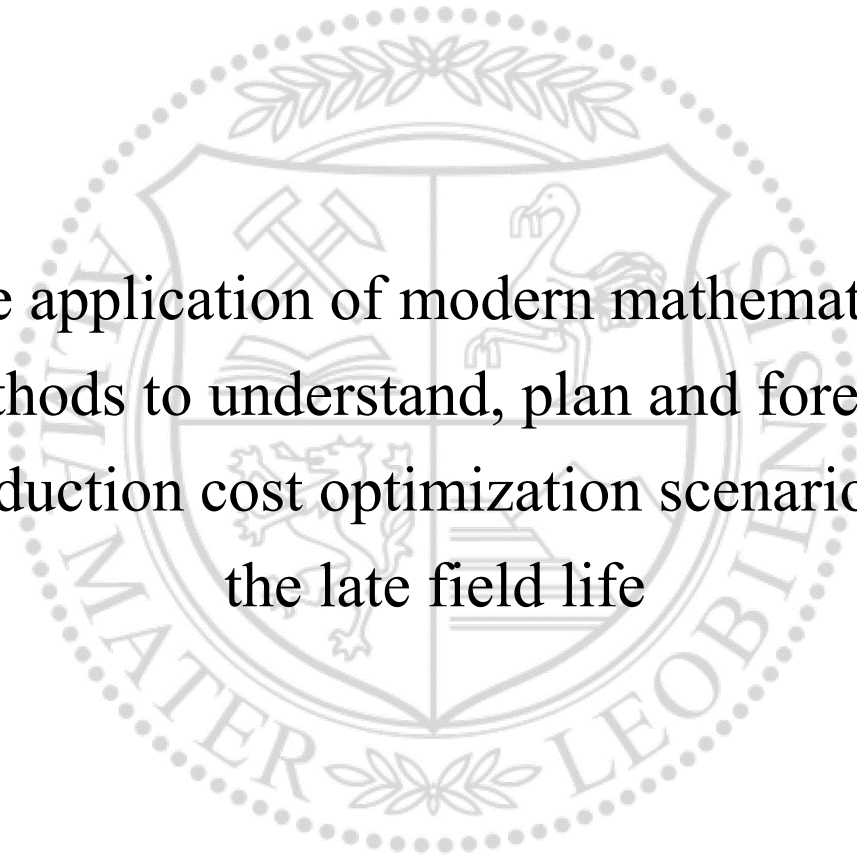




Chair of Petroleum and Geothermal Energy Recovery

Doctoral Thesis



The application of modern mathematical methods to understand, plan and forecast production cost optimization scenarios in the late field life

Dipl.-Ing. Viorica Sîrghii

May 2022

## Affidavit

I declare on oath that I wrote this thesis independently, did not use other than the specified sources and aids, and did not otherwise use any unauthorized aids.

I declare that I have read, understood, and complied with the guidelines of the senate of the Montanuniversität Leoben for "Good Scientific Practice".

Furthermore, I declare that the electronic and printed version of the submitted thesis are identical, both, formally and with regard to content.

19.05.2022

Datum/Date

A handwritten signature in blue ink, consisting of stylized initials and a surname, written over a horizontal line.

Unterschrift/Signature

## **Abstract**

Digitalization has had a significant impact on the complexity of oil and gas operations and has remodeled the entire industry. The ever-increasing amount of generated data accompanied by the energy transition, compliance with contemporary health, safety, and environmental regulations, followed by a cost-efficient production, represent both opportunities and challenges for the oil and gas companies nowadays.

Examples of machine learning and artificial intelligence are everywhere around us and have been used for solving complex engineering tasks to bring more efficiency and safety to our daily lives and within major industrial operations.

The aim of this research was to develop a novel approach for predicting and preventing failures in wells equipped with artificial lift production systems by using machine learning tools and artificial intelligence algorithms based on the extensive amount of data an oil and gas company is generating every day.

Here I created a diagnostic tool for the automatic identification of the sucker rod pump states and malfunctions using digitally generated dynamometer cards. The proposed solution based on artificial neural networks led to a high precision recognition algorithm which can be used in preventing potential well failures and optimized production.

Another major task was the identification of trends exhibited by the sucker rod pumps' behavior and forecasting future pump states based on the identified trend. Various models have been found, tested and the most fitting approach was selected. The selected model was able to accurately and reliably predict results, almost identical to the real data points. These predictions can be used in daily operations for avoiding potential failures and malfunctions in sucker rod pumps, reduce costs, risks, and increase the mean time between failures.

A novel automatic system for detecting and predicting unwanted events in the sucker rod pump operation was created. The results carry high degrees of precision, accuracy, and flexibility, which allow the application and extension of the model to other similar cases.

Presented methods and functionalities based on artificial intelligence techniques have demonstrated its power as an enabling technology capable of delivering outstanding outcomes and help solve complex problems.

## Kurzfassung

Die Welt, in der wir leben, verändert sich in rasantem Tempo. Die Digitalisierung hat die Komplexität des Öl- und Gasgeschäfts erheblich beeinflusst und die gesamte Branche umgestaltet. Die ständig wachsende Menge an generierten Daten, die mit Energiewende, Einhaltung der aktuellen Gesundheits-, sowie Sicherheits- und Umweltvorschriften und einer kosteneffizienten Produktion einhergehen; stellt für die Öl- und Gasunternehmen sowohl Chance als auch Herausforderung dar.

Beispiele für maschinelles Lernen und künstliche Intelligenz (KI) sind allgegenwärtig und werden für die Lösung komplexer technischer Aufgaben eingesetzt, um mehr Effizienz und Sicherheit in unser tägliches Leben und in große industrielle Prozesse zu bringen.

