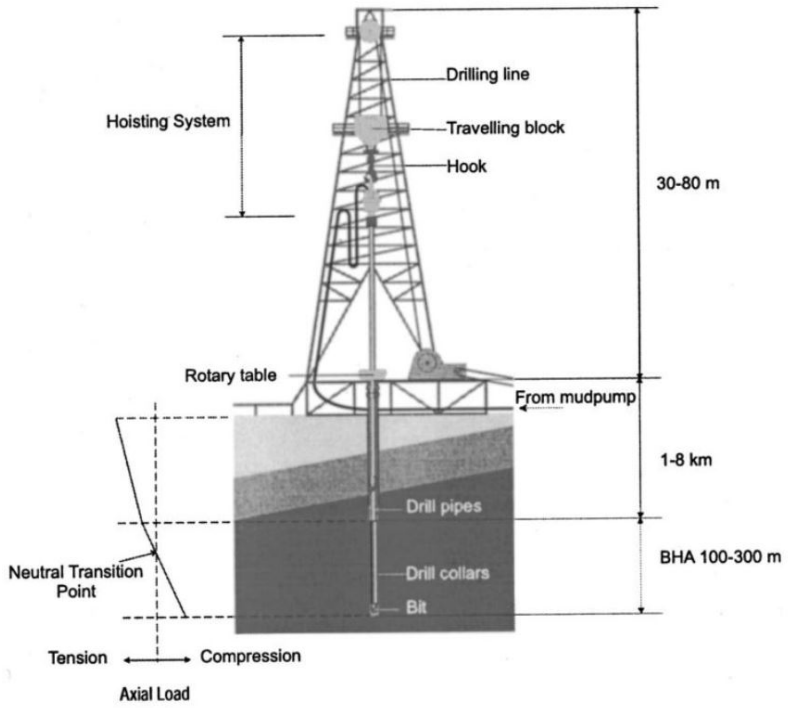


Figure 1.1: A schematic view of a drilling rig. Reproduced from [1].

The drill-string is the name given to a system that transmits the power from the surface to the downhole. It is slender, with a length to diameter ratio much smaller than a human hair. It is basically composed of two parts, the drill pipes, which are a sequence of tubes and the Bottom Hole Assembly (BHA). This assembly is composed of much thicker and heavier tubes, compared to the previous tubes and is called the heavyweight drill pipe (HWDP), Measurement While Drilling (MWD) equipment, the drill-bit and a variety of equipments used for many purposes like deviation control, shocking subs and others. Nowadays this assembly of parts can reach a depths of 9 kilometers which can lead to destructive vibrations if not well controlled from the surface.

The hoisting system is responsible for controlling the hook load that is applied at the top of the drill-string and is responsible to maintain it suspended and most of it on traction. The drill-string final part is maintained on compression to reach

desired Weight On Bit (WOB) on the rock.

The rotary system is responsible for generating the power necessary to drive the drill-bit and an electric motor is more commonly used. This system can be of two types, a rotary table or a top driver.

The drilling mud is a fluid substance that is pumped at the surface and goes through the inside of the drill string down to the drill-bit. Its primary uses are to remove the cutting rock, refrigerate and lubricate the drill-bit and assure pressure in the borehole to guarantee its stability. Also, part of the data measured by the MWD system can be transmitted through pressure pulse.

1.4 Vibrations in drill-string

During the drilling operations, vibrations are induced by the interaction of the drill-string with the environment and can reach harmful levels. These interactions are external forces that are caused by the interactions of the drill-string with the borehole which are mainly the drill-bit with the rock, the restrictions imposed by the stabilizers in the BHA and the forces and torques transmitted by the top driver. High amplitude vibrations lead to a low Rate of Penetration (ROP) and potential damage to the BHA.

The drill-string vibrations are normally classified based on the axis that they occur. There are three main types of vibration, axial, lateral and torsional.

- Axial Vibration: In this type of vibration the drill-string moves along its axis of rotation. Its most dangerous type is the Bit Bounce and happens when the drill-bit impacts and get loose of the formation at a high speed. Usually, it happens at frequencies ranging from 1-10 Hz [13].

- Lateral Vibration: This type of vibration occurs transversally to the drill string's axis of rotation in the annular gap. The most critical situation is called whirl and it can be of three types, backward, forward and chaotic. The whirl occurs when the rotation center moves laterally as it rotates. The forward whirl is when the section rotates around its center in the same direction as the drill-string. The backward is when the section rotates in the opposite direction of the drill string's rotation direction. The chaotic is when the section impacts the borehole wall chaotically. Usually, it happens at frequencies ranging from 0.5 to tens of Hz [13].

- Torsional Vibration: This type of vibration happens when the drill-string rotates regularly in the surface, but irregularly downhole. The most harmful situation is when the drill-bit sticks to the rock formation while the surface

remains rotating. When the stored energy is high enough the drill-bit slips allowing the drill-bit to reach speeds of up to 10 times higher than top rotary speed. This phenomenon usually happens at frequencies ranging from 0.05 – 0.5 Hz [13].

In real-world scenarios, the vibration modes described above happen simultaneously. Describing all the physical phenomena happening in the drilling operation usually results in a lack of clarity of the parameters and high computational cost.

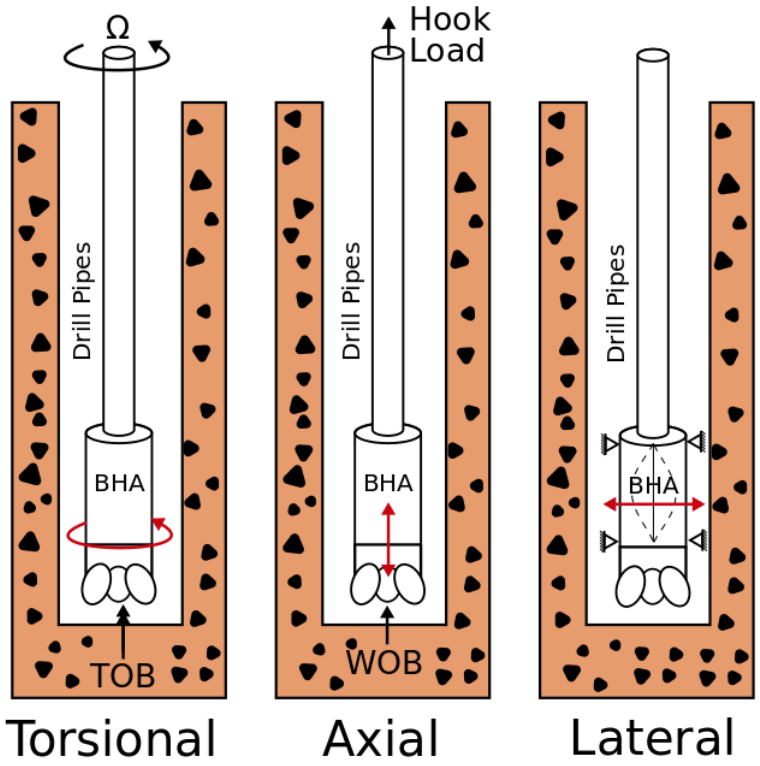


Figure 1.2: Drill-string vibration modes, adapted from [2].

1.5 Basic principles of machine learning

In this dissertation is going to be used a machine learning technique to predict de S_S, therefore this section gives a brief explanation of its basic concepts and its main techniques.

Machine learning is a branch of artificial intelligence. It is a class of algorithms that are able to learn from data. By "learn" it is meant the capacity of the algorithm to improve its performance measurement from gathering experience on doing a determined task [14]. It shall be used when dealing with some conditions. Informally speaking they are: there is a pattern; it is not possible or extremely expensive to pin it down mathematically, and there are data available [15].

Even though it is not completely agreed upon, the types of machine learning algorithms are commonly divided into categories according to their purpose [15]. In Fig. 1.3 these categories can be seen. There are several machine learning techniques in each category, each method with its pros and cons.

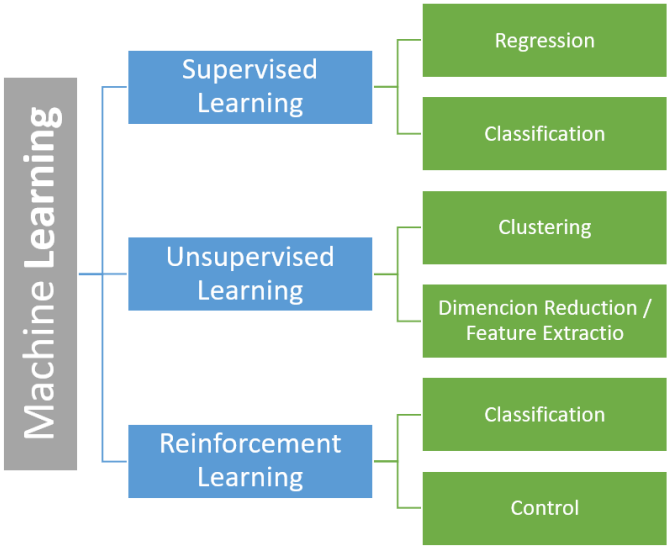


Figure 1.3: Machine learning types.

Supervised Learning

In supervised learning, the algorithm receives the data as input and during the training phase, it knows the output. The method's objective is to adapt its internal parameters to best match the inputs with the outputs, a way to see its task is as a function approximation. The types of supervised learning are:

- **Regression:** In this type of task, the algorithm is asked to predict a continuous-valued attribute based on the input. For example the prediction of the intensity of the vibrations on the drill-string.
- **Classification:** In this type of task, the algorithm has to specify to which category the input belongs to. For example the classification of the type of vibration happening during the drilling or even the vibration intensity category (low, medium, high).

Unsupervised Learning

In unsupervised learning the algorithm doesn't receive the desired output, it "discovers" it. Generally speaking, the objective is to obtain an output that preserves as most information as possible of the input data, helping the extraction of meaningful insights and features. These algorithms learn in the training stage to extract rules,

patterns and to group the data points which helps in extracting meaningful insights and better describe the data to users or even to other machine learning algorithms. These kinds of learning algorithms can be divided into two tasks.

- **Clustering:** The algorithm autonomously separates the data into clusters of similar features. For example, given a set of pet pictures, the algorithm separates it in dogs and cats.
- **Dimension Reduction; Feature Extraction:** In this type of task, the algorithm extract features from the input. Continuing the previous example, it could learn that dogs have brown eyes and rounded ears as cats have blue eyes and sharp ears.

Reinforcement Learning

In the reinforcement learning method, the algorithm aims at using the interaction with the environment to learn to take actions that do what it's intended to, maximizing its reward function. With time, the algorithm explores all the possible states and learn the best action on each. It is mostly used in control tasks.

1.6 Artificial neural networks

Among several machine learning techniques, artificial neural networks (ANN) are a class of algorithms that have been gaining a great amount of attention. They are algorithms based on the biological neural networks, with its synapses and neurons. An ANN is an array of neurons (nodes) connected with synapse that transmits the signal from a node to the next multiplied by a *weight* and summed by a *bias*. This *weight* can be interpreted as the importance of that synapse and the *bias* as a mean correction. The traditional ANN scheme, with one hidden layer, is demonstrated in Fig. 1.4.

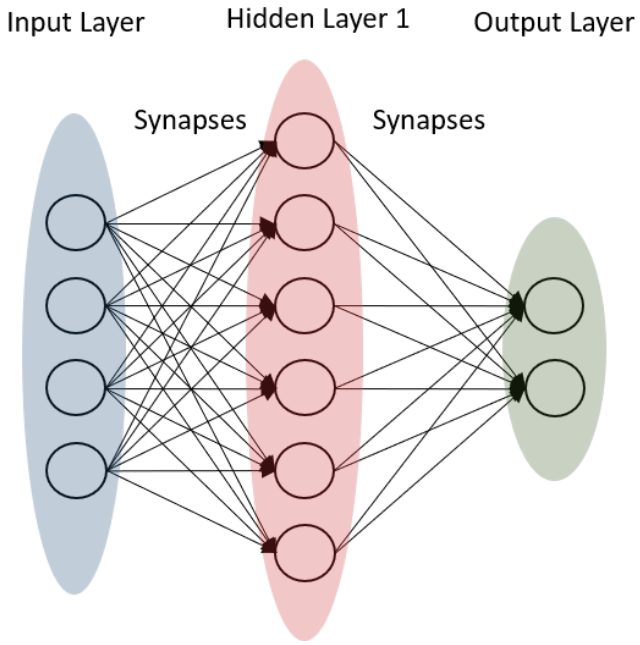


Figure 1.4: ANN scheme.

A single neuron, works the following way, after receiving the signals multiplied by its synapse's weight, the respective node sums them all and process it with the activation function as demonstrated in the Fig. 1.5.

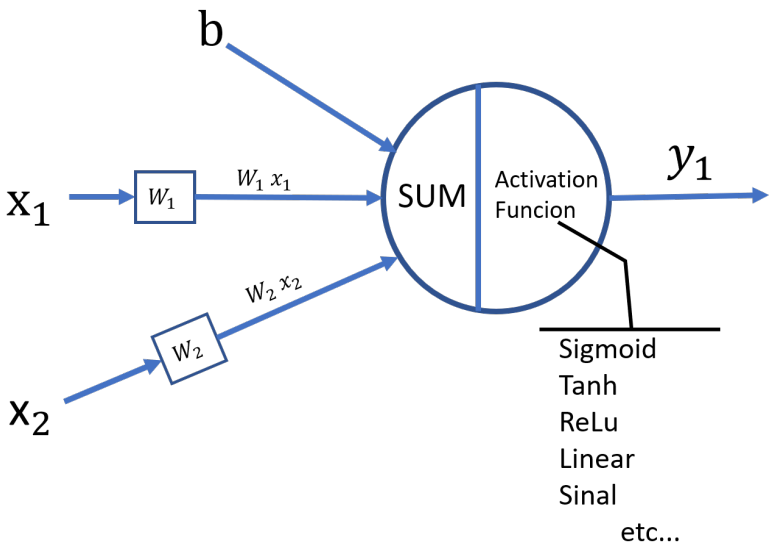


Figure 1.5: Neuron scheme.

In the figure above, b represents the *bias*, X_1 and X_2 represents the inputs and W_1 and W_2 it's respective *weights*. y_1 is the neuron's output. The input to the neuron's activation function is z and is calculated as shown in Eq. 1.1:

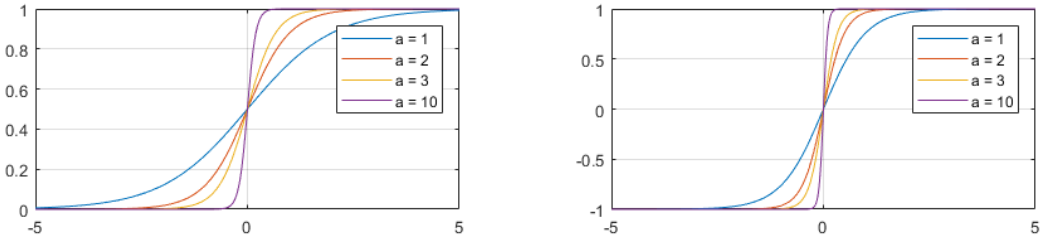
$$z = \sum [X_1 W_1 + X_2 W_2 + b] \tag{1.1}$$

Once z is calculated, it is fed to the neuron's activation function. There are several different activation functions, the most traditional ones are the *sigmoid* and tanh, as demonstrated in Eq. 1.2 and 1.3:

$$\sigma(x) = \frac{1}{1 + e^{-ax}} \ , \tag{1.2}$$

$$\tanh(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}} \ 1 \tag{1.3}$$

The advantage of using one of these functions is that they are capable of mapping an input that can have values from $[-\infty, \infty]$ to a range that varies from $[0, 1]$ or $[-1, 1]$ respectively as can be seen in Figs. 1.6a and 1.6b.

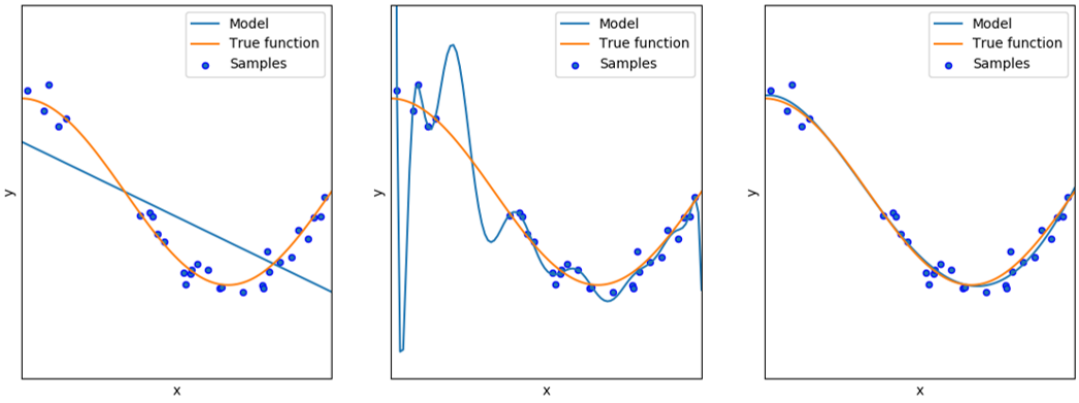


(a) Sigmoid function. (b) tanh function.

Figure 1.6: Traditional activation functions.

But for the ANN to learn, it has to be able to update it's internal *weights* and *bias* in order to best match the inputs with the outputs. This process is called training.

In the training phase, one of the most important things one should be aware of is when to stop it. If early stopped, an underfitting of the model Fig. 1.7a happens. This can happen for two reasons, the model had not enough time to train or the model is not complex enough to match the data. In Fig. 1.7b is represented the overfitting. It happens for reasons opposed to the underfitting, model too much complex or training time too long. It can be seen that the model memorizes all the data noise. In Fig. 1.7c is a training process stopped at the proper time. It does not oversimplify the data as in the underfitting nor it memorizes the data noise [3].



(a) Underfitting (b) Overfitting (c) Proper fitting

Figure 1.7: Examples of underfitting, overfitting and proper fitting.
Source: Adapted from [3].

The method used to track the training progress is to divide the dataset in at least 2. The training dataset and the test dataset. During the training phase, the cost function is checked for the train and test datasets. In Fig. 1.8 can be seen a traditional error curve of the training process. In red is represented the error of the training dataset and in blue the error of the test dataset. It can also be seen how after the point of lowest training dataset error (Proper fitting moment) it starts to increase while the error of the training dataset continues to drop and overfits the model.

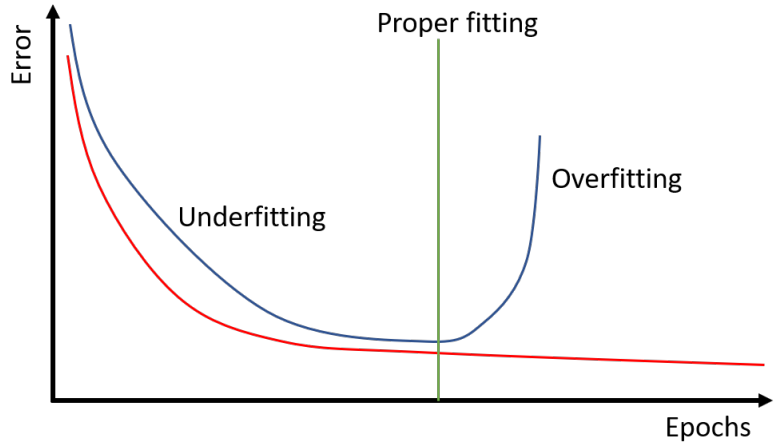


Figure 1.8: Common error curve.

To train the model, some method to compute the error and the update shall be chosen. There are several methods to do such a thing, the most common are the Stochastic Gradient Descendant (SGD) and Adam. They are focused on the Backpropagation [16] method to compute the *weights* and *bias* updates. The back-propagation is the act of giving an input to the network and evaluate its output with a cost function and then backpropagate the error information backward in the

network. This way the internal parameters can have their influence on the error individually evaluated and updated. The most common way to do this evaluation is by Gradient Descendant. It derives the cost function in function of the *weights* and *bias*. For that, it follows the chain rule and product rule in differential calculus:

$$\Delta W = -\alpha \frac{\partial F(W)}{\partial W} \ , \tag{1.4}$$

where ΔW is the weight update, $F(W)$ is the cost function and W is the *weight*. The network is trained in order to find the best possible value for the *weights* and *bias* so that the output is as close as possible to the desired target, and the cost function is minimized.

In [17] was proven that a traditional ANN with one hidden layer and using non-linear activation functions is capable to approximate any given function. This shows how robust this method is, when well developed and applied. Neural networks with their training methods and tasks are very versatile algorithms that can be understood as a canvas. Several machine learning methods based on this technique were created. Long-Short-Term-Memory Neural Networks, Extreme Learning Machine, and Deep Learning are some of the most used techniques when dealing with estimation.

1.6.1 Long-Short-Term-Memory (LSTM) Neural Networks

The LSTM, originally developed by [18], is an architecture based in gated Recurrent Neural Networks (RNN) [19]. It is a technique developed to be applied specially in long sequential data, gated RNNs are based on the idea of memory. LSTM creates a kind of path throw time where it can store its memory based on the arriving data sequence. For example, when a data arrives at the network, the internal gates autonomously chose to the best memory to feed from. The data tend to be processed by the most optimized block for that specific sequence. This architecture can be used for classification, regression, and forecasting tasks.

1.6.2 Extreme learning machine (ELM)

ELM is a supervised learning technique for classification, regression, clustering, data compression, and feature learning. This method consists of an ANN created with one or more hidden layers with random weights, but here they are not trained or changed. Just the output layer is trained and in most cases just a single step is necessary. According to this technique's creator [20], because of this, the ELM can learn a thousand times faster than traditional networks that use the backpropagation method to train.

1.6.3 Deep Learning

This method can be used for both supervised or unsupervised learning [21, 22]. It is an ANN with more than 2 hidden layers between the input and the output [22]. These extra layers enable a characteristic that can be seen as feature extraction from the previous layers requiring fewer neurons than a traditional shallow ANN. In Fig. 1.9 a traditional schematic model of this technique is shown, but it is worth mentioning that bein deep is the characteristic of having multilayers, so they can be, for example, an adaptation of the LSTM layer which gains the name of Stacked LSTM (S-LSTM), explored by [23], or even an ELM technique with multiple hidden layers.

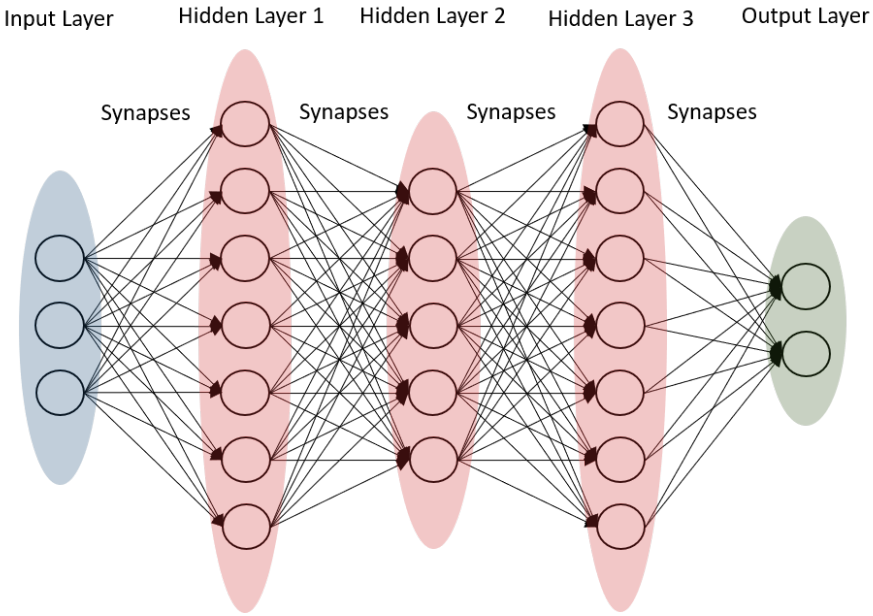


Figure 1.9: Traditional deep learning scheme.

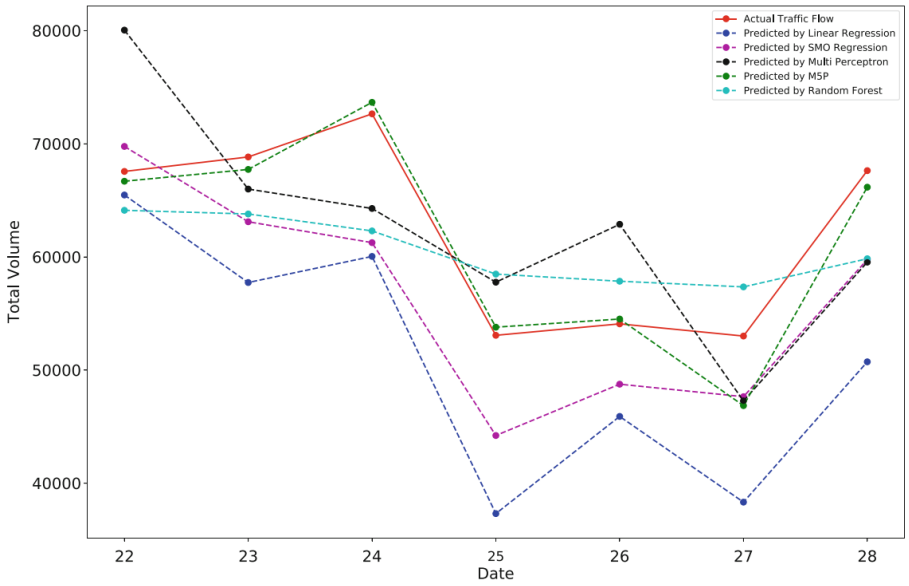
1.7 Bibliographic Review

The data from the drilling operation treated in this work are a time series. This means that the data points are sequential and indexed in order of observation. As researching for bibliography in the oil exploration industry results mostly in shallow articles that give a very brief explanation of the method used, this bibliographic review was subdivided in two subsections: Machine learning applied to sequential data, which provided richer articles and Machine learning applied to the fossil fuel exploration industry, which could show whether the industry actually sees values in machine learning techniques and is in deed exploration this field of approach.