Ziel dieser Forschungsarbeit war die Entwicklung eines neuartigen Ansatzes zur Vorhersage und Verhinderung von Ausfällen in Bohrlöchern, die mit künstlichen Aufzugsproduktionssystemen ausgestattet sind, durch den Einsatz von Werkzeugen des maschinellen Lernens und Algorithmen der künstlichen Intelligenz für die umfangreichen Datenmengen, die ein Öl- und Gasunternehmen täglich erzeugt.

Die primäre Herausforderung dieser Arbeit war die Entwicklung eines Diagnosewerkzeugs für die automatische Erkennung des Zustands und der Fehlfunktionen der Gestängetiefpumpe anhand digital erzeugter Dynamometerkarten. Die vorgeschlagene Lösung auf der Grundlage künstlicher neuronaler Netze führte zu einem hochpräzisen Erkennungsalgorithmus, der zur Verhinderung potenzieller Bohrlochausfälle und zur Optimierung der Produktion eingesetzt werden kann.

Eine weitere wichtige Aufgabe war die Identifizierung von Trends im Verhalten der Gestängetiefpumpen und die Vorhersage künftiger Pumpenzustände auf der Grundlage der identifizierten Trends. Verschiedene mathematische Modelle wurden getestet, und der passendste Ansatz wurde ausgewählt. Das gewählte Modell war in der Lage, genaue und zuverlässige Ergebnisse vorherzusagen, die fast identisch mit den realen Datenpunkten sind. Diese Vorhersagen können im täglichen Betrieb eingesetzt werden, um potenzielle Ausfälle und Fehlfunktionen in Gestängetiefpumpen zu vermeiden, Kosten und Risiken zu reduzieren und die mittlere Zeit zwischen Ausfällen zu erhöhen.

Ein neuartiges automatisches System zur Erkennung und Vorhersage unerwünschter Ereignisse beim Betrieb von Gestängetiefpumpen wurde entwickelt. Die Ergebnisse zeichnen sich durch ein hohes Maß an Präzision, Genauigkeit und Flexibilität aus, was Anwendung und Erweiterung des Modells auf andere ähnliche Fälle ermöglicht.

Die vorgestellten Methoden und Funktionen, die auf Techniken der künstlichen Intelligenz beruhen, haben die Leistungsfähigkeit der KI als eine Technologie gezeigt, die hervorragende Ergebnisse liefert und zur Lösung komplexer Probleme beitragen kann.

## **Acknowledgments**

This work would not have been possible without the support and help of many people. First of all, I would like to express my deepest appreciation and gratitude to Dipl. -Ing. Dr. mont. Rudolf Fruhwirth, who provided me with invaluable guidance throughout the entire process, read my numerous corrections, always encouraged me and helped make some sense of the confusion.

I would also like to thank Univ. – Prof. Dipl. -Ing. Dr. mont. Herbert Hofstätter for his patience, guidance, and support, for the continuous faith in me over all these years, the way he cared about my work and the promptness to which he responded to my requests.

I would like to extend my recognition to Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Norbert Seifert for his generous contribution, advice, and his patience in reading and correcting my multiple revisions of the thesis.

Special thanks to OMV for the financial support and in particular to Dipl. -Ing. Reinhard Pongratz for his contribution to this project and technical support.

I would like to thank my family for their endless care and moral support, for the times when their endless love and encouragement were my main pillars in moving forward.

Last but not least, special thanks to my colleagues and friends who were always there when I needed help or someone to talk to.

# Table of Contents

1	Introduction .....	- 18 -
1.1	Research Problem and Motivation .....	- 18 -
1.2	Outline .....	- 19 -
2	Literature Review .....	- 21 -
2.1	Machine Learning .....	- 21 -
2.1.1	Preparing the Data Set .....	- 21 -
2.1.2	Selecting the Algorithm.....	- 22 -
2.1.3	Training the Neural Network .....	- 22 -
2.1.4	Applying the Algorithm.....	- 23 -
2.2	Machine Learning Styles .....	- 23 -
2.2.1	Supervised Learning.....	- 23 -
2.2.2	Unsupervised Learning.....	- 23 -
2.2.3	Semi-supervised Learning .....	- 23 -
2.2.4	Reinforcement Learning .....	- 24 -
2.3	Artificial Neural Networks .....	- 24 -
2.3.1	Artificial Neural Network Components .....	- 24 -
2.3.2	Artificial Neural Networks Types .....	- 31 -
2.4	Trend identification in Time Series Data.....	- 37 -
2.4.1	Time Series Forecasting.....	- 38 -
2.4.2	Time Series Forecasting Application.....	- 38 -
2.4.3	Time Series Trend Analysis .....	- 38 -
2.4.4	Time Series Seasonality Analysis .....	- 39 -
2.5	Moving Average Models.....	- 39 -
2.5.1	Moving Average Model (MA) .....	- 40 -
2.5.2	Autoregressive Integrated Moving Average (ARIMA).....	- 40 -
2.5.3	ACF and PACF .....	- 40 -
3	Application of ML.....	- 42 -
3.1	Application of ML in Oil and Gas Industry.....	- 42 -
3.1.1	Upstream.....	- 42 -
3.1.2	Midstream.....	- 43 -
3.1.3	Downstream .....	- 43 -
3.1.4	Back-office Management .....	- 43 -
3.2	Application of ML in Other Industries.....	- 44 -
4	Data Acquisition .....	- 46 -
4.1	Available Data.....	- 46 -
4.1.1	Digital Dynamometer Cards.....	- 46 -
4.1.2	Analogue Dynamometer Cards.....	- 46 -
4.1.3	Workover Reports.....	- 47 -

4.1.4	Production Data.....	- 47 -
4.1.5	Equipment Data.....	- 48 -
4.1.6	Well Schematics.....	- 48 -
4.1.7	Surface Data .....	- 49 -
4.2	Data Integration .....	- 50 -
5	Data QC/QA.....	- 51 -
5.1	DC Data Overview .....	- 51 -
5.2	Data Visualization .....	- 52 -
6	Features for Model Building.....	- 54 -
6.1	Statistical Features .....	- 57 -
6.2	Physical Features .....	- 64 -
6.3	Elliptic Fourier Transform Features .....	- 79 -
6.4	Feature Selection.....	- 101 -
6.4.1.	Sequential Forward Selection (SFS).....	- 101 -
6.4.2.	Sequential Backward Selection (SBS).....	- 102 -
6.4.3.	Random Search Extension (RSX) .....	- 102 -
6.4.4.	Exhaustive Search Extension (ESX).....	- 103 -
6.4.5	Results Evaluation.....	- 103 -
7	Heuristic QC model .....	- 111 -
7.1	ANN Training .....	- 111 -
7.1.1	Training Results .....	- 112 -
7.1.2	Challenges Encountered in ANN Training.....	- 113 -
8	Heuristic Pump State Model .....	- 116 -
8.1	Sucker Rod Pump Diagnosis .....	- 116 -
8.2	Sucker Rod Pump States.....	- 117 -
8.3	Sucker Rod Pump States Identification – ANN Training.....	- 119 -
8.4	Challenges Encountered in ANN Training .....	- 122 -
8.4.1	Distinguishing Between Similar Dynamometer Cards Shapes .....	- 122 -
9	Trend Analysis .....	- 124 -
9.1	ARIMA .....	- 124 -
9.1.1	The Main Challenge .....	- 125 -
9.2	Forecasting .....	- 129 -
9.3	Trend Analysis .....	- 129 -
9.3.1	Forecasted Values.....	- 131 -
9.3.2	Trend of the Forecasted Values.....	- 132 -
10	Conclusions and Recommendations.....	- 135 -
10.1	Contributions.....	- 135 -
10.2	Findings .....	- 135 -
10.3	Recommendations .....	- 136 -
11	References .....	- 138 -

12 Appendices ..... - 141 -  
Appendix A: Snapshots of Data Matrix..... - 141 -  
Appendix B: QC Models Performance Evaluation ..... - 142 -  
Appendix C: Pump States Models Performance Evaluation ..... - 145 -



# List of Figures

- Figure 1: Artificial Neural Network Components (McCullum 2020)..... - 25 -
- Figure 2: Single neuron structure ..... - 25 -
- Figure 3: Neuron with multiple inputs..... - 26 -
- Figure 4: ANN with one hidden layer ..... - 27 -
- Figure 5: Activation functions (left – Sigmoid, right – ReLu)..... - 27 -
- Figure 6: ANN with ReLu activation function..... - 28 -
- Figure 7: Mean Absolute Error..... - 29 -
- Figure 8: Gradient descent ..... - 30 -
- Figure 9: Perceptron architecture (Shukla and Iriondo 2020)..... - 31 -
- Figure 10: Feed-Forward neural network architecture ..... - 32 -
- Figure 11: Deep Feed-Forward network architecture..... - 32 -
- Figure 12: Recurrent neural network architecture ..... - 33 -
- Figure 13: Long/Short Term Memory network architecture ..... - 33 -
- Figure 14: Gated Recurrent Unit architecture ..... - 34 -
- Figure 15: Auto Encoder architecture ..... - 34 -
- Figure 16: Markov Chain architecture..... - 35 -
- Figure 17: Hopfield Network architecture..... - 35 -
- Figure 18: Boltzmann Machine architecture..... - 36 -
- Figure 19: Deep Belief network architecture ..... - 36 -
- Figure 20: Deep Convolutional network architecture ..... - 37 -
- Figure 21: Deconvolutional neural network architecture ..... - 37 -
- Figure 22: Example of upward trend..... - 39 -
- Figure 23: Seasonality example ..... - 39 -
- Figure 24: An example of an analogue dynamometer card downloaded from GDB ..... - 47 -
- Figure 25: An example of workover reports file downloaded from GDB ..... - 47 -
- Figure 26: An example of production data file downloaded from GDB ..... - 48 -
- Figure 27: An example of production data downloaded from GDB ..... - 48 -
- Figure 28: An example of equipment data file downloaded from GDB ..... - 48 -
- Figure 29: An example of well schematics file downloaded from GDB..... - 49 -
- Figure 30: An example of surface data file downloaded from GDB ..... - 50 -
- Figure 31: Examples of files per day..... - 51 -
- Figure 32: Surface dynamometer card ..... - 52 -
- Figure 33: Example of erroneous measurement..... - 53 -
- Figure 34: Example of good measurement..... - 53 -
- Figure 35: Raw sucker rod position, m ..... - 58 -
- Figure 36: Raw sucker rod load, N ..... - 58 -
- Figure 37: Raw sucker rod velocity, m/s ..... - 58 -
- Figure 38: Raw sucker rod yank, N/s..... - 58 -
- Figure 39: Raw sucker rod position, m ..... - 59 -
- Figure 40: Raw sucker rod load, N ..... - 59 -
- Figure 41: Raw sucker rod velocity, m/s ..... - 59 -
- Figure 42: Raw sucker rod yank, N/s..... - 59 -
- Figure 43: Raw sucker rod position, m ..... - 60 -
- Figure 44: Raw sucker rod load, N ..... - 60 -
- Figure 45: Raw sucker rod velocity, m/s ..... - 60 -
- Figure 46: Raw sucker rod yank, N/s..... - 60 -
- Figure 47: Processed sucker rod position, m..... - 60 -
- Figure 48: Processed sucker rod load, N..... - 60 -
- Figure 49: Processed sucker rod velocity, m/s ..... - 61 -
- Figure 50: Processed sucker rod yank, N/s ..... - 61 -
- Figure 51: Raw sucker rod position, m ..... - 61 -
- Figure 52: Raw sucker rod load, N ..... - 61 -