1.7.1 Machine learning applied to sequential data

The main purpose of [4] was to apply machine learning to forecast the traffic flow in Porto, Portugal. For that, traffic data from the first tree weeks of each month were used and the algorithms had to predict the fourth week's traffic. The following machine learning methods were tested: Linear Regression, Sequential Minimal Optimization (SMO) regression; Multilayer Perceptron; M5Base Regression Tree; Random Forrest.

The traffic data were measured in 21 different positions in the roads of Porto. No details were given about the preprocessing of the data despite the exclusion of one of the 23 attributes because it was a constant measurement. In Fig. 1.10 it can be seen that the M5Base Regression method obtained the best results.



5

Figure 1.10: The prediction of traffic flow with different regression analysis. Source: [4].

In [5] a an online sequential extreme learning machine (OS-SLM) is proposed with one hidden layer to forecast the solar radiation. By OS is meant the capacity of the algorithm to read in real time the sequential data, make the necessary calculus, give the predicted solar radiation and continuously recalibrate it to be as accurate as possible in the course of time. In Fig. 1.11 a scheme of the algorithm is presented. The data history is given to the algorithm in the offline training phase, than it is tested and continuously appropriated during the online phase. The final result was very promising, gave results very similar to simpler SLM method but with the plus that it is online and continuously retrained.

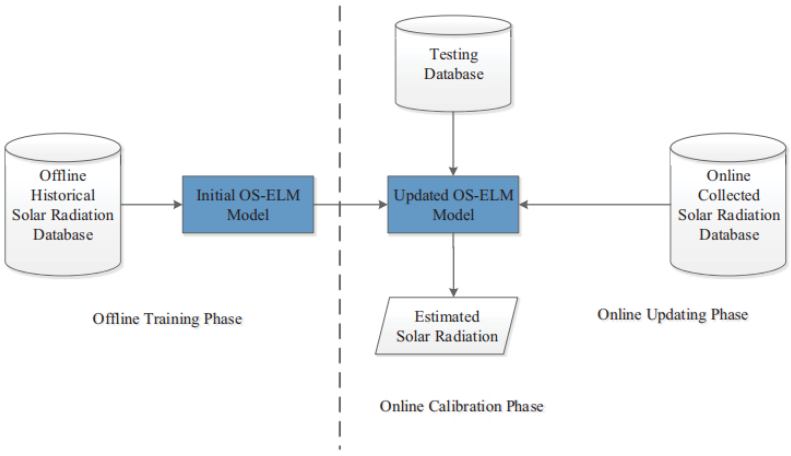


Figure 1.11: Main Framework of Real-Time Prediction of Solar Radiation.
Source: [5].

Despite the field of application, the task of this OS-SLM is very similar to the one proposed in this dissertation and the fact that the author obtained good results is a good indicator of the applicability of this technique. [24] also applied OS-ELM in sequential data for prediction task and obtained good results forecasting gas utilization ratio of blast furnaces.

A comparison between three NN based machine learning methods were made in [6] with the purpose to predict the water quality. The author compared the obtained forecast results of a simple ANN, LSTM NN and OS-ELM in three time steps, 3, 4 and 5 days ahead. In Fig. 1.12 can be seen that the LSTM technique obtained the lowest root-mean-square deviation (RMSE) of the predicted dissolved oxygen (DP) value been by this the most indicated technique for this specific problem.

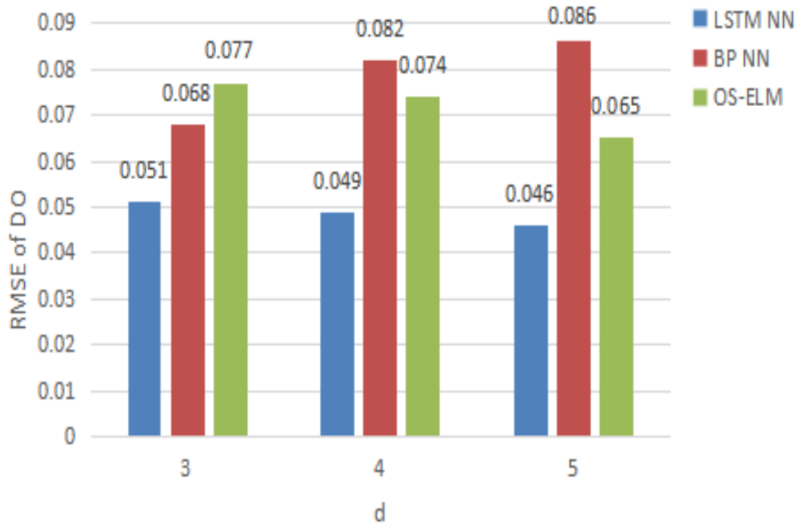


Figure 1.12: RMSE of DO with different time steps.
Source: [6].

An architecture based on S-LSTM layers was proposed by [7] in order to make

bearing fault diagnosis. The S-LSTM is a traditional LSTM but with multiple hidden layers. This gives the benefit of the deep neural networks associated with the recurrent networks of the LSTM. The idea behind this multi layering is that higher LSTM layers can capture abstract concepts in the sequences, which should help for the desired task. And in deed helped, the network obtained 99% accuracy, outperforming other techniques. The author outlined some important points inherited to this approach: No need for handcrafted features or advanced signal processing techniques which are essential for other method; the higher potential for mining inherent characteristics because of the deep layers; the higher computation cost for the training stage compared to other techniques; the dimension of the input is chosen by trial and error, no better method is found in literature.

In Fig. 1.13 is presented an accuracy comparison of the Stacked LSTM (Hierarchical LSTM/S-LSTM) proposed by [7] and some other ANN based techniques, traditional LSTM (1-layer LSTM), suport vector machine (SVM), backpropagation neural network (BP-NN) and convolutional neural network (CNN).

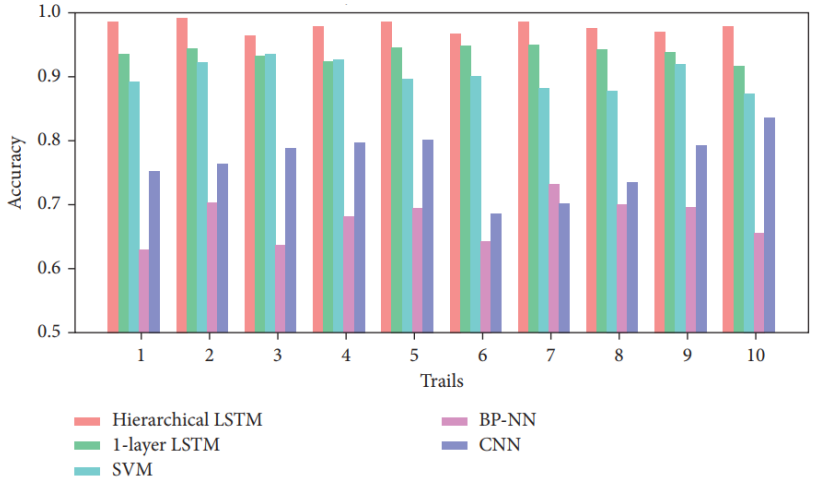


Figure 1.13: Accuracy of different methods in 10 trails.
Source [7]

An ensemble of extreme learning machine (Ens-ELM) was proposed by [25] in order to predict the daily wave height. The core characteristic of the ELM is the randomness of its internal parameters. Each of the ELM of the ensemble started with parameters in distinct regions, allowing with this a possible better generalization. The work compared Ens-ELM, simple ELM, OS-ELM and support vector regression (SVR) techniques and inferred that the Ens-ELM outperformed the other techniques for this application. It also gave better results than the ANN applied to the same problem by [26].

1.7.2 Machine learning applied to the fossil fuel exploration industry

In the drilling area there is still no reported use of machine learning techniques to estimate, in real time, with just surface data streamed with mud pulse, the torsional vibration severity factor. However, this section is going to explore the most important ones in the field.

Drilling dynamics and vibration

A drilling dynamics simulator was proposed by [27] where was utilized data from the surface and from downhole. This paper presented an approach based on Neural Networks (NN) to model the non-linear behavior of the multi-input/output drilling system for predictive control. In the author's opinion, the objective of demonstrating the method feasibility was achieved despite the fact that data from just one well was used.

In [28] was modeled the ROP using the vibration data from a fully automated laboratory drilling rig. A technique that combines neural networks with a sequential forward selection was used, which is a method that tests increasingly complex networks until it finds the best. The author concluded that utilizing the proposed model enhanced the quality and precision of the model.

Lithology estimation

In [29], field data from wired drill-strings were utilized. It compared different Support Vector Machine (SVM) techniques, one-versus-rest and one-versus-one, and a Random Forrest (RF) approach to classify in real time the logged lithology data. It succeeded and obtained great results, the non-linear nature of the problem was well dealt with the chosen techniques. Others have worked in the identification of reservoir lithology [30–34] and obtained good results.

Mud pulse data processing

A deep neural network strategy was implemented by [35] in order to process, in real-time, the low Signal do Noise Ratio (SNR) data transmitted by mud pulse received at the surface at a frequency of 0.5 Hz. Different recognition methods were tested with different SNR signal in this paper, the deep neural network obtained a final result ranging from 2 to 3% better than the traditional methods.

Drilling control

A neural network was used to improve the ROP by predicting and managing the drill-bit wear at [36]. It used data from other wells to, in real-time, optimize the drilling parameters that usually are controlled by the drill operator, like RPM, weight on bit (WOB) and mud pressure. With this method, the drilling control actions that the drilling operator take actions based on his personal experience and algorithm input

In [8] a combination of supervised and unsupervised techniques, PCA to compress and filter the data, K-means was utilized to create the operational clusters and the decision tree to choose the best way to leave the unstable zone. Pure surface data from 6 different wells were used to train the model. A drilling energy efficiency coefficient approach to infer the drilling conditions and choose the best ones was utilized. In Fig. 1.14 an overview of the proposed method is shown.

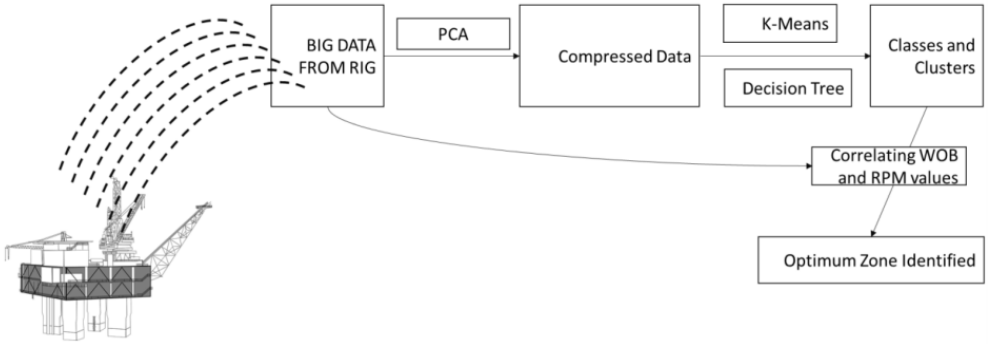


Figure 1.14: Scheme of the data processing tool created by [8] to optimize drilling.

1.7.3 Considerations

After a series of tests trying to process the data as a sequence, it was not obtained any significant results. After some reflection about this issue, the conclusion was that, despite the fact that the data being recorded in sequence, two facts disturb the neural network and they are:

- As will be shown in Chapter 2.3.2 in the creation of the indexes, once a window of data is given, the S_S information is there, not depending on the sequence itself.
- The acquisition rate of 5 seconds for the surface data is extremely low so when analyzed in sequence it becomes too noisy.

Therefore, even though most of the literature aims in interpreting sequential data by RNNs, in the case of this work it showed poor results. For this reason, the

RNN approach was abandoned. The pure maintenance of the data position in the input vector was enough for the interpretations.

Chapter 2

Data preparation

This work was made in partnership with Petrobras, therefore the company provided data from two ultra-deep wells located at a hydrographic basin in Brazil. This chapter aims to explain the nature of the data and the wells. The calculation of the S_S and other indexes with the intention to extract information of the data is made. For confidentiality purposes and to make it easier to follow, the wells and rock formations are going to be refered to as Well A, Well Br1, Well Br2, Rock A and Rock B.

2.1 Overview of the field drilling data

In these drilling operations, there were measurement tools in the surface, in the BHA and the drill-bit. For the equipment located downhole, there are two major ways to transmit the data to the surface. It can be through the utilization of wired drill-pipes which provides a very high transmission rate with a high signal to noise ratio, but unfortunately, this technology is very expensive and was not used in the wells treated in this work. The other main alternative is through the utilization of mud pulses, it is a restraint valve located at the BHA that restrains the passage of the drilling mud. This restriction leads to pressure variations noted in the surface by the drilling mud pump that are interpreted. The transmission rate of this technology is very low, bits per second, and it has a very low signal to noise raise ratio.

In Table 2.1 all the data that the surface has access to during the drilling are described. The measurements labeled as LAS-(name of the measurement) are done in the surface and saved at intervals of 5 seconds. The file index is Elapsed Time with steps of 5 seconds. TELE950-IWOB and ARC9 are measurement tools located downhole in the BHA. The data measured by them are transmitted through the mud pulse streaming method, and because of its limitations, the data from downhole arrives within a period of five minutes, one at a time, at the best scenario, and there are some moments where data simply do not arrive at all. Furthermore,

there is no information about the meaning of the following variables: DWOB_EU, DTOR_EU, CRPM, VIB_LAT, VIB_TOR, VIB_X. Because of these limitations, these data were not used in the analysis made in this work.

Table 2.1: Data Description.

NAME	UNIT	DATA SOURCE	DESCRIPTION
TIME	.s	:	Time (hh mm ss / dd-MMM-yyyy)
DEPTH	.m	DnMWorkflow	Depth Index
BIT_DEPTH	.m	DRILLING SURFACE	Bit Depth
COBTM	.	DRILLING SURFACE	Composite On Botton Status
AZIM_CONT	.deg	TELE950-IWOB	Continuous Hole Azumuth
INCL_CONT	.deg	TELE950-IWOB	Continuous Inclination (Hole Deviation)
HKLA	.1000 lbf	DRILLING SURFACE	Height of block above rig floor
SWOB	.1000 lbf	DRILLING SURFACE	Surface Weight On Bit
DWOB_EU	.1000 lbf	TELE950-IWOB	Uncorrected Downhole Weight on Bit
STOR	.1000 ft.lbf	DRILLING SURFACE	Surface Torque
DTOR_EU	.1000 ft.lbf	TELE950-IWOB	Uncorrected Downhole Torque
RPM	.c/min	DRILLING SURFACE	Rotational Speed
CRPM	.c/min	TELE950-IWOB	Collar Rotational Speed
TFLO	.gal/min	DRILLING SURFACE	Total flow rate of all active pumps
SPPA	.psi	DRILLING SURFACE	Standpipe Pressure
VIB_LAT	.gn	TELE950-IWOB	Transverse RMS Vibration
VIB_TOR	.1000 ft.lbf	TELE950-IWOB	Torsional RMS Vibration
VIB_X	.gn	TELE950-IWOB	RMS Vibration, X-Axis
GR_CAL	.gAPI	ARC9	Calibrated Gamma Ray
ROP	.m/h	DRILLING SURFACE	Rate of Penetration

At the end of the drilling, the drill-string is removed and the measured data can be saved directly from the memory of two other sensors located downhole that don‘t transmit anything to the surface. This generates two files and they are:

- **R5K:** This file is originated from the recorded memory of a measurement tool named BlackBox Plug (BBPLUG) located at the BHA. The only measurement this sensor makes is a radial acceleration at a frequency of 400 Hz. A moving average is calculated with a 2.56 second window and saved at 2.56 seconds.

- **R6K:** This file is originated from the recorded memory of a measurement tool named BlackBox HD (BBHD) located at the drill-bit. This sensor makes measurements at a frequency of 800 Hz and saves the data with a period of 3.2 seconds for Well A and Well Br1 and within 115 seconds for Well Br2. These wells are presented in the next section. The following variables are calculated in 3.2 second windows between recordings (or 115s in the case of the Well Br2): Min, Mean and Max RPM; Min, Mean and Max Lateral Vibration; Min,

Mean and Max Axial Vibration; Stick-Slip, Whirl and Vibration indicators. No information was provided about these indicators so they were ignored.

2.2 Introducing the wells

All the data treated in this work are relative to the 17,5” phase. This is because of the lack of quality of the data from other phases. An overview of the Well A, Well Br1 (well B run 1) and Well Br2 (well B run 2) main characteristics are at the table 2.2. Well Br1 and Br2 are actually the same well, but during the drilling in the run 1, the BHA broke. It was hooked and a similar BHA was used, but this new one had tools to make change the direction during drilling.

Table 2.2: Overview of the drilling of the Well A and B at the 17,5" phase.

Well A 17,5" phase - PDC + SKH616S			
Initial depth	3217 m	**Initial date**	11/04/2014
Final depth	5028 m	**Final date**	17/04/2014
Well type	Vertical	**Global ROP**	20.4 m/h
Formation 1 (3217 - 5020 m)	Ariri	**Rock type**	Halita/Anidrita/ Carnalita/Taquidrita
Formation 2 (5020 - 5028 m)	Barra Velha	**Rock type**	Calcário / Anidrita

Well Br1 17,5" phase - Xceed + SKH616M			
Initial depth	3078m	**Inicial date**	16/04/2013
Final depth	3300m	**Final date**	20/04/2013
Well type	Vertical	**Global ROP**	12.33 m/h
Formation	Ariri	**Rock type**	Halita/Anidrita/ Carnalita/Taquidrita

Well Br2 17,5" phase - Xceed + SKH616M			
Initial depth	3300 m	**Inicial date**	20/04/2013
Final depth	4263 m	**Final date**	04/05/2013
Well type	Build-up & tangent	**Global ROP**	11,40 m/hr
Formation	Ariri	**Rock type**	Halita/Anidrita/ Carnalita/Taquidrita

It can be noted that Ariri (Rock A) formation is presented in all of these drilling operations, but the Barra Velha (Rock B) is just presented on Well A. In Fig. 2.1

the transition from Rock A to Rock B at approximately 121.1 h can be seen. It is clear that the drilling scenario changed downhole.

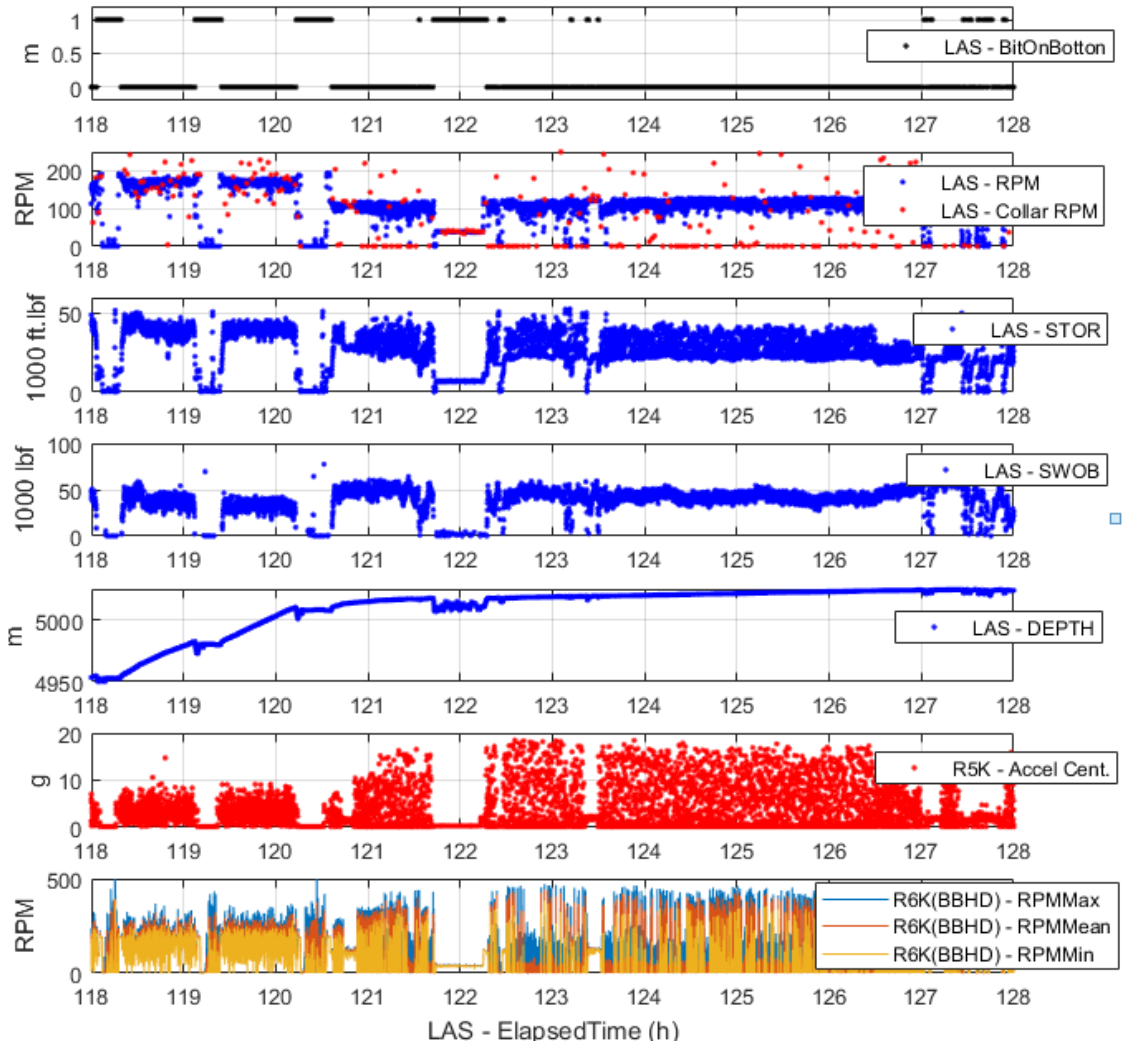


Figure 2.1: Transition from Rock A to Rock B at Well A.

In Fig. 2.1, the "LAS - BitOnBot" data are 0 when the drilling is happening and 1 when it's not. LAS - RPM is the RPM of the drill-string measured at the surface, LAS - Collar RPM is the RPM measured at the BHA, transmitted to the surface by mud pulse. LAS - STOR is the torque applied to the drill-string in the surface. LAS - SWOB is the weight on bit estimated at the surface, without taking in consideration external forces acting on the drill-string. LAS - DEPTH is the depth, measured at the surface, at which each measurement was made. R5K - Accell. Cent. is the centripetal acceleration measured at the BHA. R6K - RPM (Min, Mean and Max) are the RPMs measured at the bit.

2.3 Data processing

The drilling data provided by Petrobras are completely raw, this means that the entire operation, including all sorts of stops either for maintenance, for drill-pipe increments or even when the surface torque exceeds the top-driver maximum load and the operation has to stop are included in the data.

2.3.1 Data synchronization and cutting

In Fig. 2.2 a glimpse of the data can be seen. There are several measurements been made and saved with different sensors which are completely unsynchronized. The first main step was to correct it and make them all synchronized in order to make them all compatible and turn possible further processing.

One of the measurements is the Composite On Botton Status (COBTM), which represent if the drill-bit is touching the bottom (BitOnBotton), so there's a way to know if the drilling is happening or not. This BitOnBotton variable equals 1 if the drill-bit is not touching the bottom and equal 0 if it is.

The Fig. 2.2 shows the start of a drilling operation. It can be subdivided into seven parts represented in the figure as seven vertical lines that are better described next. At 1 all the drill-string is at rest, hanging, so there is no WOB applied. The top-driver start acting and all the drill-string gains rotational speed until the desired operational level. The Blackbox Plug starts measuring the centrifugal acceleration and the Blackbox HD starts measuring RPM. Between 2 and 3 there is the stabilization of the downhole rotational speeds. After 3, the drill-string goes down until it touches the bottom. At 4 it touches the bottom, the BitOnBotton measurement switches to 0. At 5, finally the WOB appears on the readings, the STOR increases because of the Bit-rock interaction and the downhole readings from the BlackBox Plug and HD start to oscillate more vigorously indicating the start of the bit-rock interaction. Between 6 and 7 the drilling operation actually started with the WOB and RPM stabilized on the desired operational level.

This step by step also occurs at the end of each drilling, with a similar error in the BitOnBotton variable. So, for safety purposes, all the data from the first 5 minutes and the last 5 minutes of drilling (BitOnBotton = 0) were ignored.