Figure 53: Raw sucker rod yank, N/s.....	- 62 -
Figure 54: Processed sucker rod load, N.....	- 62 -
Figure 55: Processed sucker rod position, m.....	- 62 -
Figure 56: Processed sucker rod position, m.....	- 62 -
Figure 57: Processed sucker rod position, m.....	- 62 -
Figure 58: Raw sucker rod velocity, m/s.....	- 62 -
Figure 59: Raw sucker rod position, m.....	- 63 -
Figure 60: Raw sucker rod load, N.....	- 63 -
Figure 61: Raw sucker rod velocity, m/s.....	- 63 -
Figure 62: Raw sucker rod yank, N/s.....	- 63 -
Figure 63: Processed sucker rod velocity, m/s.....	- 63 -
Figure 64: Processed sucker rod yank, N/s.....	- 63 -
Figure 65: Processed sucker rod position, m.....	- 64 -
Figure 66: Processed sucker rod load, N.....	- 64 -
Figure 67: From stroke detector, raw sucker rod position, m; Normalized absolute differences in the whole stroke.....	- 65 -
Figure 68: From stroke detector, raw sucker rod position, m; Normalized absolute differences at the end of the stroke.....	- 65 -
Figure 69: From stroke detector, raw sucker rod load N; Normalized absolute differences at the end of the stroke.....	- 66 -
Figure 70: Fractal dimensions, raw sucker rod load, N; Rescaled range statistical analysis.....	- 66 -
Figure 71: Fractal dimensions, raw sucker rod position, m; Rescaled range statistical analysis.....	- 66 -
Figure 72: Fractal dimensions, raw sucker rod position, m; Rescaled range statistical analysis, log-log power function.....	- 67 -
Figure 73: Fractal dimensions, raw sucker rod load, N; Rescaled range statistical analysis, log-log power function.....	- 67 -
Figure 74: Fractal dimensions, raw sucker rod velocity, m/s; Rescaled range statistical analysis, Hurst exponent.....	- 67 -
Figure 75: Fractal dimensions, raw sucker rod velocity, m/s; Rescaled range statistical analysis, log-log power function.....	- 67 -
Figure 76: Fractal dimensions, raw sucker rod yank, N/s; Rescaled range statistical analysis, Hurst exponent.....	- 68 -
Figure 77: Fractal dimensions, raw sucker rod yank, N/s; Rescaled range statistical analysis, log-log power function.....	- 68 -
Figure 78: From stroke detector, processed sucker rod position, m; Strokes per minute ratio.....	- 68 -
Figure 79: From stroke detector, processed sucker rod yank, N/s; Strokes per minute ratio.....	- 68 -
Figure 80: From stroke detector, raw sucker rod position, m; Fourier transform, root mean square error.....	- 69 -
Figure 81: From stroke detector, raw sucker rod position, m; Fourier transform, strokes per minute.....	- 69 -
Figure 82: From stroke detector, processed sucker rod velocity, m/s; Strokes per minute ratio.....	- 69 -
Figure 83: From stroke detector, processed sucker rod load, N; Strokes per minute ratio.....	- 69 -
Figure 84: From stroke detector, processed sucker rod position, m; Instantaneous phase, strokes per minute.....	- 70 -
Figure 85: From stroke detector, processed sucker rod load, N; Instantaneous phase, strokes per minute.....	- 70 -
Figure 86: From stroke detector, processed sucker rod velocity, m/s; Instantaneous phase, strokes per minute.....	- 70 -
Figure 87: From stroke detector, processed sucker rod yank, N/s; Instantaneous phase, strokes per minute.....	- 70 -

Figure 88: From stroke detector, processed sucker rod position, m; Instantaneous phase, coefficient of performance .....	- 71 -
Figure 89: From stroke detector, processed sucker rod load, N; Instantaneous phase, coefficient of performance .....	- 71 -
Figure 90: From stroke detector, processed sucker rod position, m; Instantaneous phase, root mean square error .....	- 71 -
Figure 91: From stroke detector, processed sucker rod load, N; Instantaneous phase, root mean square error .....	- 71 -
Figure 92: From stroke detector, processed sucker rod velocity, m/s; Instantaneous phase, root mean square error .....	- 72 -
Figure 93: From stroke detector, processed sucker rod yank, N/s; Instantaneous phase, root mean square error .....	- 72 -
Figure 94: From stroke detector, processed sucker rod yank, N/s; Instantaneous phase, coefficient of performance .....	- 72 -
Figure 95: Fractal dimensions, processed sucker rod position, m; Rescaled range statistical analysis, coefficient of performance .....	- 73 -
Figure 96: Fractal dimensions, processed sucker rod position, m; Rescaled range statistical analysis, Hurst exponent .....	- 73 -
Figure 97: Fractal dimensions, processed sucker rod velocity, m/s; Rescaled range statistical analysis, coefficient of performance .....	- 73 -
Figure 98: Fractal dimensions, processed sucker rod load, N; Rescaled range statistical analysis, coefficient of performance .....	- 73 -
Figure 99: Fractal dimensions, processed sucker rod yank, N/s; Rescaled range statistical analysis, Hurst exponent .....	- 74 -
Figure 100: Fractal dimensions, processed sucker rod position, m; Rescaled range statistical analysis, coefficient of performance .....	- 74 -
Figure 101: Fractal dimensions, processed sucker rod velocity, m/s; Rescaled range statistical analysis, Hurst exponent .....	- 74 -
Figure 102: Fractal dimensions, processed sucker rod load, N; Rescaled range statistical analysis, Hurst exponent .....	- 74 -
Figure 103: Full-stroke integral, processed sucker rod position, m; From standardized sucker rod load .....	- 75 -
Figure 104: Full-stroke integral, sucker rod cycle time, s; From standardized sucker rod load .....	- 75 -
Figure 105: Full-stroke integral, processed sucker rod velocity, m/s; From standardized sucker rod load .....	- 75 -
Figure 106: Full-stroke integral, sucker rod cycle time, s; From standardized sucker rod load .....	- 75 -
Figure 107: Full-stroke integral, sucker rod cycle time, s; From processed sucker rod load .....	- 76 -
Figure 108: Full-stroke integral, processed sucker rod velocity, m/s; From processed sucker rod load .....	- 76 -
Figure 109: Full-stroke integral, processed sucker rod position, m; From processed sucker rod load .....	- 76 -
Figure 110: Up-stroke integral, processed sucker rod position, m; From standardized sucker rod load .....	- 76 -
Figure 111: Up-stroke integral, processed sucker rod velocity, m/s; From standardized sucker rod load .....	- 77 -
Figure 112: Up-stroke integral, sucker rod cycle time, s; From standardized sucker rod load .....	- 77 -
Figure 113: Up-stroke integral, processed sucker rod velocity, m/s; From processed sucker rod load .....	- 77 -
Figure 114: Up-stroke integral, sucker rod cycle time, s; From processed sucker rod load .....	- 77 -
Figure 115: Up-stroke integral, processed sucker rod position, m; From processed sucker rod load .....	- 78 -