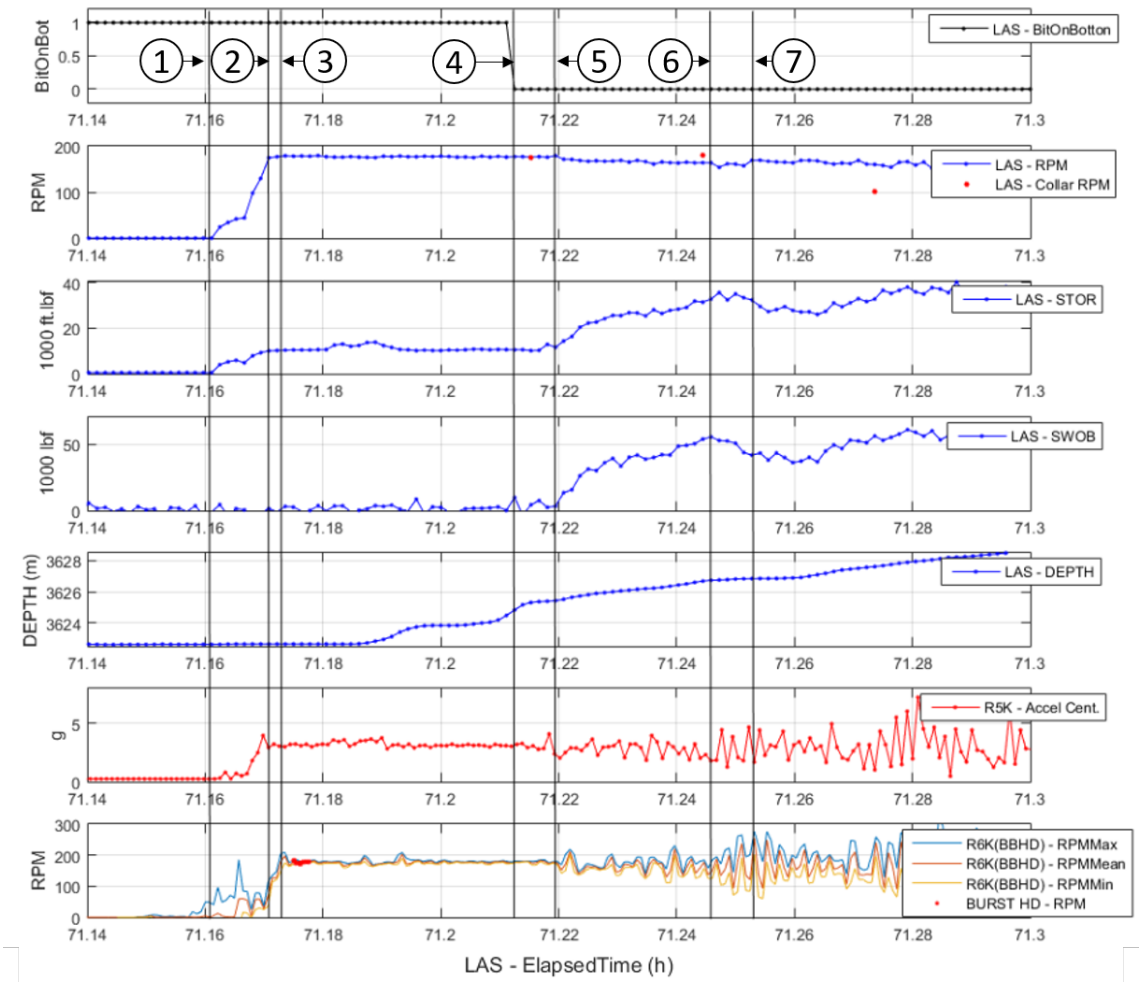


Figure 2.2: Drilling start.

In Fig. 2.3 a whole drilling interval is shown, from the beginning to the end. Some behavioral characteristics can be noted such as a direct correlation between a variation of the RPM on the bit and the centripetal acceleration on the BHA with the torque on the surface (LAS-STOR). It means that by increasing the torsional vibration at the bit a direct change in the torque measured in the surface is noted while almost nothing is noted in the surface RPM.

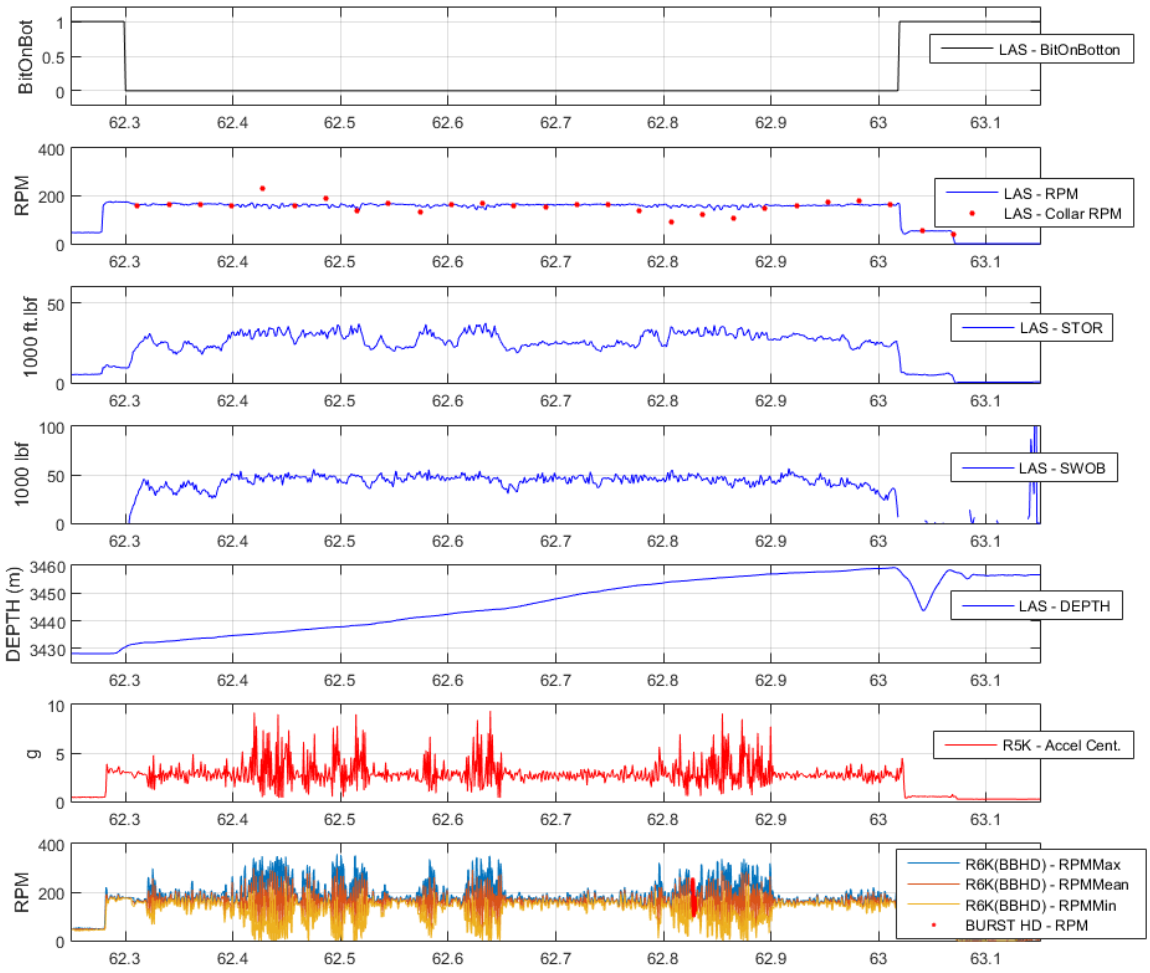


Figure 2.3: One drilling interval characterization.

2.3.2 Creation of indexes

Another important factor is that the data measured on the surface have a time index completely different from the downhole measurements. Because of this, as the focus of this work is the estimation of the torsional vibration intensity, the index was adapted. The industry uses the following expression to calculate this index:

$$S_S = \frac{\dot{\Theta}_{max} - \dot{\Theta}_{min}}{2\Omega} \ , \tag{2.1}$$

where $\dot{\Theta}_{max}$ is the maximum torsional speed at the bit and $\dot{\Theta}_{min}$ is the minimum torsional speed at the bit. Ω is the steady-state speed at the top-driver. In most models, it is considered as a constant value, but for this work, it is going to be the mean speed at the top-driver during operation.

The S_S is very important for the whole of this work. It is the attribute that will be estimated by the neural network. Because of that, a series of intervals were tested ranging from 15 seconds to minutes. The 30 second window represented very well the drilling operation so it was the chosen one. This means 6 measurements from the surface data (it has an acquisition period of 5 seconds), and 12 measurements for the downhole data (it has an acquisition period of 2.56 seconds). Therefore the S_S is calculated as Equation 2.2. This index for now on is going to be called as experimental S_S because it came from field-data.

$$S_S(\tau) = \frac{\max\{\dot{\Theta}(t)\} - \min\{\dot{\Theta}(t)\}}{2 \text{ mean}\{\Omega(t)\}}, t\epsilon[\tau_{-30segundos}, \tau] \ . \tag{2.2}$$

It was also observed that the surface torque increases its oscillation when the S_S increases. For this reason and better feed the ANN, a $STOR_{var}$ which measures the oscillation severity of the surface torque was also calculated.

$$STOR_{var}(\tau) = \frac{\max\{STOR(t)\} \ - \min\{STOR(t)\}}{2 \text{ mean}\{STOR(t)\}}, t\epsilon[\tau_{-30segundos}, \tau] \ . \tag{2.3}$$

ROP is another important factor in drilling. Even though it is calculated on the surface and should have a recording period of 5 seconds as the other data, it is saved within a period of 5 minutes. Therefore, it was also calculated. Simply deriving the $DEPTH$ in time results in very explosive values for the ROP. Because of that, a method similar do the calculation of the S_S and $STOR_{var}$ was adopted.

$$ROP(\tau) = \frac{\max\{DEPTH(t)\} \ - \min\{DEPTH(t)\}}{\max\{ElapsedTime(t)\} \ - \min\{ElapsedTime(t)\}}, t\epsilon[\tau_{-30segundos}, \tau] \ . \tag{2.4}$$

A window of the calculated values is demonstrated in Fig. 2.4. It is showing the rock formation transition moment, where the parameters of S_S, $STOR_{var}$, and ROP have a significant change.

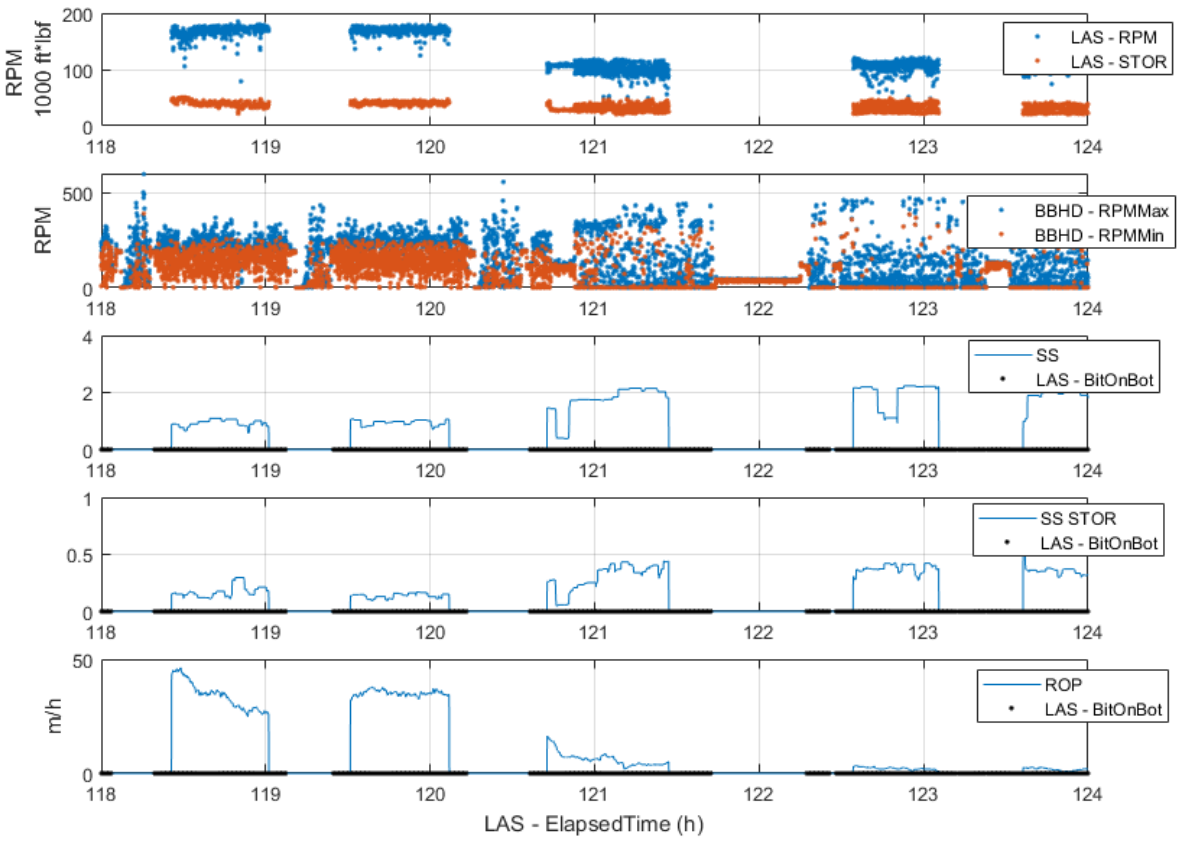


Figure 2.4: Calculated values of S_S, $STOR_{var}$ and ROP.

In Fig. 2.4, both the calculated S_S value, the $STOR_{var}$, and the ROP are somehow correlated. But, if carefully analyzed, it can be seen some discrepancies. In Fig. 2.5, at hour 118.8, for example, there is a peak in the $STOR_{var}$ and the ROP diminishes almost 50% while the S_S maintained the same level. Scenarios like this are common along the drilling data.

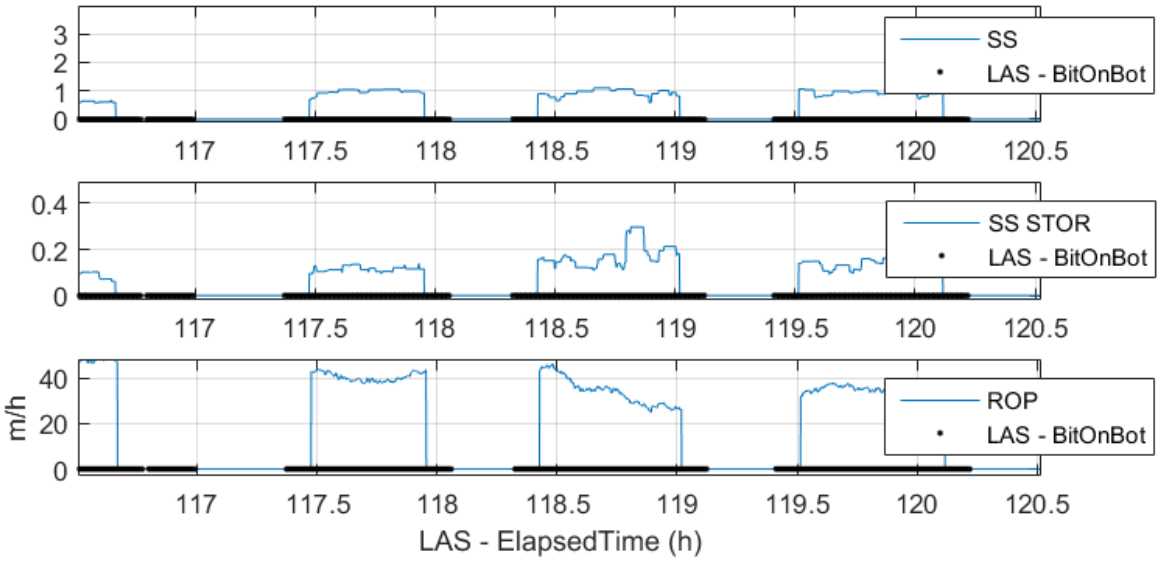


Figure 2.5: Calculated values of SS, $STOR_{var}$ and ROP.

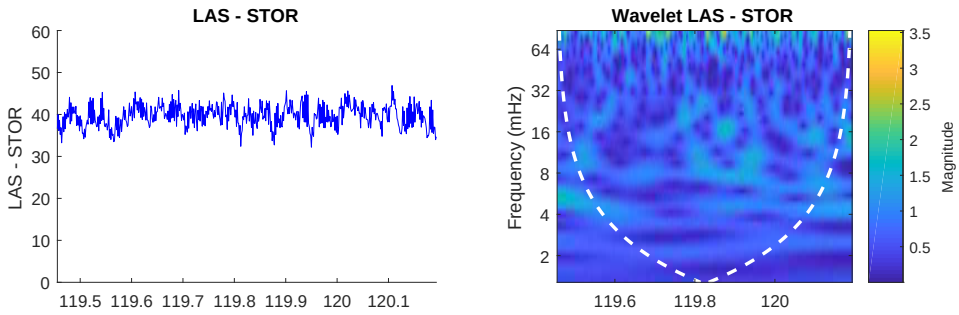
2.3.3 Analysis in frequency domain

Even though the acquisition frequency of the surface data being very low, 0.2Hz, in an attempt to better understand and extract information about the provided data some analyses were done in the frequency domain. The method that shown better results was the Discrete Wavelet Transform [37]. This transformation not even is able to represent the data in the frequency domain but also preserves the correlation with the time index. Because of this, its possible to make a direct comparison between the Wavelet Transform and the temporal series. The Generalized Morse Wavelet was used because of it's better equivalence between time domain and frequency [38–40] with the symmetry parameter of $y = 3$ and the product time-badwith $\beta = 60$, which are the default ones.

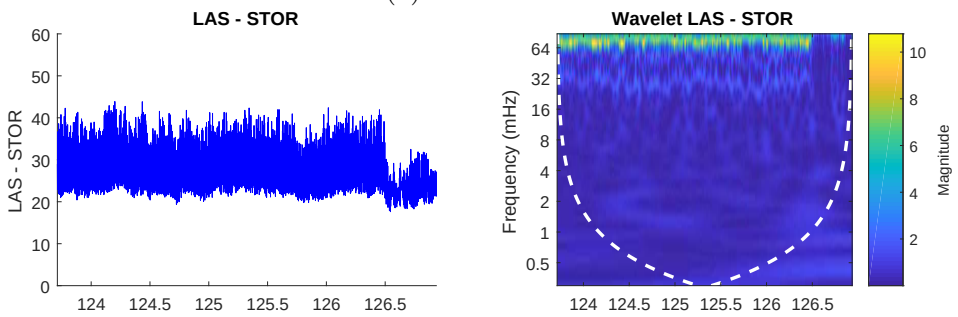
The analysis shown here is from Well A at three different time intervals. The Interval A belongs to Rock A and goes from $t = 119.4h$ to $t = 120.2h$. The Interval B belongs to Rock B, goes from $t = 123.7h$ to $t = 127h$ and was in a moment with less torsional vibration than the Interval C. Interval C goes from $t = 133.6h$ to $t = 135h$ and is from Rock B, having the highest torsional vibration. These three intervals were specially chosen from almost the same depth so there are as few as a possible changes in the drill-string length therefore in its mechanical properties.

Wavelets Transform of these three intervals from the surface torque and the RPM on the bit were made. It is worth remembering that the surface torque only has a frequency of acquisition of $0.2Hz$ while the RPM on the bit has a slightly higher frequency of $0.32Hz$. It means that both frequency scales analysis are very limited, but with the one of the RPM on the bit being a bit larger. The analysis cutting frequencies of the Wavelet Transform were of $0.2/2 = 0.1Hz$ or $100mHz$ for the surface data and $0.32/2 = 0.16Hz$ or $160mHz$ for the data from the bit. In Fig. 2.6 the Wavelet Transform from all the three intervals from the surface torque is presented and. in Fig. 2.7 is the Wavelet Transform of the RPM on the bit.

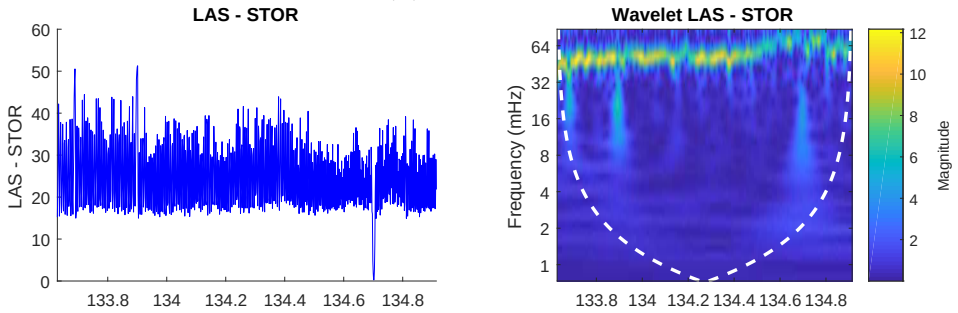
The white dashed line in all Wavelet Transform analysis made in this work denotes the corner of influence. Above this line the results are reliable.



(a) Interval A.



(b) Interval B.



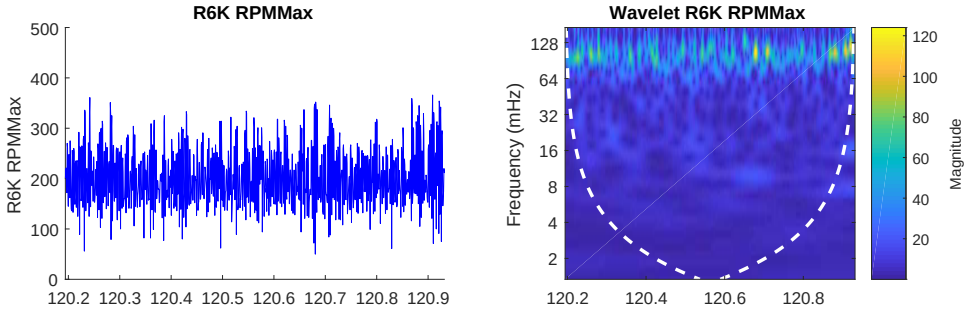
(c) Interval C.

Figure 2.6: Wavelet Transform of the surface torque at the intervals A, B and C.

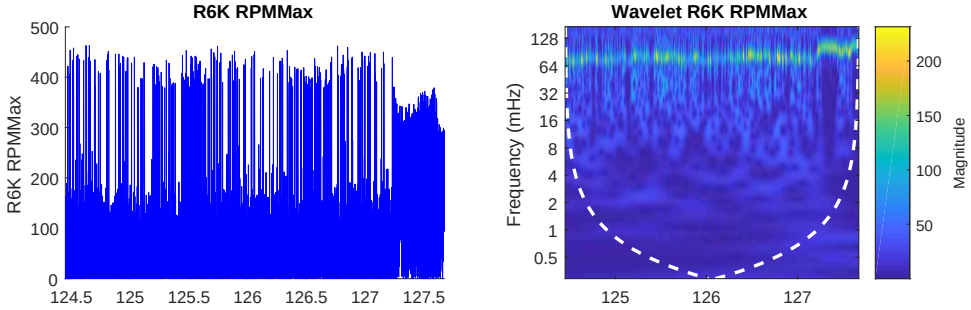
In Fig. 2.6 is presented a comparison of the behavior of the time series of the surface torque and its wavelet transform from Intervals A, B, and C. In Fig. 2.6a the surface torque's wavelet does not show any remarkable periodic behavior minding the limited frequency scale.

The Wavelet Transform of Interval B, Fig. 2.6b, presented an energy concentration at approximately $80mHz$. When comparing to the Wavelet Transform made on the same interval, but from the RPM on the bit, 2.7b, the same frequency appears but with a much higher magnitude, indicating that some of the dynamics of the torsional vibrations can be perceived at the surface.

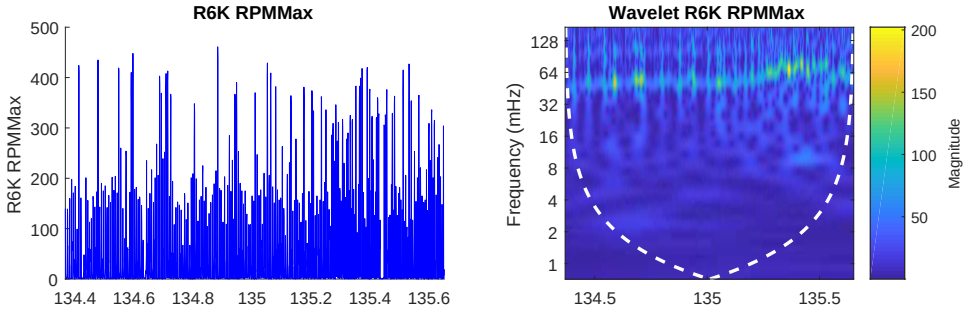
In interval C, Fig. 2.6c and 2.7c, it is even clearer the appearance of the torsional vibration on the Wavelet Transform, both on the surface and the data from the bit.



(a) Interval A.



(b) Interval B.



(c) Interval C.

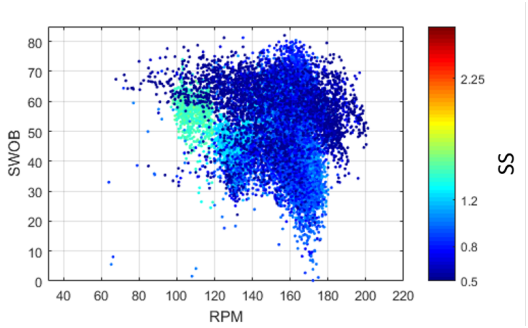
Figure 2.7: Wavelet Transform of the RPM on bit at the intervals A, B and C.

After comparing all the Wavelet Transforms made from the data coming both from the surface and from the bit, even with a very limited frequency range, it can be said that part of the dynamics occurring on the bit, regarding torsional vibrations, can be perceived at surface. Unfortunately, it is just true when dealing with very low-frequency dynamics because of the limited frequency analysis range.

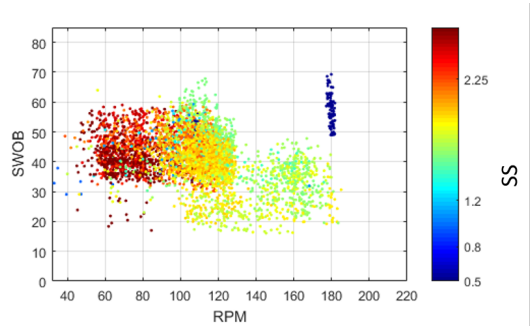
Because of this limitation, the use of the Wavelet Transform or FFT response as input to the DNN described in the further chapters were not considered. But it may be right to think that having a higher measurement frequency on the surface data would imply in more information on higher frequency torsional vibration dynamics, therefore, being an information-rich input to the DNN itself.

2.3.4 Torsional vibration severity map

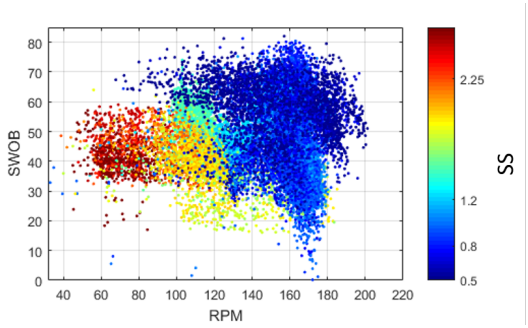
The industry default way of visualization of the drilling operation is, as shown in Fig. 2.8, the torsional vibration severity map. A map where the drilling RPM is the X-axis and the WOB is the Y-axis. The color of each point is the S_S intensity. One main limitation of this map is that it can just be used for the drilling visualization one rock at a time. When dealing with real drilling, the rock, despite still being in the same rock formation, changes a lot, so as the bit rock interaction. Hereafter an extrapolation of this map's functionality is made and different data from both Well A and Well B, both Rock A and Rock B are projected.



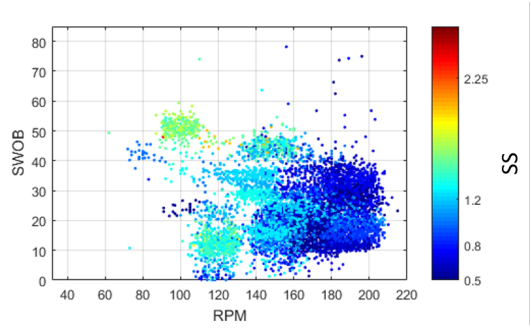
(a) Map of Well A formation A.



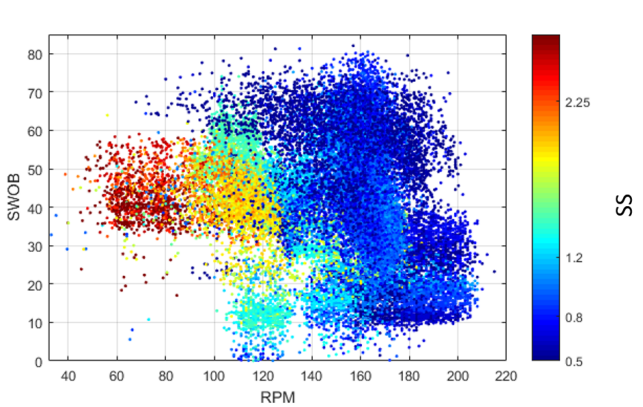
(b) Map of Well A formation B.



(c) Map of Well A formation A and B.



(d) Map of Well B run 1 and 2.



(e) Map of Well A and Well B.

Figure 2.8: Torsional vibration severity map.

In Fig. 2.8 the differentiation of formation or wells are very subtle, the clusters are overlapping and the stability map does not behave as expected [2, 41], with the

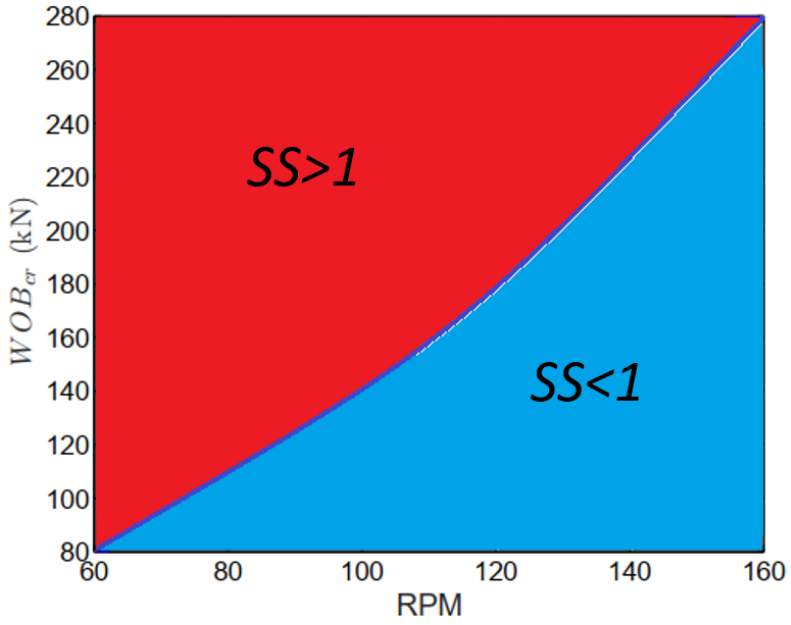


Figure 2.9: Torsional vibration severity map.
Adapted from [2]

S_S distribution as in Fig. 2.9. In this map, the values above the blue line are at the most intense type of torsional vibration, the Stick-Slip ($S_S >= 1$), the values below are with torsional vibration, but not with stick-slip and have $S_S < 1$.

Another problem with this visualization method is that other variables that are ignored and are also important to better understand the drilling operation scenario like the ROP for example are ignored.

Applying PCA

Principal component analysis (PCA) is a tool that makes a vector orthogonalization in a way that transforms a dataset with possibly correlated variables in a linearly non-correlated one. These variables are called principal components (PC) [42]. These PCs concentrates the information of the dataset in their first components. The main idea behind the application of this tool is to reduce the dimensionality of the problem without losing a significant amount of information. Possibly turning easier the training process for the future network.

This basis transformation consists of the creation the orthogonal basis in a way that the first component is in the direction that has the highest variance. The second component is in the second direction with a higher data variance and thus subsequently.

In the process of the PCA calculation an input matrix X^T that consists of n rows representing different measurements in time and m columns representing different types of measurements is used. This matrix is by default normalized and then a

singular value decomposition (SVD) of $X = W\Sigma V^T$ is made, where W is a square matrix of eigenvectors of de covariance matrix XX^T, Σ, is a diagonal, rectangular matrix where its diagonal numbers are the eigenvalues of XX^T and V is the matrix of eigenvalues of X^TX. The PCA transformation is given by:

$$\begin{aligned} Y^T &= X^TW \\ &= V\Sigma^TW^TW \\ &= V\Sigma^T \end{aligned} \tag{2.5}$$

By definition of the SVD, W is an orthogonal matrix, so each row of Y^T is a rotation of the respective row of X^T. This way, the first row of Y^T is of the *scores*, by that is meant the projection of X^T in the first principal component of W. The second row of Y^T is the *score* of the second principal component and thus subsequently.

The formation of the matrix X^T

All this work is meant to be used during the drilling operation, therefore, just surface data and the calculated $STOR_{var}$ and ROP were used in this calculation.

Between all the surface data, some considerations were made when choosing which one should be used. $SRPM$, $SWOB$ and $STOR$, were chosen because they are controlled parameters. $STOR$ is limited by the top-driver, $SRPM$ level and $SWOB$ are controlled in real-time by the operator. Beyond these variables, the calculated $STOR_{var}$ and the ROP were also used. With these variables was formed an input matrix X^T with 5 columns:

$$X^T = \begin{bmatrix} SRPM_{t1} & SWOB_{t1} & STOR_{t1} & ROP_{t1} & STOR_{var(t1)} \\ SRPM_{t2} & SWOB_{t2} & STOR_{t2} & ROP_{t2} & STOR_{var(t2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ SRPM_{tn} & SWOB_{tn} & STOR_{tn} & ROP_{tn} & STOR_{var(tn)} \end{bmatrix}$$

The use of other variables such as $SPPA$ and $TFLO$ were also studied, but they meant no significant alteration in the final result, minding the DNN output. Therefore they were not used for the PCA analysis.

Data normalization

The default normalization method used in PCA is as demonstrated in equation 2.6:

$$X_{norm} = (X - \text{mean}(X))/\text{std}(X) \ . \tag{2.6}$$

It removes the mean value of the distribution and divides it by the standard deviation. By doing this, the new distributions become zero-mean and with unitary standard deviation (std). Nevertheless, this work is idealized to be used during the drilling operation. Because of this, two problems arrive.

The first is that both the mean value and the standard deviation are going to change with time. Because of this, the two values would have to be recalculated with each new reading and a new PCA would have to be calculated which would change the already calculated principal components.

The second is that by removing the mean and standard deviation values of the distribution, part of the drilling characteristics is removed. For example, if a drilling operation had its RPM set at 150 RPM and a WOB of 50 klbf, it is an important information, therefore the normalization method could not ignore it.

Because of these two reasons and taking into account that the different measurements in the X^T matrix have values with different orders of magnitude, a normalization method that calculates de log of each value of X^T was chosen as in [43] and is demonstrated below:

$$X^T = \begin{bmatrix} \log \dfrac{SRPM_{t1}}{RPM_0} & \log \dfrac{SWOB_{t1}}{WOB_0} & \log \dfrac{STOR_{t1}}{TOR_0} & \log \dfrac{ROP_{t1}}{ROP_0} & \log STOR_{var(t1)} \\ \log \dfrac{SRPM_{t2}}{den} & \log \dfrac{SWOB_{t2}}{WOB_0} & \log \dfrac{STOR_{t2}}{TOR_0} & \log \dfrac{ROP_{t2}}{ROP_0} & \log STOR_{var(t2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \log \dfrac{SRPM_{tn}}{den} & \log \dfrac{SWOB_{tn}}{WOB_0} & \log \dfrac{STOR_{tn}}{TOR_0} & \log \dfrac{ROP_{tn}}{ROP_0} & \log STOR_{var(tn)} \end{bmatrix},$$

where $RPM_0 = 1rpm$, $WOB_0 = 1lbf$, $ROP_0 = 1m/h$ were used to adimensionalize the values before the log operation. $STOR_{var}$ already is admensionalized, so there is no necessity to make this operation.

Standard PCA

Despite the fact that the purpose of this work is to deal with the data during operation and, because of this, the principal components could not change over time. This fact makes the Standard PCA unusable in a real-time processing because at each new arrived data, new PCs would have to be calculated, therefore, new projections would be obtained. This makes the network always untrained for the arriving data. Even so, it was made to evaluate and compare the results with the Adapted PCA.

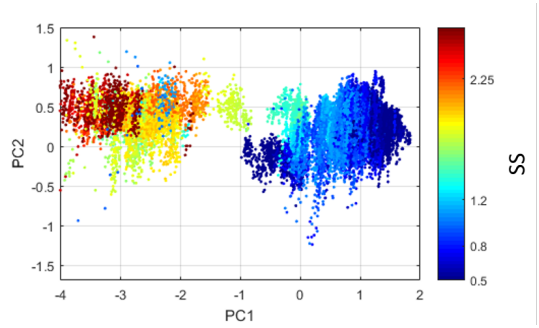
It was calculated the PCA over all matrix X^T, composed of all the data from Well A and B, both Rock A and B. At the end of the process, was obtained the W matrix of eigenvector below, where each column represents one principal component. The vector Ex is the representativity of each eigenvector respectively.

$$Ex = \begin{bmatrix} 73,1\% \\ 13,0\% \\ 10.1\% \\ 2.3\% \\ 1.5\% \end{bmatrix}, \qquad W = \begin{bmatrix} 0.1089 & -0.1234 & 0.0798 & 0.0001 & 0.9831 \\ 0.0856 & 0.8101 & -0.2829 & -0.4930 & 0.1153 \\ 0.1399 & 0.4751 & -0.0813 & 0.8634 & 0.0506 \\ 0.9162 & -0.0115 & 0.3646 & -0.1000 & -0.1325 \\ -0.3491 & 0.3204 & 0.8798 & -0.0374 & 0.0075 \end{bmatrix}.$$

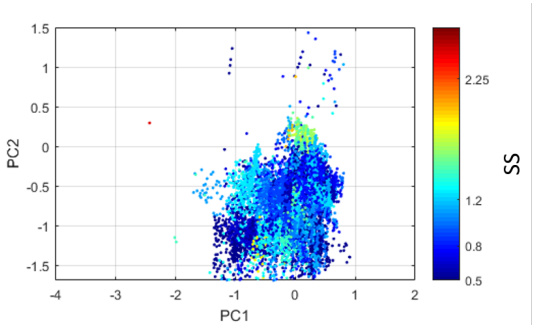
Once the PCs were obtained, a projection of X^T was made in these principal components as described in the following equations. The result of this projection is the Score vectors.

$$\begin{aligned} \Big[Score1\Big] &= \Big[X^T\Big]\Big[PC1\Big] \ , \\ \Big[Score2\Big] &= \Big[X^T\Big]\Big[PC2\Big] \ , \\ \Big[Score3\Big] &= \Big[X^T\Big]\Big[PC3\Big] \ , \\ \Big[Score4\Big] &= \Big[X^T\Big]\Big[PC4\Big] \ , \\ \Big[Score5\Big] &= \Big[X^T\Big]\Big[PC5\Big] \ . \end{aligned}$$

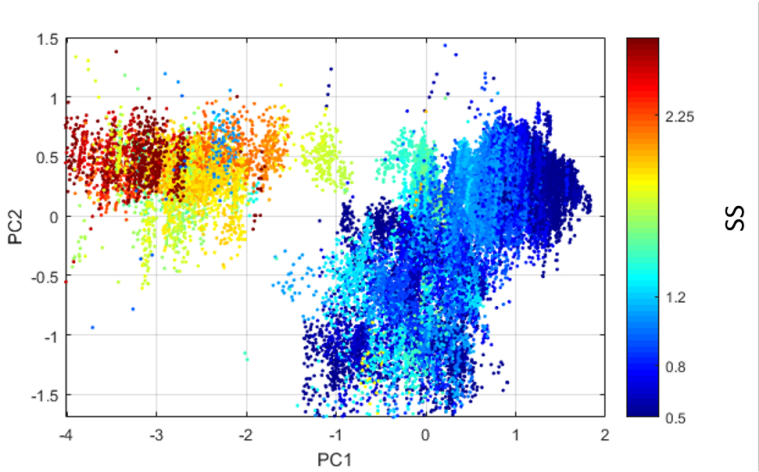
By joining all the 5 Score vectors obtained in the projections made, it is possible to obtain a new X^T matrix with it's data reorganized in a new linearly non-correlated basis. When plotting the first two columns (Score1 and Score2) in a scatter map. In Fig. 2.10 is the result of this map, where the color is the S_S intensity.



(a) Map of the Global PCA of the Well A.



(b) Map of the Global PCA of the Well B



(c) Map of the Global PCA of the Well A and B

Figure 2.10: Map of the Global PCA of the Well

In Fig. 2.10a, referring just to the drilling of the Well A, a strong separation between Rock A and B can be seen, where the cluster on the left and with the most intense S_S levels the Rock B and the cluster on the right, with the less intense S_S intensity the Rock B. In Fig. 2.10b that refers just to Well B can be seen just one big cluster that refers to Rock A. This last cluster is slightly shifted from the cluster from Well A Rock A but still partly superposed. By joining the projections from both Wells, in Fig. 2.10c, it is possible to note that differently from the default map used by industry, Fig. 2.8, this map separates rocks.

Adapted PCA

To make possible the online application of this tool, an adaptation of the PCA is proposed as in [43]. Was created an input matrix X^T that corresponds to a cluster of data of the optimal drilling scenario minding the S_S. For that, all the data from the Well A, when drilling the Rock A and had $S_S <= 1$ was chosen. With this matrix X^T of the optimal conditions created, the principal components were calculated. The main idea behind this method is to observe discrepancies in the drilling scenario.

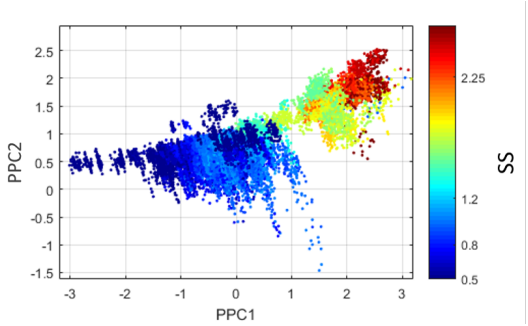
Once all the data of Well A when drilling Rock A and having $S_S < 1$ were extracted, the PCA was calculated. Below are shown the matrix W, where each column represents one principal component with its importance decreasing from left to right. The vector Ex comes by normalizing the Y^T in a way that the sum of its internal components equals one. The first term of Ex is the representativity of the first principal component ($PC1$), the second term represents the second principal components ($PC2$) and thus subsequently.

$$Ex = \begin{bmatrix} 57,8\% \\ 28,4\% \\ 10,4\% \\ 2,2\% \\ 1,2\% \end{bmatrix}, \qquad W = \begin{bmatrix} 0.0486 & 0.1674 & -0.0750 & 0.7590 & 0.6228 \\ -0.2564 & 0.2205 & 0.9392 & 0.0586 & 0.0024 \\ -0.2610 & 0.3600 & -0.1196 & -0.6064 & 0.6483 \\ -0.3160 & 0.7733 & -0.2780 & 0.1816 & -0.4380 \\ 0.8740 & 0.4425 & 0.1435 & -0.1405 & 0.0013 \end{bmatrix}.$$

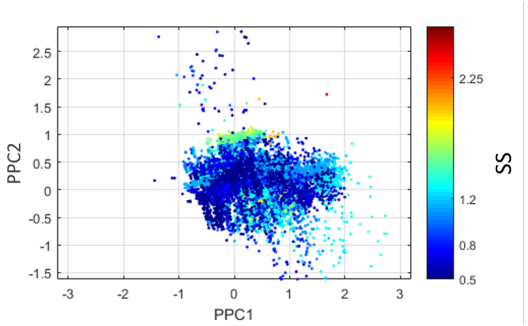
Once the PCs were obtained, a projection of all the log-normalized dataset (X^T) was made in the PCs. The $Score$ vectors was obtained as described below:

$$\begin{aligned} \Big[Score1\Big] &= \Big[X^T\Big]\Big[PC1\Big] \ , \\ \Big[Score2\Big] &= \Big[X^T\Big]\Big[PC2\Big] \ , \\ \Big[Score3\Big] &= \Big[X^T\Big]\Big[PC3\Big] \ , \\ \Big[Score4\Big] &= \Big[X^T\Big]\Big[PC4\Big] \ , \\ \Big[Score5\Big] &= \Big[X^T\Big]\Big[PC5\Big] \ . \end{aligned}$$

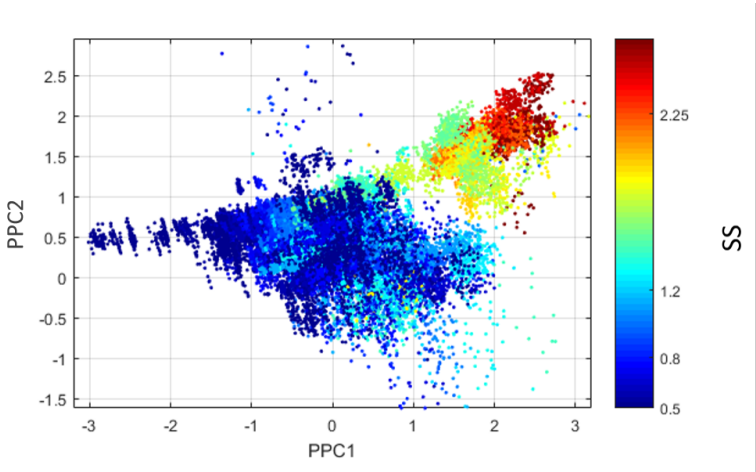
In Fig. 2.11 the result of this method can be seen when is made a graphic of $Score1$ x $Score2$. The color of the dots are the S_S calculated for each point with the color scale on the right of each figure.



(a) Map of the Adapted PCA of Well A.



(b) Map of the Adapted PCA of Well B.



(c) Map of the Adapted PCA of Wells A and B.

Figure 2.11: Adapted PCA map.

In Fig. 2.11a that refers just to Well A, it can be seen a separation between the formations A and B. The left cluster that predominantly is blue, with lower levels of S_S refers to the formation A and the cluster at the right, which is reddish and has higher levels of S_S, refers to formation B. Fig. 2.11b refers to the Well B, run 1 and run 2. It can be seen one major cluster and that makes sense since all this well had the same formation and same operational conditions. In Fig. 2.11c both Well A and Well B are represented and still a separation between formations.

This map, when compared to the original torsional vibration severity map used by the industry, Fig. 2.8, represents a better representativity of the drilling scenario either for separating the formations or for compacting the dimension of the problem. When compared with the map obtained by the Standard PCA, this adaptation shown poorer results separating the Rock formations. Which makes feasible to infer that when it is done a PCA with all the data (Standard PCA) more information is carried in less vectors. However, the results obtained by this adaptation are still promising. The application of the PCA can also be seen as feature extraction. In other words, it could make easier for the network to train and obtain an estimation as accurate as possible.

Chapter 3

Deep Neural Network Model

The problem this work handles is an evaluation of a sequential measured data in order to estimate the S_S during the drilling operation. Although it is a dynamic problem where the data is measured sequentially, the history of the states does not imply in any meaningful information about the current state that a window of data can't bring. In order to state such things, several tests with different types of recurrent neural networks were made, and the results shown were invariably worse than the ones obtained with a simpler DNN explained in this section, for reasons that also will be further discussed.

All the neural network variations used in this work were developed in Python because it is widely used in the machine learning field and have some powerful libraries such as Keras, Tensorflow, Scikit learn and Pythorch. Among all the possibilities, in this dissertation was used Keras because of its large community, extensive number of already implemented functions and GPU based processing. These characteristics allow more time to be spent on actually idealizing and creating the Neural Network itself.

3.1 The architecture

Following is the explanation of what was done in each part of the neural network used in this work so as its training process.

3.1.1 Input layer

The focus of this work is the estimation of the S_S. It, as demonstrated in Chapter 2, is an index that is calculated with data from a window of samples, therefore it is very important to give different time samples to the network be able to make the estimation.

Some tests were done with different sequence lengths that ranged from 1 (no

sequence, just one-time sample) to 50 samples. Anything beyond 6 samples shown no improvement in result. Therefore a sequence length of 6 was chosen as input to the network. This means that a S_S estimation to be made in t_0 it is necessary data measured at surface from $t_0, t_{-1}, t_{-2}, t_{-3}, t_{-4}$ and t_{-5}. As at each time instant are given to the model 5 variables and it needs them from 6-time steps, the neural network works with a total of 30 input variables in order to make one estimation of one value of S_S.

3.1.2 Hidden layers

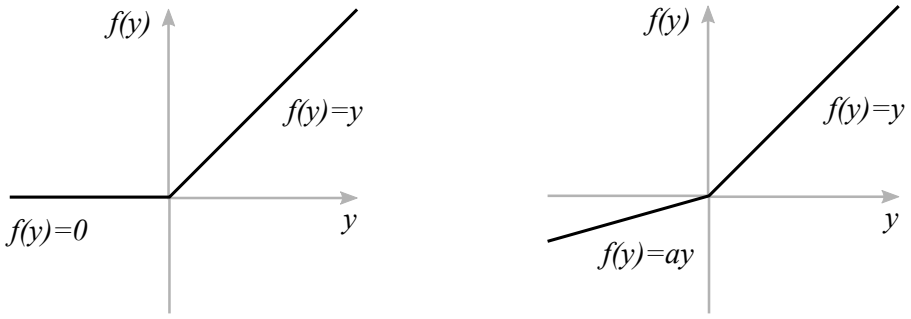
The number of neurons and the total of hidden layers were chosen within a method like the mesh convergence in finite element method. It was increased until the results stopped getting significantly better. The final layout of the neural network used were variable according to the dataset and will be explored in the Results in Chapter 4.

Although *sigmoid* and *tanh* were very used and discussed in the later 1990s and 2000s, the saturation is a great problem with both. As can be seen in Fig. 1.6 the saturation of large values at 1 and of small values at 0 or -1 for *sigmoid* and *tanh* respectively. And further, these functions are only really sensitive to changes around input zero [44].

Because of the characteristic described above, the vanishing gradient problem becomes a major one when dealing with deeper networks. The vanishing gradient occurs when the partial derivative of the error function with respect to the current weight in each iteration of training is very small, preventing the weight from updating. It happens most when dealing with activation functions such as *tanh* and *sigmoid* because they have gradients in the range (-1,1) and (0,1) respectively. Because of the chain rule, as the depth increases, the chance of vanishing the gradient increases. This phenomenon makes difficult to know which direction the weights should move in order to improve de cost function [44].

After around the 2010s, rectified linear activation unit (ReLU) started to be discussed. They show a great improvement in overall performance and permitted the development of very deep neural networks [44], page 226.

In 2015, a future derivation of the ReLU, the parametric ReLU (PReLU) was developed [45]. In Fig. 3.1 the shape of these activation functions is shown.



(a) ReLU. (b) PReLU.

Figure 3.1: ReLU vs PReLU.

Even though ReLU normally shows results better than the traditional non-linear activation functions when used in deeper neural networks, it still has a saturation below zero. The PReLU comes with the purpose to mitigate this. Its methodology was so successful when dealing with deep neural networks that for the first time, an artificial intelligence algorithm surpassed the human level on identifying and classifying images [45].

This new activation function works by learning the α parameter during training with the backpropagation method as well. This allows the activation function to continuously adapt to the *weights* and *bias*.

In this work a mix of PReLU and *tanh* functions was used. The fist layers were all of PReLU functions in order to prevent the vanishing gradient problem. The last layer was entirely of *tanh*.

3.1.3 Dropout layers

A deep neural network has a vast capacity to fit the training dataset, because of this, overfit becomes a serious problem [46]. In order to deal with such characteristic, each hidden layer was followed by a dropout layer [47] with a probability of 30%. The dropout layer is responsible to "turn off", in this case, 30%, of the neurons randomly and by doing this, changing the path of the internal operation. This technique was originally thought to be used just during training as a method to prevent the network to overfit and reducing the network generalization error [47, 48]. It is a powerful regularizing technique because all the neurons from each layer have to be equally important throughout all the possible networks [48].

In Fig. 3.2 a deep neural network with three hidden layers is represented. The red X on the neurons represents the ones that were randomly suppressed by the dropout layers.