Figure 116: Down-stroke integral, sucker rod cycle time, s; From processed sucker rod load .....	- 78 -
Figure 117: Down-stroke integral, processed sucker rod position, m; From processed sucker rod load.....	- 78 -
Figure 118: Down-stroke integral, processed sucker rod velocity, m/s; From processed sucker rod load.....	- 78 -
Figure 119: Down-stroke integral, processed sucker rod position, m; From standardized sucker rod load.....	- 79 -
Figure 120: Down-stroke integral, processed sucker rod velocity, m/s; From standardized sucker rod load.....	- 79 -
Figure 121: DC card representation with the first harmonic EFT component .....	- 82 -
Figure 122: Descriptor (0), raw sucker rod position, natural log of amplitude spectrum ...	- 83 -
Figure 123: Descriptor (1), raw sucker rod position, natural log of amplitude spectrum ...	- 83 -
Figure 124: Descriptor (2), raw sucker rod position, natural log of amplitude spectrum ...	- 83 -
Figure 125: Descriptor (3), raw sucker rod position, natural log of amplitude spectrum ...	- 83 -
Figure 126: Descriptor (4), raw sucker rod position, natural log of amplitude spectrum ...	- 83 -
Figure 127: Descriptor (0), raw sucker rod position, unwrapped phase spectrum .....	- 83 -
Figure 128: Descriptor (1), raw sucker rod position, unwrapped phase spectrum .....	- 84 -
Figure 129: Descriptor (2), raw sucker rod position, unwrapped phase spectrum .....	- 84 -
Figure 130: Descriptor (3), raw sucker rod position, unwrapped phase spectrum .....	- 84 -
Figure 131: Descriptor (4), raw sucker rod position, unwrapped phase spectrum .....	- 84 -
Figure 132: Descriptor (0), raw sucker rod load, natural log of amplitude spectrum .....	- 84 -
Figure 133: Descriptor (1), raw sucker rod load, natural log of amplitude spectrum .....	- 84 -
Figure 134: Descriptor (2), raw sucker rod load, natural log of amplitude spectrum .....	- 85 -
Figure 135: Descriptor (3), raw sucker rod load, natural log of amplitude spectrum .....	- 85 -
Figure 136: Descriptor (4), raw sucker rod load, natural log of amplitude spectrum .....	- 85 -
Figure 137: Descriptor (5), raw sucker rod load, natural log of amplitude spectrum .....	- 85 -
Figure 138: Descriptor (6), raw sucker rod load, natural log of amplitude spectrum .....	- 85 -
Figure 139: Descriptor (7), raw sucker rod load, natural log of amplitude spectrum .....	- 85 -
Figure 140: Descriptor (8), raw sucker rod load, natural log of amplitude spectrum .....	- 86 -
Figure 141: Descriptor (9), raw sucker rod load, natural log of amplitude spectrum .....	- 86 -
Figure 142: Descriptor (10), raw sucker rod load, natural log of amplitude spectrum .....	- 86 -
Figure 143: Descriptor (11), raw sucker rod load, natural log of amplitude spectrum .....	- 86 -
Figure 144: Descriptor (12), raw sucker rod load, natural log of amplitude spectrum .....	- 86 -
Figure 145: Descriptor (13), raw sucker rod load, natural log of amplitude spectrum .....	- 86 -
Figure 146: Descriptor (14), raw sucker rod load, natural log of amplitude spectrum .....	- 87 -
Figure 147: Descriptor (15), raw sucker rod load, natural log of amplitude spectrum .....	- 87 -
Figure 148: Descriptor (16), raw sucker rod load, natural log of amplitude spectrum .....	- 87 -
Figure 149: Descriptor (17), raw sucker rod load, natural log of amplitude spectrum .....	- 87 -
Figure 150: Descriptor (18), raw sucker rod load, natural log of amplitude spectrum .....	- 87 -
Figure 151: Descriptor (19), raw sucker rod load, natural log of amplitude spectrum .....	- 87 -
Figure 152: Descriptor (20), raw sucker rod load, natural log of amplitude spectrum .....	- 88 -
Figure 153: Descriptor (21), raw sucker rod load, natural log of amplitude spectrum .....	- 88 -
Figure 154: Descriptor (22), raw sucker rod load, natural log of amplitude spectrum .....	- 88 -
Figure 155: Descriptor (23), raw sucker rod load, natural log of amplitude spectrum .....	- 88 -
Figure 156: Descriptor (24), raw sucker rod load, natural log of amplitude spectrum .....	- 88 -
Figure 157: Descriptor (0), raw sucker rod load, unwrapped phase spectrum .....	- 89 -
Figure 158: Descriptor (1), raw sucker rod load, unwrapped phase spectrum .....	- 89 -
Figure 159: Descriptor (2), raw sucker rod load, unwrapped phase spectrum .....	- 89 -
Figure 160: Descriptor (3), raw sucker rod load, unwrapped phase spectrum .....	- 89 -
Figure 161: Descriptor (4), raw sucker rod load, unwrapped phase spectrum .....	- 89 -
Figure 162: Descriptor (5), raw sucker rod load, unwrapped phase spectrum .....	- 89 -
Figure 163: Descriptor (6), raw sucker rod load, unwrapped phase spectrum .....	- 90 -
Figure 164: Descriptor (7), raw sucker rod load, unwrapped phase spectrum .....	- 90 -
Figure 165: Descriptor (8), raw sucker rod load, unwrapped phase spectrum .....	- 90 -
Figure 166: Descriptor (9), raw sucker rod load, unwrapped phase spectrum .....	- 90 -