This work used the dropout layer enabled in both training and testing phases as proposed in [49]. In [49] is mathematically proved that calling the networks several times with the dropout enabled is equivalent to Monte-Carlo sampling. Another way

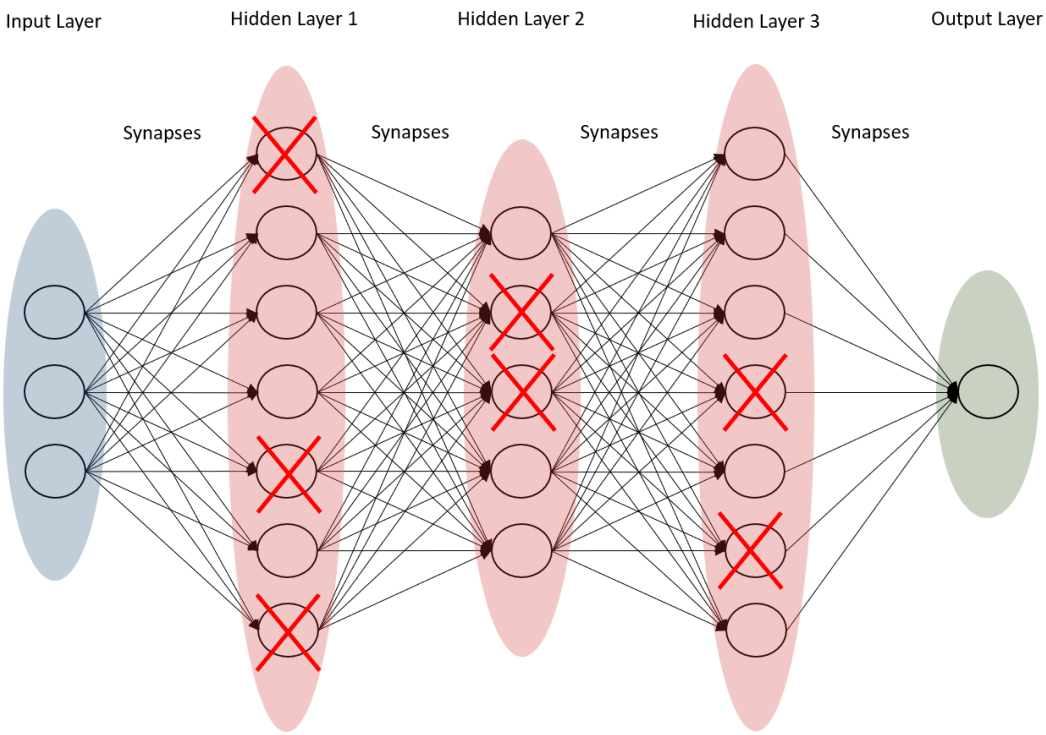


Figure 3.2: Example of Neural Network with dropout. Neurons randomly dropped.

to see this phenomenon is understanding the dropout layer as actually an ensemble of networks by considering that each time the network is called a different sort of neurons is used. Because of this, theoretically, the first and second moment (mean and variance) of the output provides the network's output and uncertainty respectively [49].

In the case of this work, evaluating the proposed neural network variability is a desirable feature because this method was developed to be used in during the operation to inform the drilling operator the downhole torsional vibration scenario so he can take the most worthy action. By evaluating the variability of the network‘s output, the driller could see if the outputs are with small or big fluctuations.

3.1.4 Output layer

Although the S_S value to be estimated is a continuous value, the *sigmoid* function, which is limited to a range of 0 an 1, was chosen in this layer. This was to choice in order to limit the output to a certain range, limiting the appearance of extreme values.

So the output layer consisted of a single neuron with a *sigmoid* activation function that mapped the multidimensional output from the non-linear previous layer to a single continuous output value that ranges from 0 to 1, matching with the normalized S_S.

The normalization of S_S was as follows. After analyzing all calculated S_S data from all possible wells (Well A, Well Br1 and Well Br2), it was noted that the value of 3.6 was the maximum.

After making a deep analisys in all the drilled lithology data and wells provided by Petrobras, a maximum value of $S_S = 4$ was entitled. Then all the calculated S_S values were divided by 4 so that they are invariably contained between 0 and 1 and the *sigmoid* function is then able to estimate its value. After the S_S value is estimated by the network, it is multiplied by 4 for viewing and tracking purposes.

The number of neurons in each layer so as the number of hidden layers were carefully chosen in a way that the best possible result was obtained while maintaining the network as simple as possible. A methodology similar to the mesh convergence in Finite Element Methods was used. It was more deeply explored in the Chapter 4 and Appendix A.

3.1.5 Weights initialization

It is essential the search for a good weight initialization to not let the network to reduce or magnify the layer input signals exponentially. With the traditional backpropagation [16] with random and uniformly distributed assigned *weights* between -0.3 and 0.3, it becomes increasingly difficult to give good results as the depth is increased because of the vanishing and exploding gradient problem. The last is exactly when the opposite of the vanishing problem occurs, it happens when dealing with activation functions that its derivative can take on larger values.

Therefore a method to choose the initial *weight* matrix has to be used. The *bias* matrices start with value zero for all its parameters as this does not imply in the previously described problem and shows good results. The chosen method to initialize de *weight* matrices was the one proposed by Kaiming He [45]. It consisted of a weight uniform distribution centered in zero with a limit of $[-limit, limit]$ where:

$$limit = \sqrt{\frac{6}{fan_{in}}}\ , \tag{3.1}$$

where fan_{in} is the number of input values in the input matrix. As discussed in [45], this relatively simple measurement considerably improved the overall performance of the network that used rectified linear units when compared with the standard random uniform distribution of [16] or [50]. When compared with other techniques like LeCun [51] or Orthogonal [52] initialization methods, the one proposed in [45] shown the best results. This technique was also applied for the initialization of the α values from the PReLU activation functions.

3.1.6 Loss function

To the optimizer optimize, it is necessary to have a cost function. The cost function used in this work was the mean square error (MSE) as in Eq. 3.2. This was chosen because among the traditional cost functions in literature this represented the one which gives the most importance to abrupt changes. As the problem this work deals with has severe abrupt differences in S_S estimation the MSE was the choice. Other cost functions like mean absolute error, mean squared logarithmic and the log(cosh) were tested by only decrement in the output accuracy were observed.

$$f = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i(\Theta))^2 \ , \tag{3.2}$$

where $\hat{y}(\Theta)$ is the output from the network regarding the respective *weights* and *bias* and y is the value the output should be.

3.1.7 Optimizer

The optimizer is the method used to update the *weight* and *bias* matrix. In this work was applied the Adam (adaptive moment estimation) [53], a first-order gradient method developed for efficient stochastic optimization. The Adam was designed in order to combine the advantages of other two optimization methods, the AdaGrad [54] and RMSProp [47] which scales updates similarly across batches. Below are the equations that show the internal process of the Adam optimization method.

$$r_t = (1 - \lambda_1)f'(\Theta_t) + \lambda_1 r_{t-1} \tag{3.3}$$

$$p_t = (1 - \lambda_2)f'(\Theta_t)^2 + \lambda_2 p_{t-1} \tag{3.4}$$

$$\hat{r}_t = \frac{r_t}{(1 - (1 - \lambda_1)^t)} \tag{3.5}$$

$$\hat{p}_t = \frac{p_t}{(1 - (1 - \lambda_2)^t)} \tag{3.6}$$

$$v_t = \beta\frac{\hat{r}_t}{\sqrt{\hat{p}_t}} \tag{3.7}$$

$$\Theta_{t+1} = \Theta_t - v_t \tag{3.8}$$

where, β, λ_1 and λ_2 are hyperparameters; Θ_t can be both the *weights* or the *bias*;

$f'(\Theta_t)$ is the cost function derived in function of Θ.

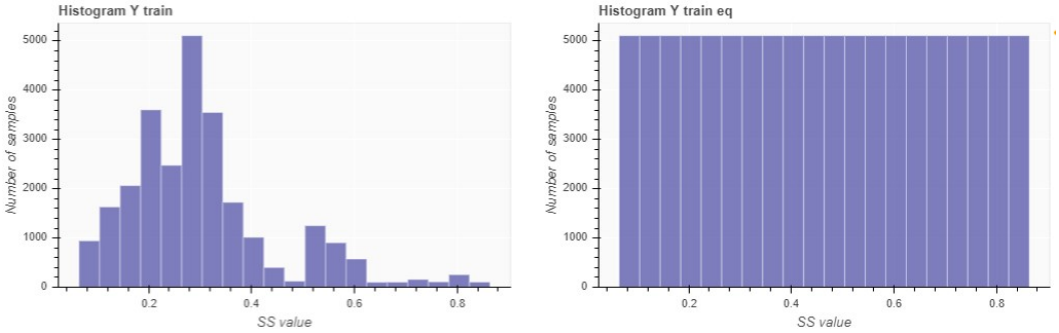
In Eq. 3.3 is the momentum like update, with the hyperparameter λ_1. In Eq. 3.4, is the RMSProp like update, with the hyperparameter λ_2. In the Eqs. 3.5 and 3.6 is represented their "corrections" by the time (t). Finally in Eq. 3.7 is the increment and in Eq. 3.8 is the *weights* and *bias* update.

Adam's biggest advantages are the following: the magnitudes of the parameters are invariant of rescaling of the gradient; the step size is strongly linked to the step size setting; the method doesn't require a stationary objective and, performs a form of step size annealing.

The default hyperparameters proposed in [53] are $\beta = 10^{-3}$, $\lambda_1 = 0.9$ and $\lambda_2 = 0.999$, and they had shown a good result. But for this problem in specific, decreasing the β to 10^{-4} significantly improved the network performance without increasing the training time. So $\beta = 10^{-4}$, $\lambda_1 = 0.9$ and $\lambda_2 = 0.999$ was used.

3.2 Training data equalization:

This step consisted of giving the same "importance" for all the S_S domain in the training phase. To better understand what it means, in Fig. 3.3a is a histogram of all the S_S values of the training data (explained in Chapter 4). It can be seen that the distribution is far from a uniform one. This implies that the values in the bins containing a higher population would train more times compared to the small ones. This step randomly replicates the data from the lower population bins until they are all the same size. Making the new distribution as uniform as possible. Some times there are bins with zero population, if that happens nothing is done at that determined bin.



(a) Original training data histogram. (b) Equalized training data histogram.

3.3 Batch size

Batch size is basically the number of training examples given to the network simultaneously in order to compute a mean error and with that a mean gradient in

order to compute the weights update. Determining the batch size of the training is an important task. Too small batch sizes lead to longer training time and may never let the network to converge because of the noisy updates, leading to the convergence to flat minimizers. This phenomenon also happens with large batch sizes [55]. Batches ranging from 32 to 512 samples tend to be a good starting point as defended in [56]. Despite [57] advocating on batches between 2-32, it showed very poor results in this problem, principally when dealing with the raw data, as will be explained in Chapter 4.

As could be seen by the previous paragraph, the batch size is not a consensus and may strongly depend on the data itself. Therefore, several different batch sizes were tried during the development of this work. Batch sizes ranging from 128 to 2048 samples got similar results. Because of the fact that by increasing batch size, due to the increased parallel processing, the computational cost got much smaller so was finally chosen a batch size of 2048 data points for the whole of this work.

Chapter 4

Results

In this Chapter the results obtained by the developed neural network are going to be discussed, so as its complexity and the benefits obtained by the data preprocessing proposed in Chapter 2.

The activation functions proposed in Chapter 3 were maintained so as the dropout layer and batch characteristics. Because of this, all the change in results obtained in this Chapter were due to increased neural network complexity and/or better data preprocessing characteristics.

The training process was carried until the MSE error from the test cluster consistently got worse or stopped showing improvement over the training. The best set of *weight* and *bias* was saved. This set of *weights* and *bias* received the name of *best model*. In other words, the best model is the set of *weights* and *bias* that combined with the chosen architecture gave the output that best matched, following the MSE criteria, the test cluster.

In Fig. 4.1 is a representation of the default error vs training epoch obtained thorough the tests made in this chapter. One training epoch consists of a cycle where all training data is provided to the network, error and gradients are calculated, and weights and bias updated. It can be seen that the error starts at very high levels and very fast arrives at low levels. After around the 50th epoch, the improvement increment at each epoch becomes very subtle. The approach was to train until epoch 150 and then pick the best model. With this model the output from both the training and testing datasets was plotted in order to compare them.

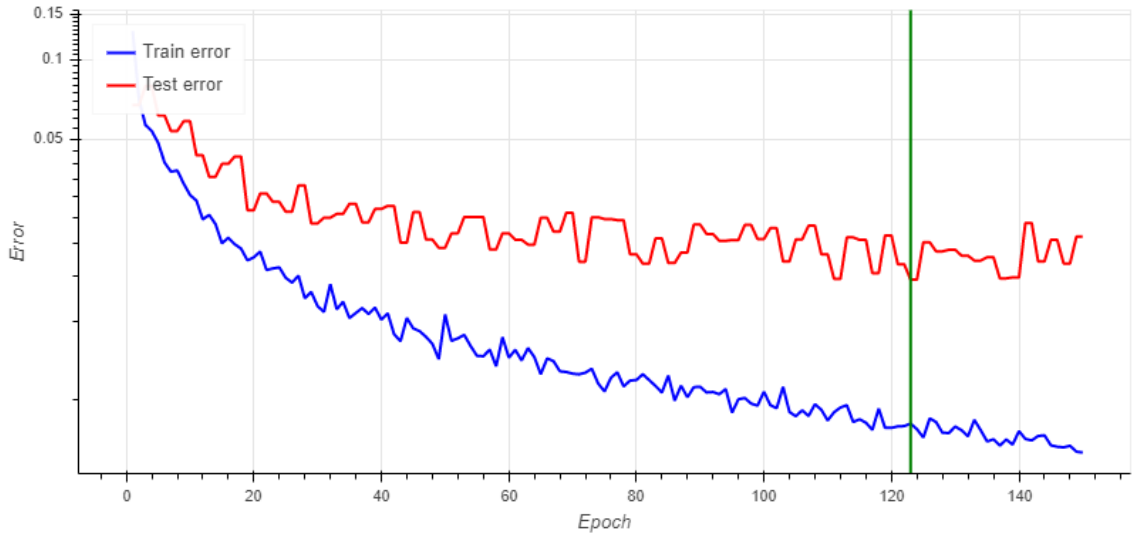


Figure 4.1: Default training error curve: train vs test error datasets.

In the figure above, the blue line represents the error from the training dataset and the red line from the test dataset. The green vertical line represents the moment where the minimum error from the test dataset was obtained, in this case at epoch 123. It is precisely from this moment which the best model is saved from. Subsequently, the output of the training and test data sets are plotted in order to compare them.

In Fig. 4.2 is all of the experimental S_S from Well A containing 25897 samples where 5489 are from Rock B, Well Br1 containing 5489 samples and Well Br2 containing 6960 samples. In the blue areas are the S_S for the Rock A and in the red area are the ones from Rock B. It can be seen with ease that when in Well A and happens the transition from Rock A to Rock B there is an abrupt change in the S_S. It also can be noted that the S_S for the Well Br2 has a different appearance, with the values being stagnated in some level for a long period of time. This happens because this run has an incredibly low acquisition period of 115 seconds versus 3.2 seconds from the Well A and Well Br1 from the downhole data. This Well was maintained in the test dataset in some tests in order to test how the model behaves when dealing with abnormal circumstances such as extreme low data acquisition rates.

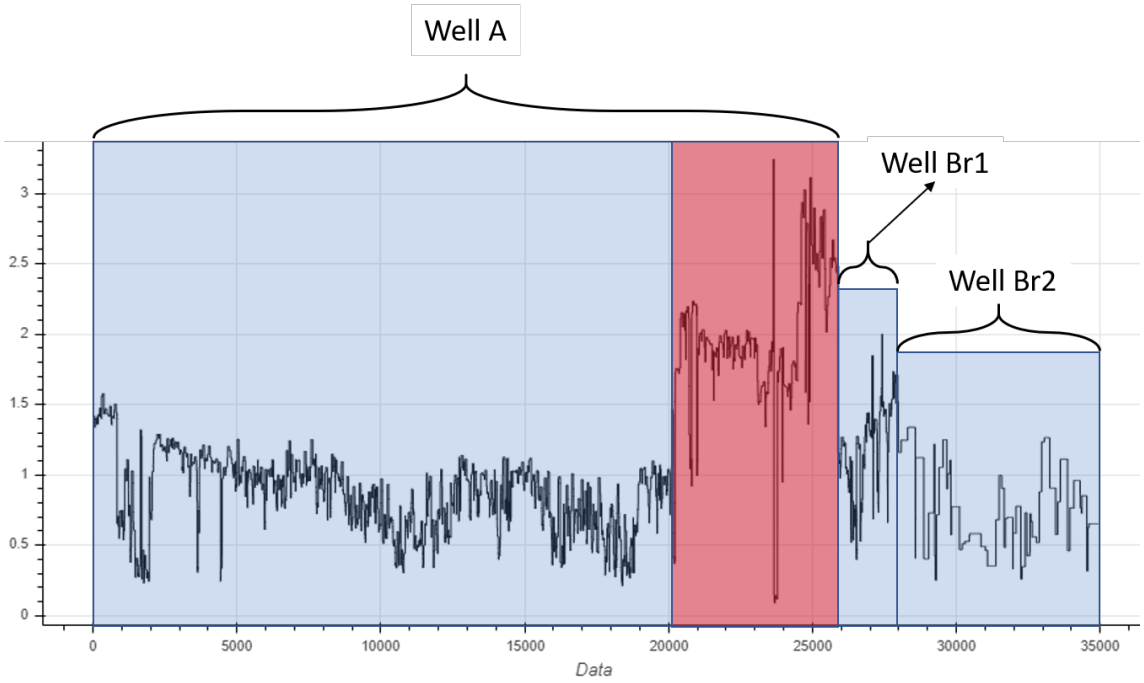


Figure 4.2: Calculated S_S for all the Wells.

In order to evaluate the ANN, the data were splited into two groups, train, and test with a proportion of 75% and 25% respectively. It was done to evaluate the capability of the ANN to estimate the value in different scenarios. Below are the explication of the four variant datasets created. It was not used the validation dataset because of the limited amount of data concerning different drilling scenarios.

1. **Case 1:** In this case, the training group was composed of 75% of data from Rock A, either from Well A and Well Br1, without Well Br2. The Test dataset was composed of 25% from data from Rock A and 100% from Rock B. The data for each cluster were randomly picked. These datasets were idealized in order to evaluate how the ANN behaves when a rock never saw before (Rock B) is presented. This evaluation should be an important indicator of how this network behaves when the drilling operation faces a completely new (to the network) lithology.

2. **Case 2:** In this case, the training group was composed of 75% of data from Rock A and B, either from Well A and Well Br1, without Well Br2. The Test dataset was composed of Rock A and B, 25% of the data from Well A and Br1 and nothing from Well Br2. The data for each cluster were randomly picked. These datasets were idealized to evaluate how the ANN behaves when it receives both Rock A and B. In other words, how it performs when it has to "take in consideration" the lithology to estimate the S_S.

3. **Case 3:** In this case, the training group was composed of 75% of data from

Rock A and B, either from Well A and Well Br1, without Well Br2. The Test dataset was composed of Rock A and B, 25% of the data from Well A and Br1 and 100% of the data from Well Br2. The data for each cluster were randomly picked. These datasets were idealized in order to evaluate how the ANN behaves when it receives both Rock A and B and data with different properties (Well Br2) and presented in the testing phase.

4. **Case 4:** In this case, the training group was composed of 75% all data, from all the Wells. The Test dataset was composed of the remaining 25% of the data from all Wells. These datasets were idealized to evaluate how the ANN behaves when it has data from all scenarios in the training phase.

For all the results shown below, two different neural networks were used. When dealing with the Raw data, a more robust one had to be used. It contained 5 hidden layers, with 400, 800, 800, 400 and 100 neurons respectively. Sequence length of 6 with 5 different measurements being a total of 30 inputs. When dealing with the pre-processed data (Adapted PCA), the network was simpler, with 5 hidden layers, with 200, 200, 200, 200 and 100 neurons respectively. The whole process to find the best architecture is described in Appendix A. These layouts were sufficient to the network converge its results to a scenario as better as possible. No noticeable gain was obtained by increasing the complexity of the network. As can be seen, after the Adapted PCA preprocessing, the network could became smaller.

Once the Neural Network's architecture was chosen, in Appendix B the variability of the output was evaluated. With this, a better understanding of the output's distribution could be acquired.

4.1 Case 1: Domain extrapolation at Rock B

Once the parameters described in Chapter 3 were chosen, the first step after becoming familiar with the set of programming tools used was to test the neural network with the raw data.

By raw data, it is meant the log of the data as it is. The measurements chosen to be used were the same used in the matrix X^T in Chapter 2 at the proposed PCA analysis, with the $SRPM$, $STOR$, $SWOB$, $STOR_{var}$, and the ROP. Was used the log of them because of the high discrepancy in the different data magnitudes, and as explained in Chapter 2, not to lose the information of the mean of the values.

In the Fig. 4.3 and at all similar ones (4.4, 4.6, 4.7, 4.9, 4.10, 4.12, 4.13), is the output from the best model trained. In black is the experimental S_S value, the "right" output. In blue is the mean output value obtained from giving the same input to the network 300 times. Because of the dropout layers, the output from

the network is stochastic. In orange is the mean value plus 2 RMS deviations and in green is the mean value subtracted from 2 RMS deviations. As the mean value, the RMS was calculated at each point, with 300 simulations as well. Therefore the interval comprehended between the orange and green lines is the interval with +/- 2 RMS deviations from the mean calculated for each input sample.

In Fig. 4.3 is shown the output from the network trained and tested with the Raw data and in Fig. 4.4 is the output from the network trained and tested with the data from the Adapted PCA. Both from the datasets from Case 1.

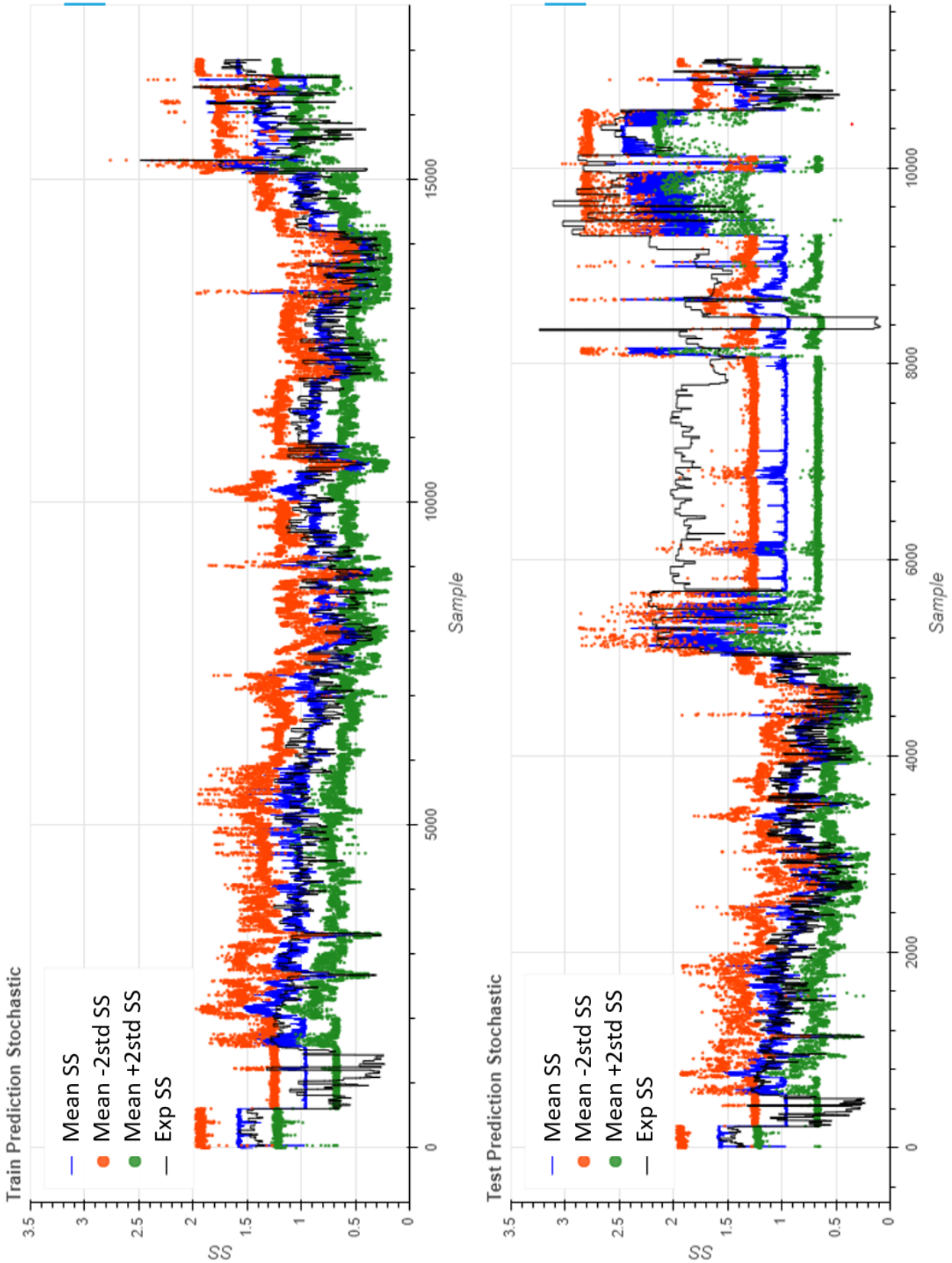


Figure 4.3: Case 1: Raw data.

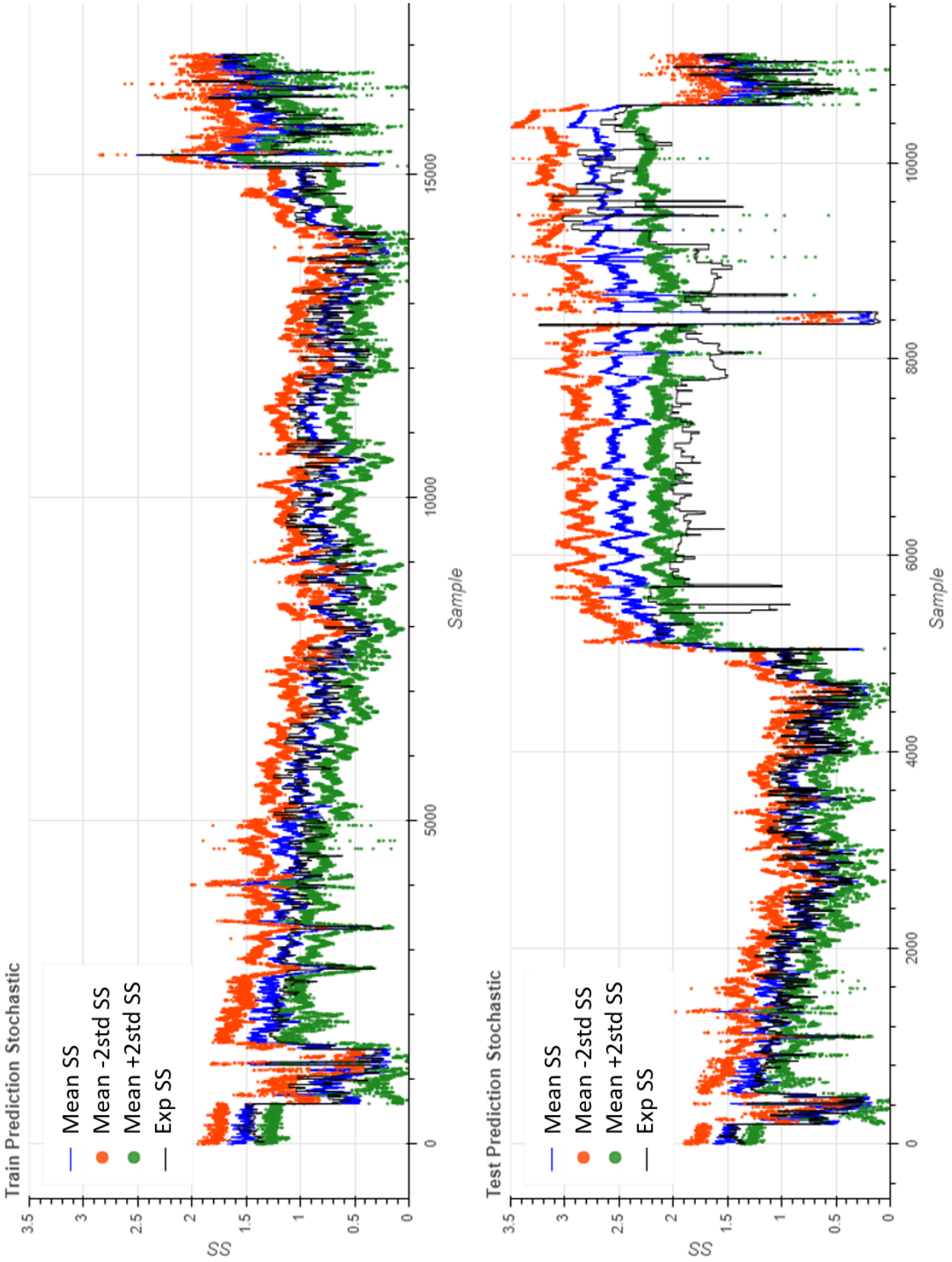


Figure 4.4: Case 1: Adapted PCA.

In Fig. 4.3 is clear that despite the fact that the network follows the right answer most of the time in the training phase (the left figure), the interval comprehended

between the +/- 2 RMS deviations is wide and saturated around 0.3. Leading to an output that brings little information to the drilling operator. When analyzing the output from the test phase (figure from the right), it continues with similar behavior as the training phase when tested with data from Rock A (the smaller S_S values comprehended between 0 to 4500 samples and from 10300 to 10600 samples).

When this network is presented with data from Rock B (values comprehended between 4500 and 10300), the network completely misses. It gives results that are misleading, with a very large RMS deviations and again with an almost constant estimation of values around 0.3.

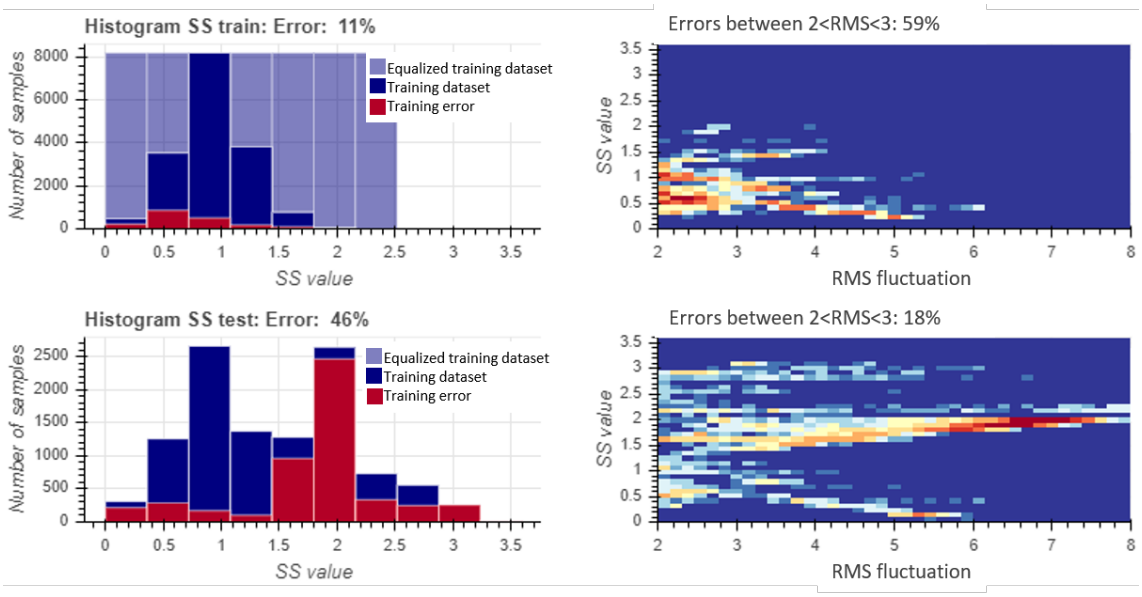
In Fig. 4.4 the results with both the train and test dataset get significantly improved over the Raw data. The RMS deviations is smaller, did not show regions with almost unchangeable estimations, and is overall more accurate. In the test phase, when dealing with the Rock B (not presented in the training phase), the network was able to acknowledge that something changed in the operation and the S_S value had increased. Although in Case 1 both networks completely mistaken the estimation of an unknown lithology, the Adapted PCA network could at least note that the S_S had increased which could be an important indicator for the drilling operator.

The Fig. 4.5a an approach to better comprehend and understand the training error was created. In the upper left, there are three histograms. In dark blue is the histogram from the S_S of the training dataset, in light blue is the histogram from the S_S of the equalized training dataset approached at Chapter 3 and in red is the histogram of the S_S estimations that fluctuated more than 2 RMS deviations from the correct value. This last histogram comprehends the errors obtained in the testing phase from the training dataset. In the lower-left 2 histograms are represented. In dark blue is the histogram of the S_S of the test dataset and in red is the histogram of the errors obtained in the testing phase with the test dataset. In the upper right is a 2D histogram where the x-axis is the fluctuation in terms of RMS deviations between the experimental S_S value and the calculated mean from the network. The y-axis is the "right" S_S value. This histogram just shows fluctuations above 2 RMS deviations because are the ones comprehended in the red histogram of the upper left. Its title brings the information of how much of those mistaken values (larger than 2 RMS deviations) were smaller than 3. The lower right figure is a 2D histogram identical to the upper one, but this one brings information about the test dataset.

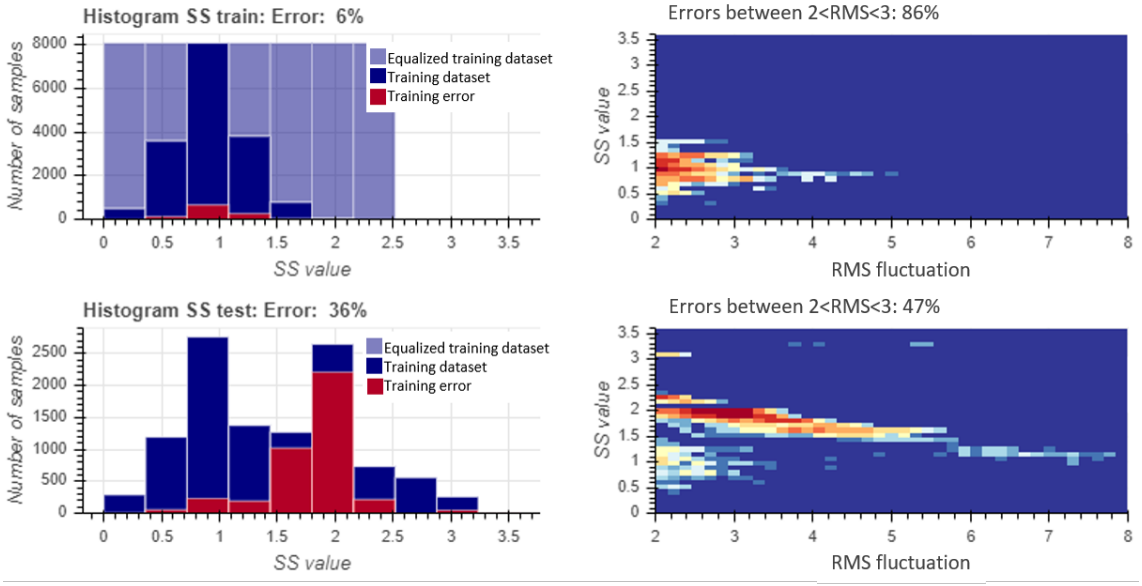
When comparing the histograms from the training and testing dataset in Figs 4.5a and 4.5b respectively, the improvement from the Raw data to the Adapted PCA data is made clear. The error rate went from 11% to 6% with the Adapted PCA data in the training dataset and from 46% to 36% in the testing dataset. But despite this, it is notorious how, in both cases, the vast amount of the mistakes

happen in the new S_S range. Although the S_S in the range from 1.8 to 2.5 could be equalized, there were very few data from these magnitudes. Because of this, very few data were replicated giving little information to the network. As this region also comprehends most of the Rock B, it is expected a poor result.

Comparing the 2D histogram of both cases, it can be seen that during the training phase the error behavior was similar, but in the testing phase the scenario was different. The network trained with the raw data had mistakes that fluctuated a lot from the estimated mean, while the network trained with the data treated with the Adapted PCA had its error much more concentrated in the range between $RMS = 2$ and $S_S = 4$ which is much better than a mean of almost 7 as in the network trained with the raw data.



(a) Case 1 error evaluation: Raw data.



(b) Case 1 error evaluation: Adapted PCA.

Figure 4.5: Case 1 error evaluation.

4.2 Case 2: No domain extrapolation, not using Well Br2

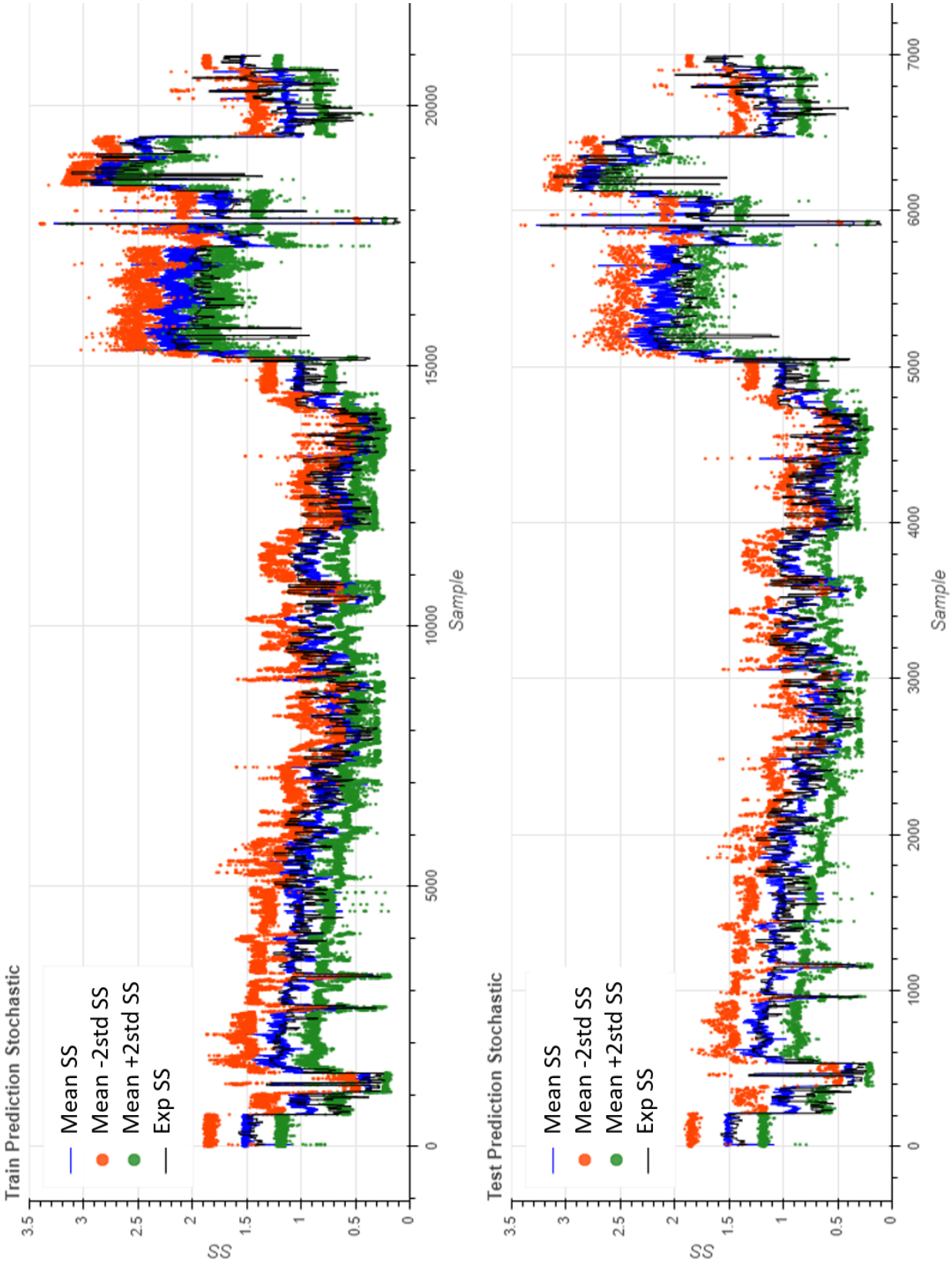


Figure 4.6: Case 2: Raw data.

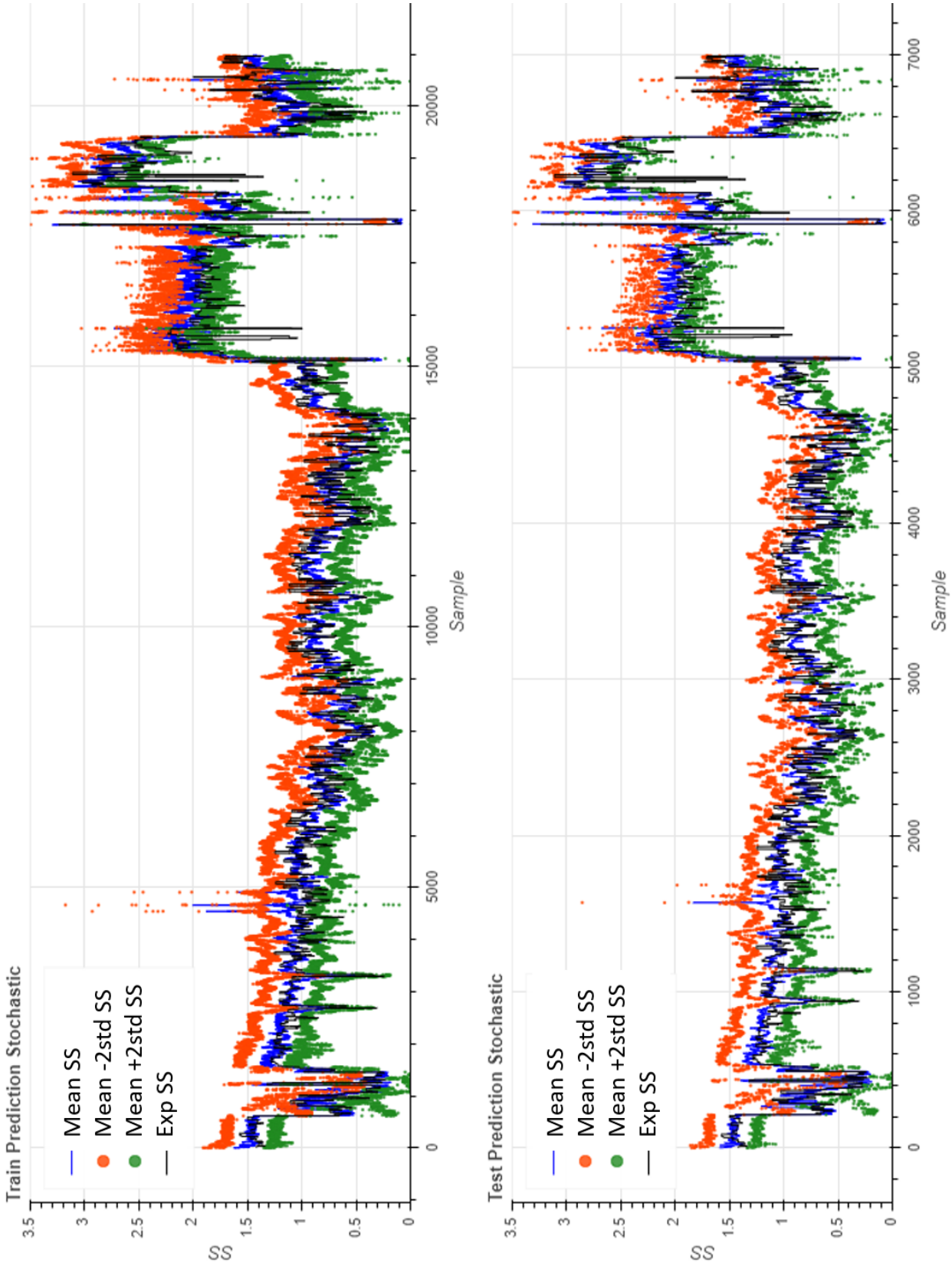


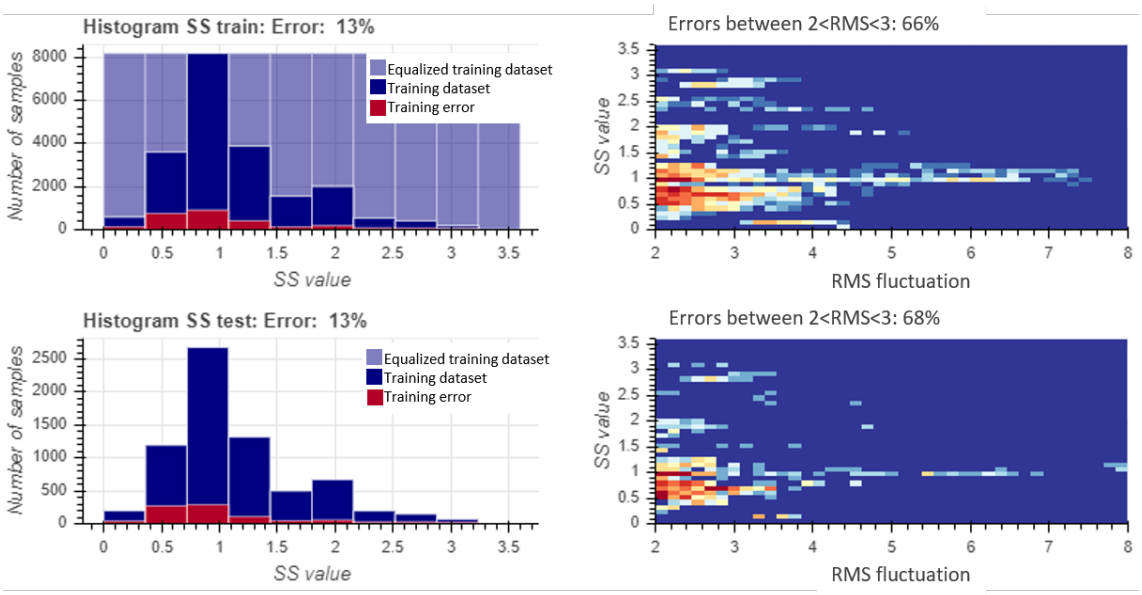
Figure 4.7: Case 2: Adapted PCA.

In Figs. 4.6 and 4.7 are represented the S_S estimation from the Case 2 from the Raw data and the Adapted PCA data respectively.

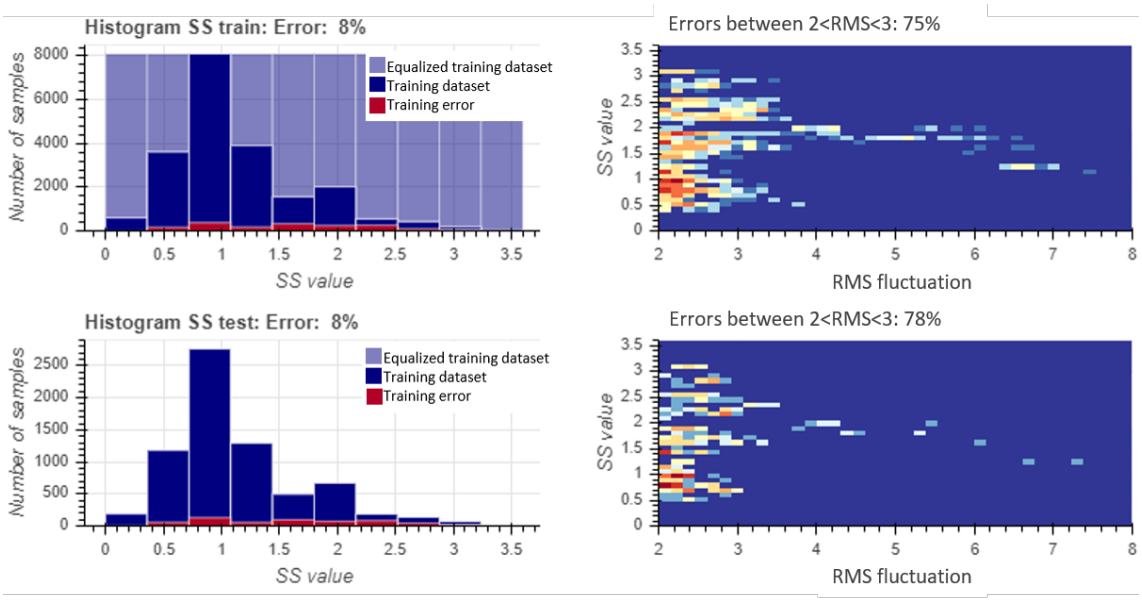
This case deals with the same scenario from both the training and testing phase. It simulates a real approach where the drilling is happening in previously drilled scenario. It perceives with ease the change of lithology so as subtle S_S variations.

When comparing the output from both the Raw input, Fig. 4.6, with the Adapted PCA, Fig. 4.7, the difference is subtle, but present. The error went from 13% in the raw data to 8% in the network trained with the Adapted PCA.

When comparing the 2D histograms, the ones from the Adapted PCA data shown a slightly higher concentration in values within a range of 3 RMS deviations.



(a) Case 2 error evaluation: Raw data.



(b) Case 2 error evaluation: Adapted PCA.

Figure 4.8: Case 2 error evaluation.

4.3 Case 3: Domain extrapolation at Well Br2

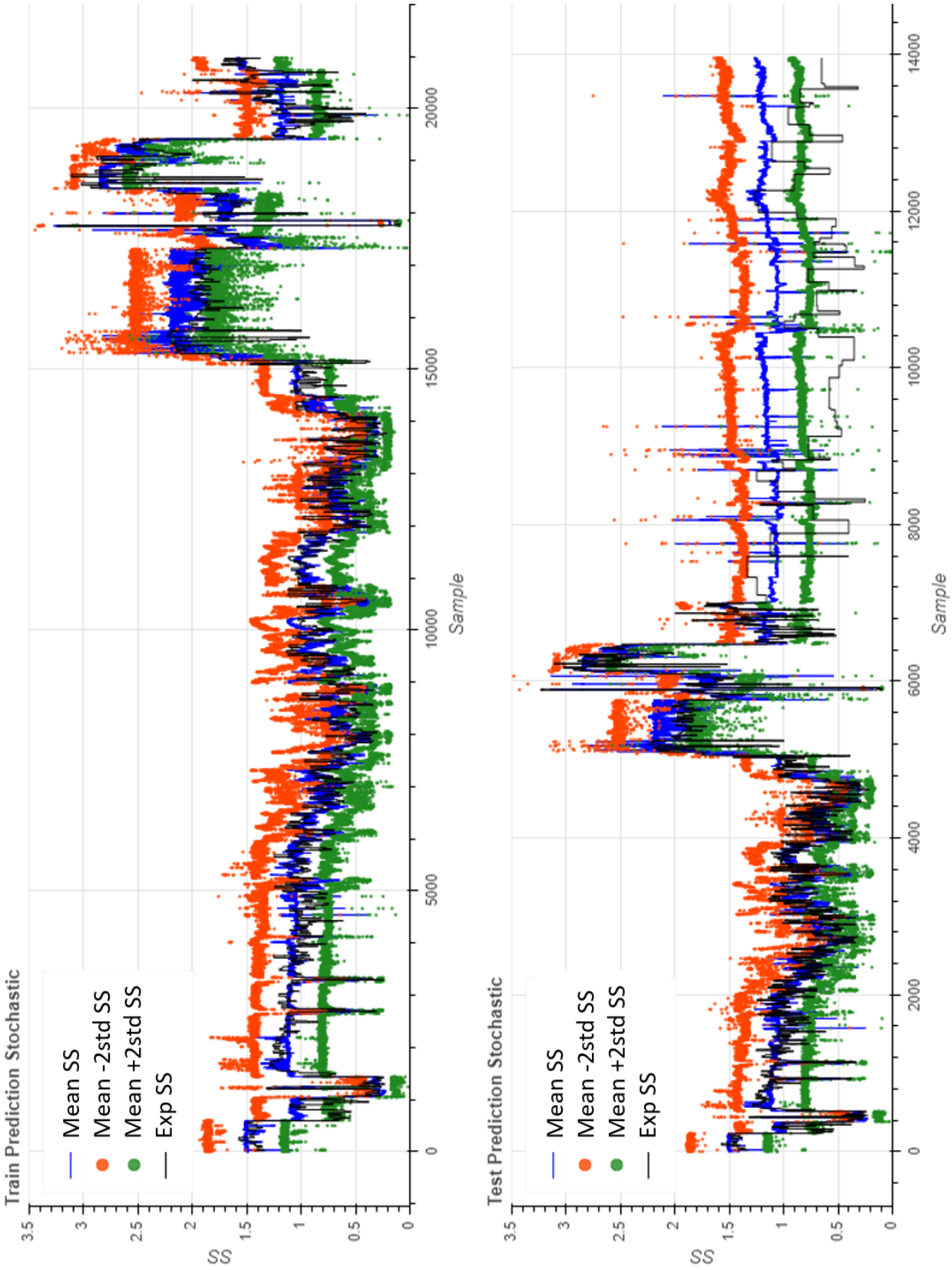


Figure 4.9: Case 3: Raw data.