Figure 223: Descriptor (18), raw sucker rod load, imaginary part of the spectrum .....	- 100 -
Figure 224: Descriptor (19), raw sucker rod load, imaginary part of the spectrum .....	- 100 -
Figure 225: Descriptor (20), raw sucker rod load, imaginary part of the spectrum .....	- 100 -
Figure 226: Descriptor (21), raw sucker rod load, imaginary part of the spectrum .....	- 100 -
Figure 227: Descriptor (22), raw sucker rod load, imaginary part of the spectrum .....	- 100 -
Figure 228: Descriptor (23), raw sucker rod load, imaginary part of the spectrum .....	- 100 -
Figure 229: Descriptor (24), raw sucker rod load, imaginary part of the spectrum .....	- 101 -
Figure 230: SFS results.....	-103-
Figure 231: Visualization of F1- Score and learning error for the SFS analysis .....	- 104-
Figure 232: SBS results.....	- 105 -
Figure 233: Visualization of F1-Score and learning error for the SBS analysis .....	- 105 -
Figure 234: Performance evaluation of the SBS analysis .....	- 107 -
Figure 235: Performance evaluation of the SFS analysis .....	- 107 -
Figure 236: RSX analysis output .....	- 108 -
Figure 237: Results visualization for the RSX analysis .....	- 109 -
Figure 238: ESX analysis output .....	- 109 -
Figure 239: Results visualization for the ESX analysis .....	- 110 -
Figure 240: MLP architecture .....	- 112 -
Figure 241: An example of bad measurement .....	- 115 -
Figure 242: Surface and downhole dynamometer card example.....	- 116-
Figure 243: Dynamometer cards pump states .....	- 118 -
Figure 244: Identified pump states.....	- 120 -
Figure 245: Every 2 minutes pump states variation in a single well .....	- 125 -
Figure 246: Original data vs resampled data, normal pump state .....	- 126 -
Figure 247: Original data vs resampled data, gas interference pump state.....	- 126 -
Figure 248: Original data vs resampled data, gas interference pump state.....	- 127 -
Figure 249: Original data vs resampled data, normal pump state .....	- 127 -
Figure 250: Original data vs resampled data, fluid pound state .....	- 128 -
Figure 251: Identified trend for the fluid pound state.....	- 130 -
Figure 252: Identified trend for the gas interference state.....	- 130 -
Figure 253: Identified trend for the gas interference + tagging state .....	- 130 -
Figure 254: Identified trend for the normal state .....	- 131 -
Figure 255: An example of forecasted data points.....	- 131 -
Figure 256: Zoomed view of the forecasted interval .....	- 132 -
Figure 257: Identified trend for the original data and the forecasted data .....	- 132 -
Figure 258: Identified trend for resampled and forecasted data .....	- 133 -
Figure 259: Zoomed view of predicted trend vs forecasted values.....	- 133 -
Figure 260: Zoomed view of predicted trendline vs original data trend.....	- 133 -

**List of Tables**

Table 1: Confusion matrix for the validation set of the best performing network..... - 113 -  
Table 2: Model performance evaluation..... - 113 -  
Table 3: Confusion matrix for the validation set of the best performing network..... - 121 -  
Table 4: Model performance evaluation..... - 121 -

## List of Abbreviations

ACF	Autocorrelation Function
AE	Auto Encoder
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AR	Autoregression Model
ARIMA	Autoregressive Integrated Moving Average
BM	Boltzmann Machine
d	Down-stroke Integral
DBN	Deep Belief Network
DC	Dynamometer Card
DCN	Deep Convolutional Network
DFF	Deep Feed-Forward Networks
dft	Fourier Transform Descriptor
DN	Deconvolutional Neural Networks
EFT	Elliptic Fourier Transform
ent	entropy
EPT	Effective Plunger Travel
err	Root Mean Square Error
ESX	Exhaustive Search Extension
exp	Hurst exponent
f	Full-stroke Integral
fd	Fractal Dimensions
FFN	Feed-Forward Neural Networks
F <sub>sP</sub>	Full-stroke integral, processed sucker rod position
F <sub>sT</sub>	Full-stroke integral, sucker rod cycle time
F <sub>sV</sub>	Full-stroke integral, processed sucker rod velocity
FT	Fourier Transform
GDB	Gewinnungsdatenbank
gex	Normalized absolute differences at the end of the stroke
GRU	Gated Recurrent Unit
gsx	Normalized absolute differences in the whole stroke
HN	Hopfield Network
Im	Imaginary Part of the Spectrum
IP	Instantaneous Phase
kurt	Kurtosis
Lacc	Learning subset Accuracy
Lent	Learning subset Entropy
Lerr	Learning Error
InA	Log of Amplitude Spectrum
L <sub>p</sub>	Processed Sucker Rod Load
L <sub>r</sub>	Raw Sucker Rod Load
L <sub>r</sub>	Raw Sucker Rod Load
LSTM	Long/Short Term Memory
MA	Moving Average Model
MARIMA	Multivariate Autoregressive Integrated Moving Average
MC	Markov Chain
med	Median