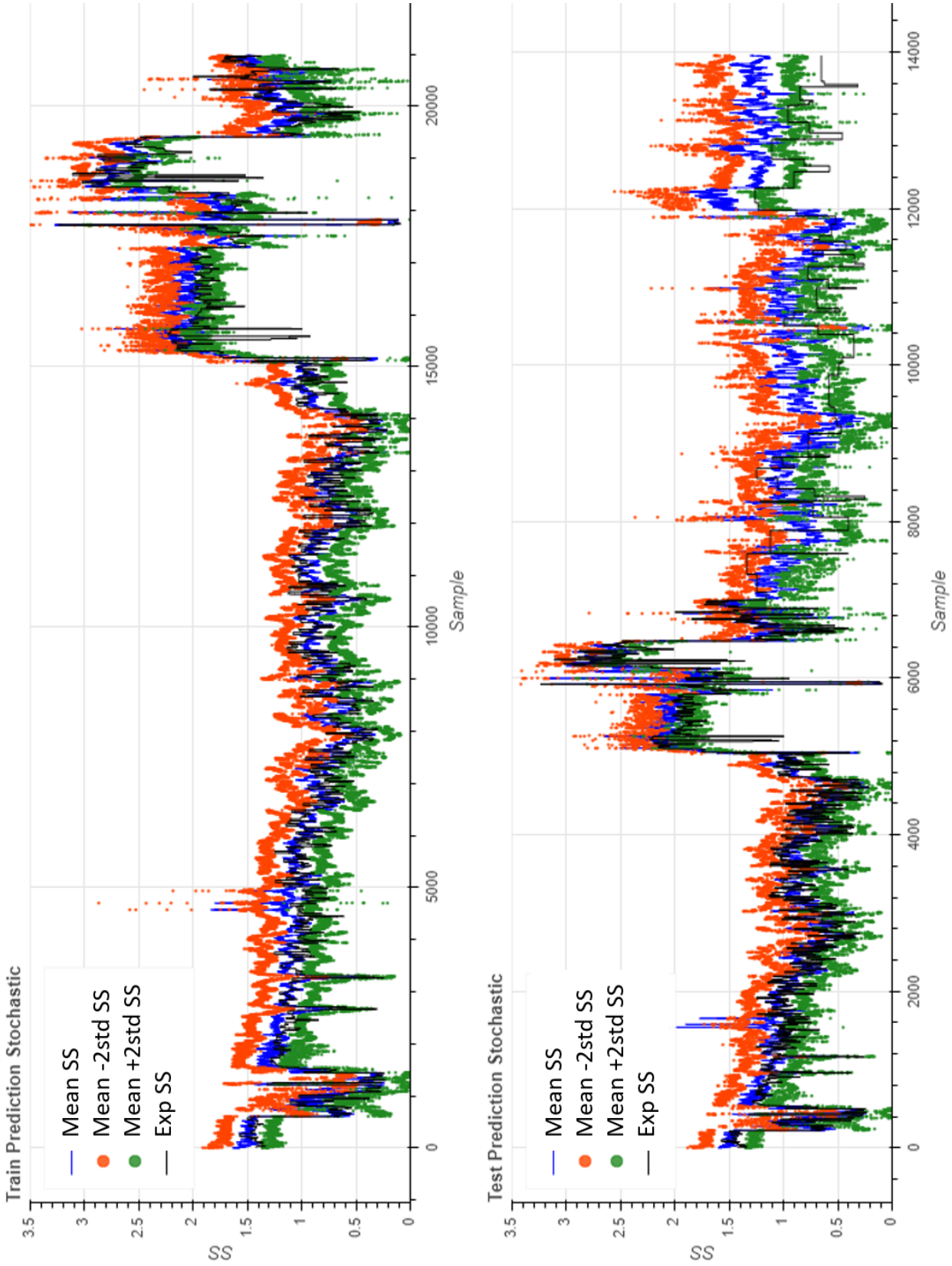


Figure 4.10: Case 3: Adapted PCA.

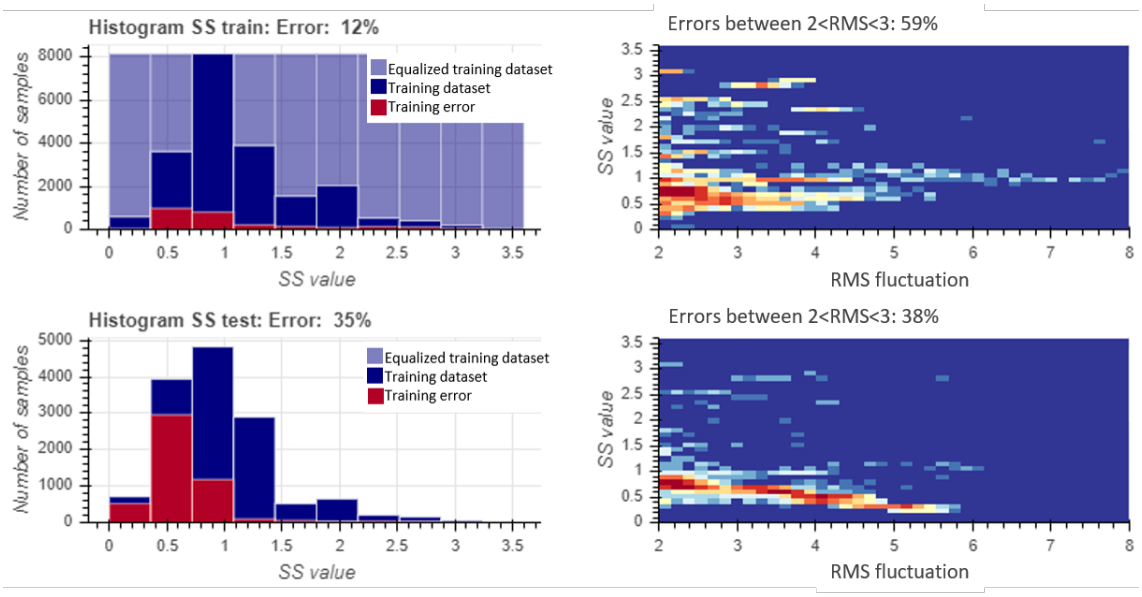
In Figs. 4.9 and 4.10 are represented the S_S estimation from the Case 3 from the Raw data and the PCA onine data respectively.

For this case both the training and test datasets had data from rock A and B, but this time, the Well Br2 which had a different S_S was presented just in the testing phase.

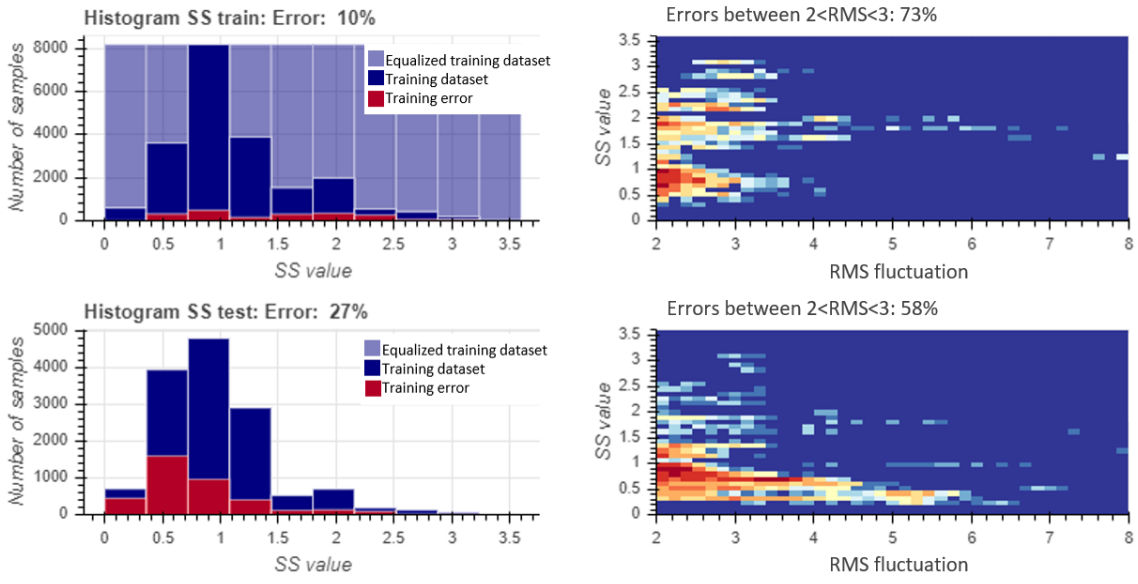
As seen in Figs. 4.9 and 4.10, the results obtained with both Raw data and Adapted PCA were visually very similar in spite of the estimations made when drilling Rock B and using Adapted PCA were more accurate. The one made with the Raw data stood in the interval of 2 RMS deviations, but the mean does not hit it for the most part. In the estimation made at Rock B with the Adapted PCA data was more accurate, with considerably smaller RMS deviations and with the estimation mean hitting the target.

When looking to the estimation for the Well Br2, for both cases it was not very good. But again the Adapted PCA data showed better results, being able to maintain for most of the time the S_S estimation interval under the right S_S.

By analyzing the histogram figures, Fig. 4.11, of this case, it can be clearly seen the superiority of the model that used the Adapted PCA. The error rate dropped from 12% to 10% while the concentration of mistaken estimations contained in a range of 2 to 3 RMS deviations went from 59% to 73% with the training dataset. The error rate dropped from 35% to 27% while the concentration of mistaken estimations contained in a range of 2 to 3 RMS deviations went from 38% to 58% with the test dataset. This means that the overall accuracy improved by using the Adapted PCA approach.



(a) Case 3 error evaluation: Raw data.



(b) Case 3 error evaluation: Adapted PCA.

Figure 4.11: Case 3 error evaluation.

4.4 Case 4: No domain extrapolation, using Well Br2

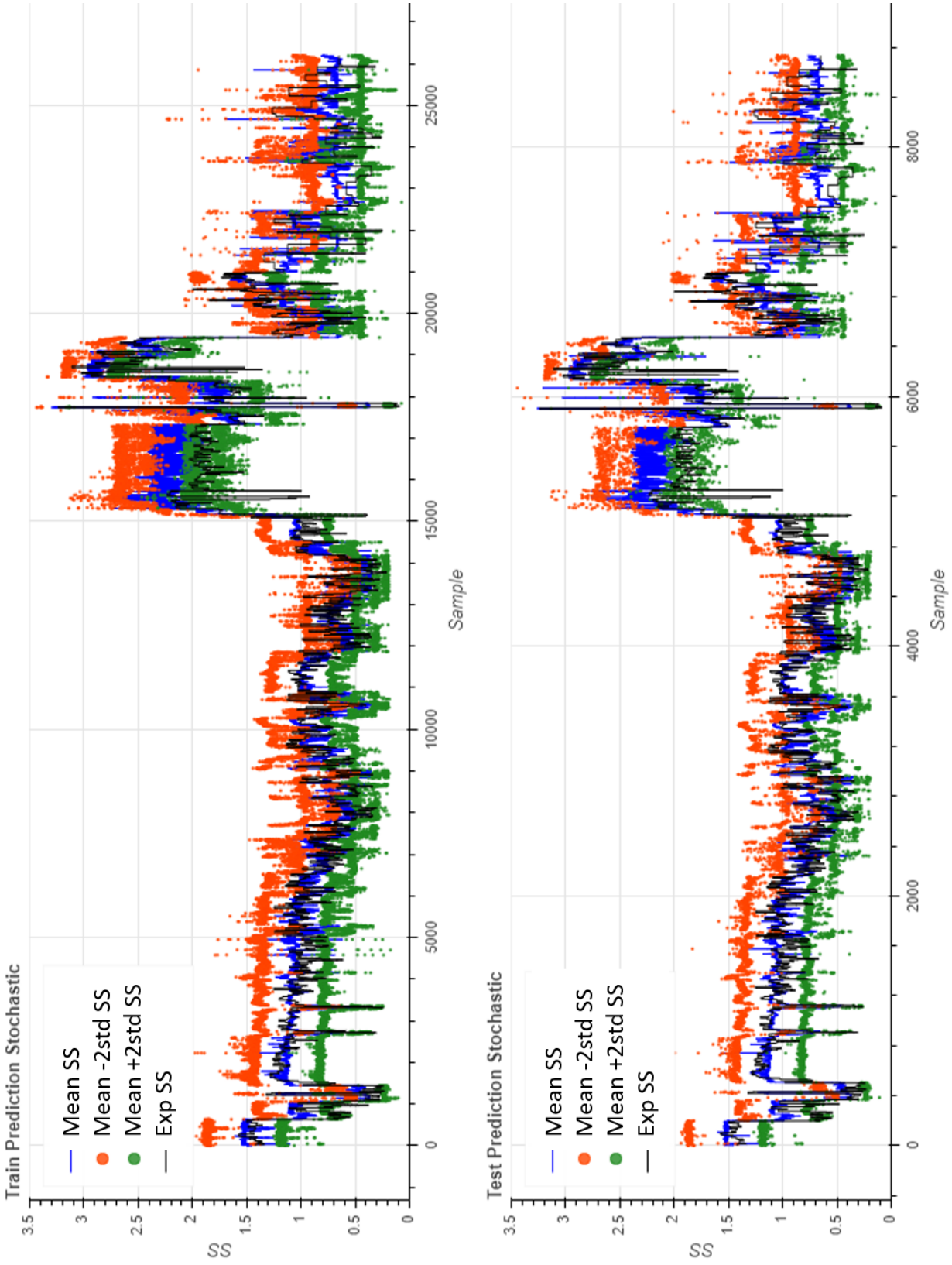


Figure 4.12: Case 4: Raw data.

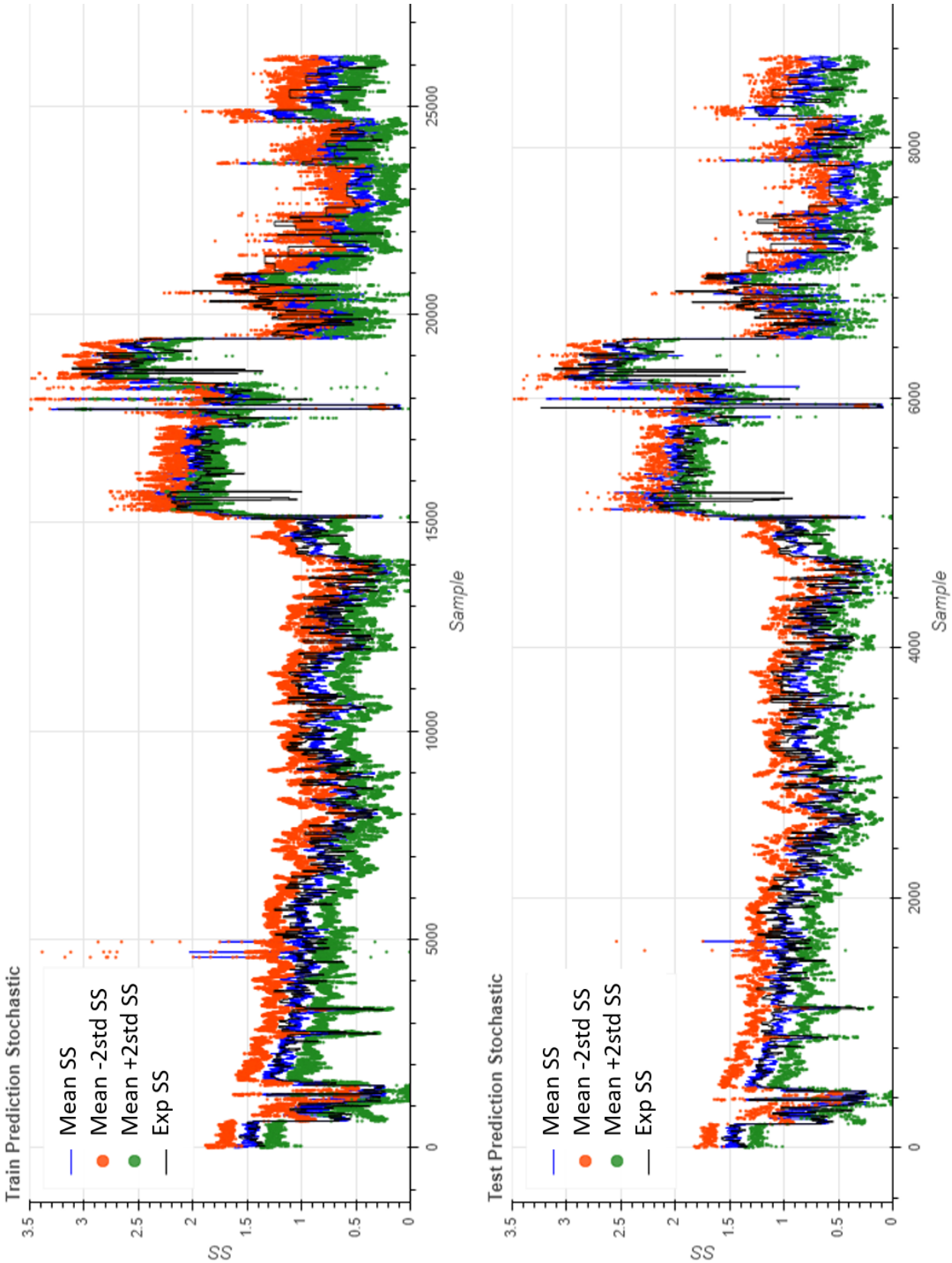


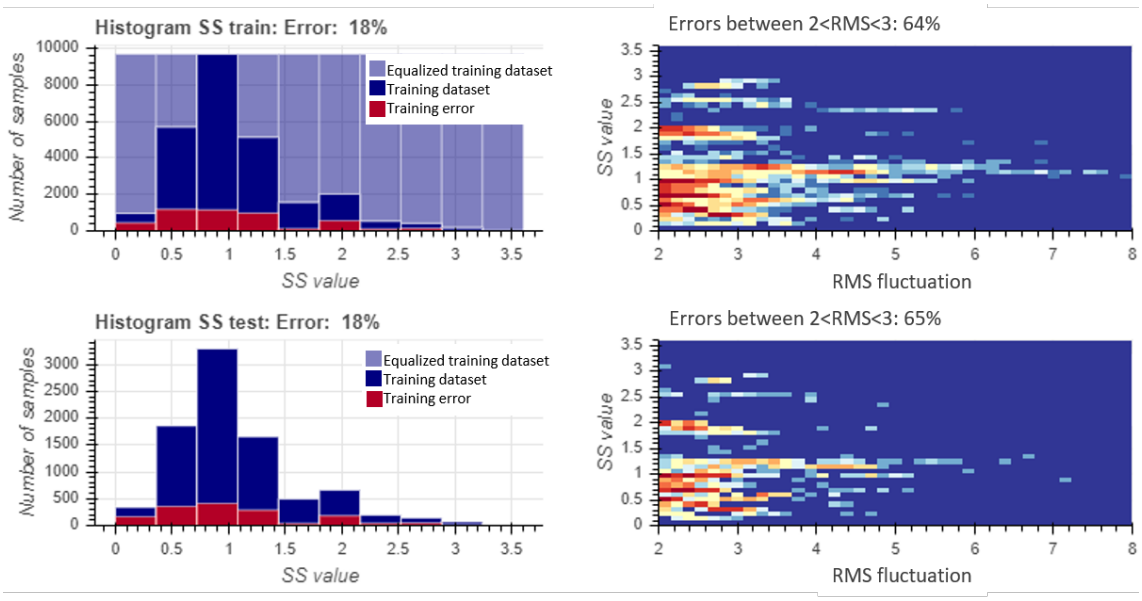
Figure 4.13: Case 4: Adapted PCA.

In Figs. 4.12 and 4.13 are represented the S_S estimation from the Case 4 from the Raw data and the PCA onine data respectively.

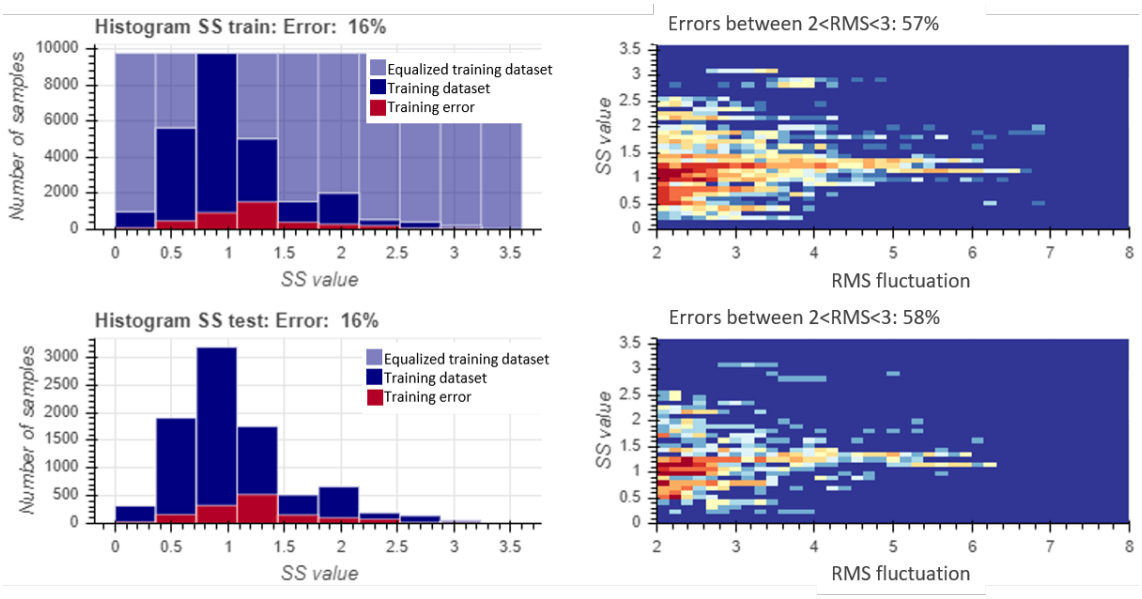
Both estimations made with the Raw data and the Adapted PCA shown similar results. The estimation, when the network is given the data from Well Br2 in the training phase, become much better for this well.

By analyzing the Figs. 4.12 and 4.13 again can be noted that the network that used the Adapted PCA overall shown a narrower interval with +/- 2 RMS from the mean, with smaller RMS deviations. The estimations for the data from Well Br2 was a little better as well.

Although the error rate improved just a little, from 18% to 16%, the mistaken estimations contained between a range of 2 and 3 RMS deviations became worse, went from 64% to 57% with the training dataset and from 65% to 58% with the test dataset. This negative effect can be assigned to the narrower interval. Finally, taking this in consideration, both estimations were very similar, but the more precise output from the Adapted PCA network may be a indicator of its superiority.



(a) Case 4 error evaluation: Raw data.



(b) Case 4 error evaluation: Adapted PCA.

Figure 4.14: Case 4 error evaluation.

Chapter 5

Conclusions

This dissertation made a exploration in the S_S estimation using only surface data with the use of Neural Networks. It also evaluated how the data preprocessing by applying the Adapted PCA affected the results.

The main objective, to estimate the torsional vibration severity factor (S_S) was accomplished with accurate rate considering intervals of 2 and 3 RMS deviations as shown in the table below:

Table 5.1: Training and testing errors obtained in each case, considering an interval of 2 and 3 RMS deviations.

		Case 1		**Case 2**		**Case 3**		**Case 4**	
		Error rate < 2 RMS	**Error rate < 3 RMS**	**Error rate < 2 RMS**	**Error rate < 3 RMS**	**Error rate < 2 RMS**	**Error rate < 3 RMS**	**Acc. rate < 2 RMS**	**Acc. rate < 3 RMS**
Raw data	**Training dataset**	11%	5%	13%	4%	12%	5%	18%	6%
Adapted PCA	**Training dataset**	6%	1%	8%	2%	10%	3%	16%	7%
Raw data	**Testing dataset**	46%	38%	13%	4%	35%	22%	18%	6%
Adapted PCA	**Testing dataset**	36%	19%	8%	2%	27%	11%	16%	7%

Among the 4 different cases that were studied, Case 2 and 4 showed the best results in the testing phase. They represent a scenario in which situations the data used to estimate S_S had previously been seen by the neural network during the training phase. It is feasible to assume that once the network has a large amount of drilling data to perform its training and thus containing a greater number of operating scenarios, it will obtain results similar to those obtained in Cases 2 and 4. The other cases, which require some extrapolation of the training domain by the network, did not show such satisfactory results, with considerably high error rates, especially for the network that used the raw data. These results tend to improve substantially once the network has data from many wells to train, as the fluctuation from this extrapolation would be shorter. Nevertheless, for the results obtained with

the data preprocessed with the Adapted PCA, the interval and magnitude of the estimated value provided information.