ML	Machine Learning
MLP	Multilayer Perceptron
MPT	Maximum Plunger Travel
MSTL	Multiple Seasonal Decomposition
MVD	Measured Vertical Depth
PACF	Partial Autocorrelation Function
ph	Unwrapped Phase Spectrum
Pp	Processed Sucker Rod Position
Pr	Raw Sucker Rod Position
Pr	Raw Sucker Rod Position
R2	Log-log power function
Re	Real Part of the Spectrum
RNN	Recurrent Neural Networks
RS	Rescaled range statistical analysis
RSX	Random Search Extension
SBS	Sequential Backward Selection
sd	Stroke Detector
sf	Statistical Features
SFS	Sequential Forward Selection
si	Stroke Integral
skew	Skewness
smpr	Strokes per Minute Ratio
spm	Strokes per Minute
SRP	Sucker Rod Pump
STL	Seasonal and Trend Decomposition
Terr	Testing Error
TVD	True Vertical Depth
Verr	Validation Error
Vp	Processed Sucker Rod Velocity
Vr	Raw Sucker Rod Velocity
Yp	Processed Sucker Rod Yank
Yr	Raw Sucker Rod Yank

# 1 Introduction

## 1.1 Research Problem and Motivation

We are living in a world that is changing at a very fast pace. Digital technology development is affecting every industry. Oil and gas companies are not an exception, especially considering these challenging times of transition faced by the energy market at the moment. A safer, more sustainable, and cost-efficient production has become the main drivers. In order to ensure maximum production levels and minimum down time, newly available technology is implemented in daily operations. The areas of application for such advanced methods are nearly infinite. Some of the most popular purposes are effectively identifying and predicting failures, conducting remote operations, and improving decision making processes overall.

The result of digitalizing operations is an ever-increasing amount of data that is being continuously created. Thanks to technological advancements the challenge of collecting and storing such large amounts of data can be easily overcome. Therefore, oil and gas companies have been using several sources to generate data such as sensors installed at different points in the entire production system and accompanying different operations. Data comes in different formats, at varying time intervals, and with diverse complexity levels. Gathered data represents valuable information that can be used in endless ways for improving daily operations and bringing business worth.

Nowadays in the world, there are around 2 million active oil wells. In more than half of these wells, the reservoir pressure is not high enough in order to produce fluids to the surface. As a result, several artificial lift systems are being used. Due to its lower installation costs and longer lifetime, sucker rod pumping (SRP) has become the most widely used artificial lift system, being present in more than 750,000 wells (PetroWiki), (Fortune Business Insights 2019), therefore SRPs represent an inexhaustible source for data generation and application. Previous research for the artificial lift segment has shown successful applications for ESP optimization by combining an expert system for analyzing sensor data (McLean et al. 1998). Others have developed a model for automated dynamometer card reading for SRP (Nazi et al. 1994), detecting liquid loading (Bouw 2017), and wax precipitation (Adeyemi, Sulaimon 2012). However, despite a rapid development in data-based models for SRP, little efforts have been input for identifying the trend in pumps behavior.

With every cycle, sucker rods are lifting big loads, which put the string under stress at both upstroke and downstroke. Additional factors related to the environment like corrosion, scale, wax depositions are contributing to wear and lead to abnormal functioning or even failure. In

order to prevent such failures, reduce maintenance operations and costs and become more sustainable, predicting the failures in SRPs would be of great use for operating companies from both technical and economical points of view.

This main purpose of this work is to develop a novel approach on using SRP data in combination with machine learning algorithms to illustrate the benefits of applying data analytics on the immense amount of data an oil and gas company can produce, how good is the quality of generated information and how it can be applied in order to benefit the company's success by predicting the tendency of SRP performance.

## **1.2 Outline**

Chapter 2 is dedicated to literature review, describing the general concepts of machine learning and incorporates research outcomes related to the topic. Different types of learning models and artificial neural networks are explained, as well as all the steps involved starting from data acquisition to building models and evaluating their performance. The importance and methods for analyzing time series data and its advantages are also described. Followed by examples of application in several industries with a larger focus on the oil and gas sector.

Chapter 3 lists the possibilities of applying machine learning in oil and gas industry, as a general concept and per sector, as well as in other industries.

Chapter 4 refers to data acquisition. It describes the sources from where the data comes from, in which formats and types. All the ways needed to assess, manage, and integrate received information are explained in detail.

Chapter 5 presents one of the most challenging parts of this work, data visualization. It is illustrated how challenging it can be to manage a large amount of data. An overview of the measurements frequency is illustrated and how judging solely on this one can already extract some valuable messages regarding pump's operation.

Chapter 6 describes the features that have been calculated in order to describe every measurement and further use as input for training artificial neural networks. All categories of features are described, challenges encountered in the process of selection, calculation and evaluation are also included.

In chapter 7 the models built for identifying measurements quality are presented. An overview of trained models, including their architecture, data preparation, results evaluations are given. The criteria for the best performing model is described followed by its application. Finally, the outcome is illustrated, and results are discussed.

Similar to chapter 7, chapter 8 is dedicated to trained artificial neural network models but this time they are used for identifying normal and abnormal behaviors in SRPs. All scenarios are presented, causes, challenges and results are discussed, as well as further application for trend analysis.