Although Cases 2 and 4 are in similar situations, trained with data from conditions similar to those of the test, the error rate of Case 4 was, for the network trained with the Adapted PCA, considerably higher than Case 2 and practically the same as Case 4 trained with the raw data. This fact is due to the uncertainty added to the model with data from Well Br2, which has different characteristics. The network could not adapt to it with the same success as Case 2. Preprocessing the data with the Adapted PCA, although it did not result in a lower error rate, resulted in a smaller RMS deviation. Considering that the final result obtained has the same error rate and narrower interval, the model improved.

It is noteworthy, once again, that these results were obtained only with the use of data from 3 wells and, one of them, Well Br2, with a very low data recording rate. The result obtained makes it quite promising to explore this technique with new data in order to make the method more robust and reliable.

It also shown that a simpler neural network, once fed with the Adapted PCA data proposed at Chapter 2, could obtain better results than a larger one fed with raw data.

5.1 Future Steps

Once the methodology was developed, a crucial future step is to validate with data from other wells. This step has the potential to create a robust tool that feeds the drilling operator with trustworthy estimations of the downhole scenario.

Hopefully, these new wells are going to have usable measurements from lateral and axial vibrations so the methodology could also be applied estimate them.

After presenting this work to Petrobras, they have shown great interest in the developed method and proposed the creation of a patent out of this. Research in patents was already made, and a similar one was found [58]. But it is constrained to using data with a much higher frequency acquisition rate. Therefore, the writing and legal process of creating a patent will be worked on in the next months. Latter, an articled with the main achievements of this work will be written and submitted.

Bibliography

[1] LEINE, R. I., VAN CAMPEN, D. H. "Stick-Slip Whirl Interaction in Drillstring Dynamics". In: Rega, G., Vestroni, F. (Eds.), *IUTAM Symposium on Chaotic Dynamics and Control of Systems and Processes in Mechanics*, pp. 287–296, Dordrecht, 2005. Springer Netherlands.

[2] LOBO, D. *Análise Dinâmica de uma Coluna de Perfuração Durante a Mudança de Características da Rocha com Quantificação de Incertezas*. Phd Thesis, Universidade Federal do Rio de Janeiro, 2016.

[3] SCIKIT-LEARN. "Underfitting vs. Overfitting". 2019. `https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html`, Acessado em 2019-09-06.

[4] OSTMEYER, J., COWELL, L. "Machine Learning on Sequential Data Using a Recurrent Weighted Average", *Neurocomputing*, v. 331, 03 2017. doi: 10.1016/j.neucom.2018.11.066.

[5] ZHANG, J., XU, Y., XUE, J., et al. "Real-time prediction of solar radiation based on online sequential extreme learning machine". In: *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 53–57, May 2018.

[6] WANG, Y., ZHOU, J., CHEN, K., et al. "Water quality prediction method based on LSTM neural network". In: *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 1–5, Nov 2017.

[7] YU, L., QU, J., GAO, F., et al. "A Novel Hierarchical Algorithm for Bearing Fault Diagnosis Based on Stacked LSTM", *Shock and Vibration*, v. 2019, pp. 1–10, 01 2019. doi: 10.1155/2019/2756284.

[8] CHRISTIAN, E. D. "Identifying the Optimum Zone for Reducing Drill String Vibrations", *SPE Annual Technical Conference and Exhibition, 9-11 October, San Antonio, Texas, USA*, 2017. ISSN: 0920-4105. doi: https:

//doi.org/10.2118/189284-STU. Available in: <https://www.onepetro.org/conference-paper/SPE-189284-STU>.

[9] W. DEGREGORY, K., KUIPER, P., DESILVIO, T., et al. "A review of machine learning in obesity: Machine learning in obesity research", *Obesity Reviews*, v. 19, 02 2018. doi: 10.1111/obr.12667.

[10] LIU, N. T., SALINAS, J. "Machine learning in burn care and research: A systematic review of the literature", *Burns*, v. 41, n. 8, pp. 1636 – 1641, 2015. ISSN: 0305-4179. doi: https://doi.org/10.1016/j.burns.2015.07.001. Available in: <http://www.sciencedirect.com/science/article/pii/S0305417915002004>.

[11] T. BUTLER, K., DAVIES, D., CARTWRIGHT, H., et al. "Machine learning for molecular and materials science", *Nature*, v. 559, 07 2018. doi: 10.1038/s41586-018-0337-2.

[12] SILVER, D., HUANG, A., MADDISON, C., et al. "Mastering the game of Go with deep neural networks and tree search", *Nature*, v. 529, pp. 484–489, 01 2016. doi: 10.1038/nature16961.

[13] MACDONALD, K., BJUNE, J. "Failure analysis of drillstrings", *Engineering Failure Analysis*, v. 14, n. 8, pp. 1641 – 1666, 2007. ISSN: 1350-6307. doi: https://doi.org/10.1016/j.engfailanal.2006.11.073. Available in: <http://www.sciencedirect.com/science/article/pii/S1350630706002251>. Papers presented at the Second International Conference on Engineering Failure Analysis (Toronto, Canada, 12–15 September 2006) Part II.

[14] MITCHELL, T. M. *Machine Learning*. 1 ed. New York, 1997. ISBN: 0070428077 9780070428072.

[15] ABU-MOSTAFA, Y. S., MAGDON-ISMAIL, M., LIN, H.-T. *Learning From Data*. AMLBook, 2012. ISBN: 1600490069, 9781600490064.

[16] E. RUMELHART, D., E. HINTON, G., J. WILLIAMS, R. "Learning Representations by Back Propagating Errors", *Nature*, v. 323, pp. 533–536, 10 1986. doi: 10.1038/323533a0.

[17] CSÁJI, B. C. *Approximation with Artificial Neural Networks*. Msc thesis, Faculty of Sciences Eötvös Loránd University, 2001.

[18] HOCHREITER, S., SCHMIDHUBER, J. "Long Short-Term Memory", *Neural Computation*, v. 9, n. 8, pp. 1735–1780, 1997. doi: 10.1162/neco.

1997.9.8.1735. Available in: <https://doi.org/10.1162/neco.1997.9.8.1735>.

[19] CHUNG, J., GULCEHRE, C., CHO, K., et al. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", 12 2014.

[20] HUANG, G.-B., ZHU, Q.-Y., SIEW, C.-K. "Extreme learning machine: Theory and applications", *Neurocomputing*, v. 70, n. 1, pp. 489 – 501, 2006. ISSN: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2005.12.126. Available in: <http://www.sciencedirect.com/science/article/pii/S0925231206000385>. Neural Networks.

[21] SCHMIDHUBER, J. "Deep learning in neural networks: An overview", *Neural Networks*, v. 61, pp. 85 – 117, 2015. ISSN: 0893-6080. doi: https://doi.org/10.1016/j.neunet.2014.09.003. Available in: <http://www.sciencedirect.com/science/article/pii/S0893608014002135>.

[22] BENGIO, Y., COURVILLE, A., VINCENT, P. "Representation Learning: A Review and New Perspectives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 8, pp. 1798–1828, Aug 2013. ISSN: 0162-8828. doi: 10.1109/TPAMI.2013.50.

[23] ZHU, X., SOBIHANI, P., GUO, H. "Long Short-Term Memory Over Recursive Structures". In: Bach, F., Blei, D. (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*, v. 37, *Proceedings of Machine Learning Research*, pp. 1604–1612, Lille, France, 07–09 Jul 2015. PMLR. Available in: <http://proceedings.mlr.press/v37/zhub15.html>.

[24] LI, Y., ZHANG, S., YIN, Y., et al. "A Novel Online Sequential Extreme Learning Machine for Gas Utilization Ratio Prediction in Blast Furnaces". In: *Sensors*, 2017.

[25] KUMAR, N. K., SAVITHA, R., MAMUN, A. A. "Ocean wave height prediction using ensemble of Extreme Learning Machine", *Neurocomputing*, v. 277, pp. 12 – 20, 2018. ISSN: 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.03.092. Available in: <http://www.sciencedirect.com/science/article/pii/S0925231217313942>. Hierarchical Extreme Learning Machines.

[26] LONDHE, S. N., PANCHANG, V. "One-Day Wave Forecasts Based on Artificial Neural Networks", *Journal of Atmospheric and Oceanic Technology*, v. 23, n. 11, pp. 1593–1603, 2006. doi: 10.1175/JTECH1932.1. Available in: <https://doi.org/10.1175/JTECH1932.1>.

[27] D. DASHEVSKIY, U. O. H., V. DUBINSKY, S., , et al. "Application of Neural Networks for Predictive Control in Drilling Dynamics", *SSPE Annual Technical Conference and Exhibition, 3-6 October, Houston, Texas*, 1999. doi: https://doi.org/10.2118/56442-MS. Available in: <https://www.onepetro.org/conference-paper/SPE-56442-MS>.

[28] BEHZAD ELAHIFAR, U. O. L., GERHARD THONHAUSER, U. O. L., RUDOLF KONRAD FRUHWIRTH, T. T. D. E. G., et al. "ROP Modeling using NeuralNetwork and Drill String Vibration Data", *SPE Kuwait International Petroleum Conference and Exhibition, 10-12 December, Kuwait City, Kuwait*, 2012. doi: https://doi.org/10.2118/163330-MS. Available in: <https://www.onepetro.org/conference-paper/SPE-163330-MS>.

[29] SUN, J., LI, Q., CHEN, M., et al. "Optimization of models for a rapid identification of lithology while drilling - A win-win strategy based on machine learning", *Journal of Petroleum Science and Engineering*, v. 176, pp. 321 – 341, 2019. ISSN: 0920-4105. doi: https://doi.org/10.1016/j.petrol.2019.01.006. Available in: <http://www.sciencedirect.com/science/article/pii/S092041051930004X>.

[30] ZHONG, Y., LI, R. "Application of principal component analysis and least square support vector machine to lithology identification", *Well Logging Technology*, v. 33, n. 5, pp. 425–429, 2009. Available in: <www.scopus.com>. Cited By :13.

[31] "Study on Identification of Oil/Gas and Water Zones in Geological Logging Base on Support-Vector Machine". In: Cao, B., Li, T.-F., Zhang, C.-Y. (Eds.), *Fuzzy Information and Engineering Volume 2*, pp. 849–857, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN: 978-3-642-03664-4.

[32] LI, X., LI, H. "A new method of identification of complex lithologies and reservoirs: Task-driven data mining", *Journal of Petroleum Science and Engineering*, v. 109, n. 5, pp. 241–249, 2013. Available in: <www.scopus.com>. Cited By :13.

[33] XIE, Y., ZHU, C., ZHOU, W., et al. "Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances", *Journal of Petroleum Science and Engineering*, v. 160, pp. 182 – 193, 2018. ISSN: 0920-4105. doi: https://doi.org/10.1016/j.petrol.2017.10.028. Available in: <http://www.sciencedirect.com/science/article/pii/S0920410517308094>.

[34] OTHMAN, A. A., GLOAGUEN, R. "Integration of spectral, spatial and morphometric data into lithological mapping: A comparison of different Machine Learning Algorithms in the Kurdistan Region, NE Iraq", *Journal of Asian Earth Sciences*, v. 146, pp. 90 – 102, 2017. ISSN: 1367-9120. doi: https://doi.org/10.1016/j.jseaes.2017.05.005. Available in: <http://www.sciencedirect.com/science/article/pii/S1367912017302225>.

[35] ZHANG, X., ZHANG, H., GUO, J., et al. "Auto measurement while drilling mud pulse signal recognition based on deep neural network", *Journal of Petroleum Science and Engineering*, v. 167, pp. 37 – 43, 2018. ISSN: 0920-4105. doi: https://doi.org/10.1016/j.petrol.2018.04.004. Available in: <http://www.sciencedirect.com/science/article/pii/S0920410518303097>.

[36] YASHODHAN KESHAV GIDH, S., ARIFIN PURWANTO, S., HANI IBRAHIM, S. B. "Artificial Neural Network Drilling Parameter Optimization System Improves ROP by Predicting/Managing Bit Wear", *SPE Intelligent Energy International, 27-29 March, Utrecht, The Netherlands*, 2012. doi: https://doi.org/10.2118/149801-MS. Available in: <https://www.onepetro.org/conference-paper/SPE-149801-MS>.

[37] GRAPS, A. "An Introduction to Wavelets", *IEEE Comput. Sci. Eng.*, v. 2, n. 2, pp. 50–61, 1995. ISSN: 1070-9924. doi: 10.1109/99.388960.

[38] SIMON, B. *Discover Signal Processing: An Interactive Guide for Engineers*. Wiley, 2008. ISBN: 978-0-470-51970-7.

[39] LILLY, J. M., OLHEDE, S. C. "Higher-Order Properties of Analytic Wavelets", *IEEE Transactions on Signal Processing*, v. 57, n. 1, pp. 146–160, jan 2009. ISSN: 1053-587X. doi: 10.1109/TSP.2008.2007607.

[40] LILLY, J. M., OLHEDE, S. C. "Generalized Morse Wavelets as a Superfamily of Analytic Wavelets", *IEEE Transactions on Signal Processing*, v. 60, n. 11, pp. 6036–6041, nov 2012. ISSN: 1053-587X. doi: 10.1109/TSP.2012.2210890.

[41] RITTO, T., AGUIAR, R., HBAIEB, S. "Validation of a drill string dynamical model and torsional stability", *Meccanica*, v. 52, 02 2017. doi: 10.1007/s11012-017-0628-y.

[42] I.T. JOLLIFFE. *Principal Component Analysis*. 2nd ed. New York, Springer-Verlag New York, 2002. doi: https://doi.org/10.1007/b98835.

[43] BONNIARD, M., CASTRO PINTO, F., DE LIMA, A. "Identificação de defeitos em bombas de grande porte através do método de decomposição ortogonal de karhunen - loève", , n. November, pp. 669–686, 2018.

[44] GOODFELLOW, I., BENGIO, Y., COURVILLE, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[45] HE, K., ZHANG, X., REN, S., et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", *IEEE International Conference on Computer Vision (ICCV 2015)*, v. 1502, 02 2015. doi: 10.1109/ICCV.2015.123.

[46] ZHANG, C., BENGIO, S., HARDT, M., et al. "Understanding deep learning requires rethinking generalization", *CoRR*, v. abs/1611.03530, 2016. Available in: <http://arxiv.org/abs/1611.03530>.

[47] HINTON, G. E., SRIVASTAVA, N., KRIZHEVSKY, A., et al. "Improving neural networks by preventing co-adaptation of feature detectors", *CoRR*, v. abs/1207.0580, 2012. Available in: <http://arxiv.org/abs/1207.0580>.

[48] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, v. 15, pp. 1929–1958, 2014. Available in: <http://jmlr.org/papers/v15/srivastava14a.html>.

[49] GAL, Y., GHAHRAMANI, Z. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning", *arXiv e-prints*, art. arXiv:1506.02142, Jun 2015.

[50] GLOROT, X., BENGIO, Y. "Understanding the difficulty of training deep feedforward neural networks". In: Teh, Y. W., Titterington, M. (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, v. 9, *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. Available in: <http://proceedings.mlr.press/v9/glorot10a.html>.

[51] LECUN, Y., BOTTOU, L., ORR, G., et al. "Efficient BackProp", 08 2000.

[52] SAXE, A., MCCLELLAND, J., GANGULI, S. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks", 12 2013.

[53] KINGMA, D., BA, J. "Adam: A Method for Stochastic Optimization", *International Conference on Learning Representations*, 12 2014.

[54] DUCHI, J., HAZAN, E., SINGER, Y. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, v. 12, pp. 2121–2159, 07 2011.

[55] KESKAR, N. S., MUDIGERE, D., NOCEDAL, J., et al. "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima", *CoRR*, v. abs/1609.04836, 2016. Available in: <http://arxiv.org/abs/1609.04836>.

[56] BENGIO, Y. "Practical recommendations for gradient-based training of deep architectures", *CoRR*, v. abs/1206.5533, 2012. Available in: <http://arxiv.org/abs/1206.5533>.

[57] MASTERS, D., LUSCHI, C. "Revisiting Small Batch Training for Deep Neural Networks", *CoRR*, v. abs/1804.07612, 2018. Available in: <http://arxiv.org/abs/1804.07612>.

[58] ERTAS, M. D., BAILEY, J. R., BURCH, D. N., et al. "Methods to estimate downhole drilling vibration indices from surface measurement". .

Appendix A - Choosing the Neural Network dimension

Choosing the right number of neurons and hidden layers of a neural network is a very difficult task. The final architecture greatly depends on the data that the network will work with. Because of this, to find a good starting point, a method that resembles the mesh convergence of the finite element method was used. All the tests made in this process used Case 1 because its the one that has the greatest domain extrapolation.

In the following tables are the convergence process made to find the neural network architecture. The number of neurons and hidden layers were step by step increased until no significant improvement could be observed. The column labeled as "Output saturated" refers to when a great number of estimations became zero or one, the limits of the sigmoid activation function of the output layer. The column "Regions with constant estimation" refers to regions where the network estimated almost the same SS. Both these columns happened most of the time when dealing with the regions of the domain extrapolation.

Table 5.2: Neural Network with Raw Data architecture convergence process

	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**	**Hidden Layer 4**	**Hidden Layer 5**	**Lowest Loss**	**Output saturated**	**Regions with constant estimation**
Activation Function	**tanh**							
	100	0	0	0	0	0.033	No	Yes
	400	0	0	0	0	0.030	No	Yes
	1000	0	0	0	0	0.028	No	Yes
	10.000	0	0	0	0	0.029	No	Yes
Activation Function	**tanh**	**tanh**						
	100	100	0	0	0	0.27	No	Yes
	1000	1000	0	0	0	0.23	No	Yes
	2000	2000	0	0	0	0.017	No	Yes
Activation Function	**PReLU**	**PReLU**	**PReLU**	**PReLU**				
	50	50	50	50	0	-		
	100	100	100	100	0	-		
	150	150	150	150	0	-		
	400	400	400	400	0	-		
	800	800	800	800	0	-		
Activation Function	**PReLU**	**PReLU**	**PReLU**	**PReLU**	**tanh**			
	50	50	50	50	25	-	Yes	Yes
	100	100	100	100	50	-	Yes	Yes
	150	150	150	150	75	-	Yes	Yes
	400	400	400	400	100	0.22	Yes	Yes
	400	800	800	400	100	0.017	No	Yes
	800	1200	1200	800	100	0.0166	No	Yes

Table 5.3: Neural Network with Adapted PCA architecture convergence process

	Hidden Layer 1	**Hidden Layer 2**	**Hidden Layer 3**	**Hidden Layer 4**	**Hidden Layer 5**	**Lowest Loss**	**Output saturated**	**Regions with constant estimation**
Activation Function	**tanh**							
	100	0	0	0	0	0.032	Yes	No
	400	0	0	0	0	0.029	Yes	No
	1000	0	0	0	0	0.029	Yes	No
	10.000	0	0	0	0	0.024	Yes	No
Activation Function	**tanh**	**tanh**						
	100	100	0	0	0	0.16	No	No
	1000	1000	0	0	0	0.12	No	No
Activation Function	**PReLU**	**PReLU**	**PReLU**	**PReLU**				
	50	50	50	50	0	0.015	No	Yes
	100	100	100	100	0	0.014	No	Yes
	150	150	150	150	0	0.015	No	Yes
	200	200	200	200	0	0.015	No	Yes
	400	400	400	400	0	0.014	No	Yes
Activation Function	**PReLU**	**PReLU**	**PReLU**	**PReLU**	**tanh**			
	50	50	50	50	10	0.022	No	No
	100	100	100	100	50	0.012	No	No
	150	150	150	150	75	0.012	No	No
	200	200	200	200	100	0.011	No	No
	400	400	400	400	200	0.011	No	No

Marked with yellow are the architectures which shown the lowest losses. The final choice was made to maintain the network as simple as possible. Because of this was chosen the one with five hidden layers for both Raw Data and Adapted PCA.

Appendix B - Evaluation of the distribution of the output

To test the distribution of the output obtained by utilizing the dropout layers, some histograms were made from the testing dataset of Case 1. 300 and 5000 simulations were made of two different moments, one at the testing sample number 1000 regarding Rock A, and other from the testing sample 7000 regarding the Rock B, the domain extrapolation.

In Figs. 5.1 and 5.2 are the histograms obtained after 5000 simulations made with the sample 1000 and 7000 of testing dataset.

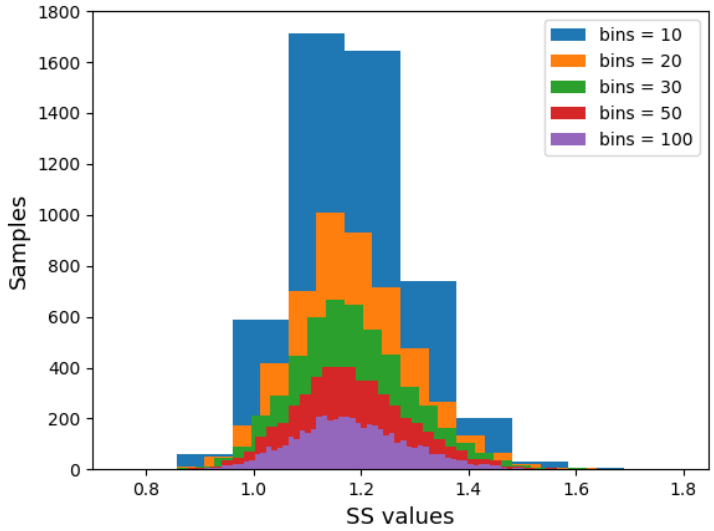


Figure 5.1: Histogram with 5000 simulation of sample 1000 (Rock A) of the test dataset of Case 1

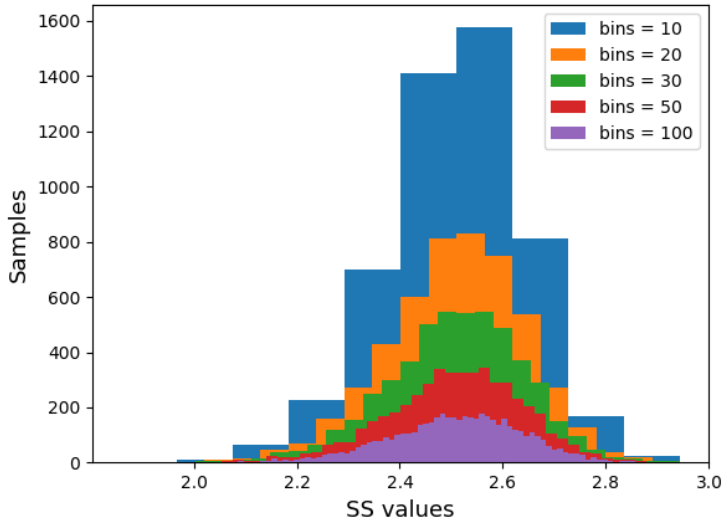


Figure 5.2: Histogram with 5000 simulation of sample 7000 (Rock B) of the test dataset of Case 1

In all the results shown in Chapter 4, for each sample was made 300 simulations. 300 was the lowest number of simulations that presented a similar result as 5000 or more simulations, regarding the RMS and mean values. In Figs. 5.3 and 5.4 are the histograms obtained after 300 simulations made with the sample 1000 and 7000 of testing dataset. Even though histograms became noisier, the results in the estimations, minding the calculated *mean* and the RMS, did not show a significant difference.

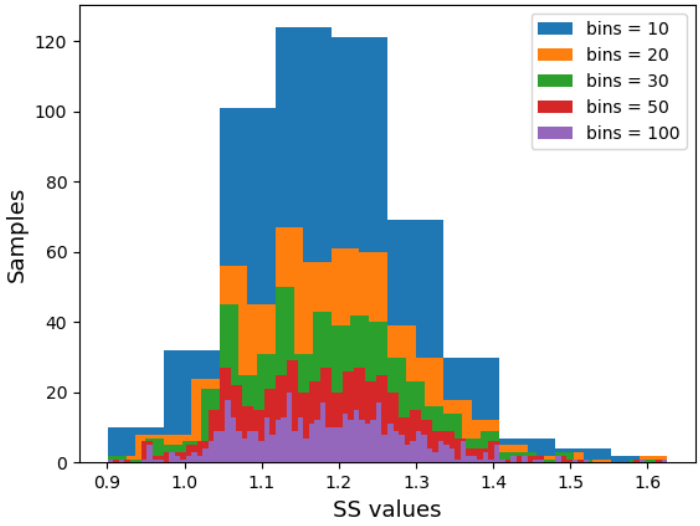


Figure 5.3: Histogram with 300 simulation of sample 1000 (Rock A) of the test dataset of Case 1

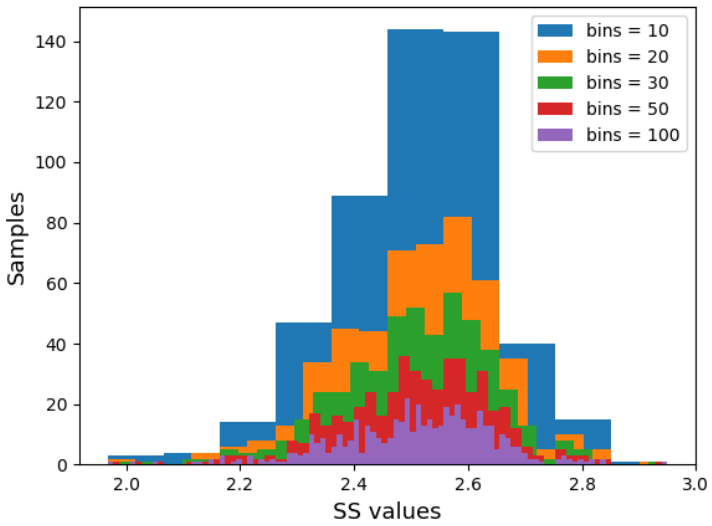


Figure 5.4: Histogram with 300 simulation of sample 7000 (Rock B) of the test dataset of Case 1

In all these histograms figures, it is clear that the distributions are not Gaussian. Because of this, representing the distribution with standard deviations is not right. Therefore, this work used RMS deviations to somehow quantify the variability of the network‘s output.