Chapter 9 incorporates the methodology used for the trend identification and forecasting. ARIMA model used for trend analysis is explained in detail. Main challenges related to data frequency and applied solutions are addressed. Finally, examples of identified trends and forecasted values are illustrated.

All results, challenges and findings are summarized and described in chapter 10 which includes conclusions and recommendations.

Sources, references and appendices can be found at the end of this report.

## **2 Literature Review**

Within the last years, the fast advancement of information technology and data availability have stimulated the development of artificial intelligence (AI). Data analysis is being used more and more in diverse fields and for multiple applications. Examples of machine learning (ML) are everywhere around us. Starting from voice assistants in our smart devices, advertisements recommendations, intelligent vacuum cleaners, self-driving cars, up to health care system, such as image recognition for identifying tumours.

Considering the ever-increasing amount of data, accessibility and affordability of computing power, data science will keep on creating smarter algorithms and machine learning will bring more efficiency in our daily lives.

In this chapter, general concepts of machine learning, the importance and evolution of intelligence algorithms will be described. Most relevant topics for understanding the concepts and findings of this research are covered, followed by examples of application.

### **2.1 Machine Learning**

“Machine learning is a branch of AI focused on building applications that learn from data and improve their accuracy over time without being programmed to do so.” (IBM Cloud Education)

In other words, algorithms are trained to identify data patterns and features which will be further used for decision making processes and predictions based on used data. The larger the data sets used, the better and more precise the algorithm will be therefore the accuracy of decisions and predictions will be higher.

Building a ML application requires a few main steps as following:

1. Preparing the data set
2. Selecting the algorithm
3. Training the algorithm
4. Applying the model

#### **2.1.1 Preparing the Data Set**

A training subset of data has to be prepared. This should represent the entire data set and will be used as input for the ML model used to solve the proposed problem. Often, the training set consists of labelled data used for feature extraction and classifications that will have to be identified by the model. Sometimes the data is unlabelled, in such cases the model will extract the features and make the classifications on its own.

In any case, the training subset has to be prepared correctly, to make sure that it does not contain any imbalances, duplicates or other characteristics that could negatively impact the training process. The training subset is then divided into two parts, one used for the training procedure itself and second for the validation, used for verification and improvement of the trained model.

### **2.1.2 Selecting the Algorithm**

Selecting a certain type of algorithm depends on the type of data available and the problem that has to be solved.

There is a variety of algorithms available such as:

**Regression algorithms** – Linear and logistic regression are used to understand the relationships in the data.

**Decision trees** – used for classified data for the purpose of making a decision based on a predefined set of rules.

**Instance based algorithms** – used to identify the likelihood of a data point belonging to a certain group based on how close this data point to a member of that group is.

**Clustering algorithms** – identifies groups of data points that have similar characteristics and tags the rest of variables based on their similarity to identified groups.

**Association algorithms** – based on association rules, the algorithm finds patterns and relationships in data.

**Artificial neural networks** – these algorithms contain a network of layers, input, hidden, and output. The input layer contains the training subset, hidden layers perform the calculations and come up with multiple conclusions on the input, which are further assigned a probability in the output layer. The results in this work were achieved by using artificial neural networks. The next sub chapter provides more details on this.

### **2.1.3 Training the Neural Network**

Training the neural network is a process which requires multiple iterations. The variables are run through the algorithm, then the resulted output is compared to the expected one, weights are further adjusted, variables are rerun through the algorithm and the returned result should be more accurate. The trained and accurate algorithm obtained represents the final model.

### **2.1.4 Applying the Algorithm**

Last step is to apply the obtained model with new data. In the best-case scenario, the model should improve its accuracy and performance over time. This will depend on the source of the data and the nature of the problem to be solved.

## **2.2 Machine Learning Styles**

There are three main categories of machine learning styles:

- Supervised machine learning
- Unsupervised machine learning
- Semi-supervised machine learning

### **2.2.1 Supervised Learning**

Supervised learning uses labelled data to train a function that can be extended to new examples. In other words, the model consists of an input and a desired output. The supervision is exerted upon the wanted output, which as a result is used to adjust model's parameters in order to improve the actual output. (Jones 2018)

The advantage of supervised learning is the requirement for less training data. This makes the training process faster and easier as results can be compared to actual labelled data points.

### **2.2.2 Unsupervised Learning**

Algorithms based on unsupervised learning use as input unlabelled data which is further segregated based on some hidden features. Due to the lack of labels, evaluating the result is the main challenge. By grouping the data using unsupervised learning, some patterns present in the raw data could be identified that otherwise would not be visible. (Jones 2017)

Unsupervised learning is focusing more on identifying patterns and relationships in the data rather than on providing automatized decisions and predictions.

### **2.2.3 Semi-supervised Learning**

Semi-supervised learning is a middle course between supervised and unsupervised learning. For the training process, a smaller labelled data set is used as a guide for the classification process whereas features are extracted from a larger, unlabelled data set. This type of learning style is convenient in the situation when there is not enough labelled data available for training as compared to supervised learning. (IBM Cloud Education)

## **2.2.4 Reinforcement Learning**

Reinforcement machine learning is similar to supervised learning, but the algorithm is not trained using sample data sets. It learns as it goes by the means of trial and error. Successful outcomes will be reinforced to provide the best decision or recommendation for the given problem.

## **2.3 Artificial Neural Networks**

The foundation of Artificial Neural Networks (ANN) has been laid in 1943 by two scientists, McCulloch and Pitts, who created a model based on two approaches. One focused on biological processes and the other on the application of neural networks to artificial intelligence. Later on, Rosenblatt created the perceptron, and Minsky with Papert showed the key issues of perceptrons. These were unable to process the exclusive-or circuit and handle effectively the work required by neural networks. This has stopped the evolution of machine learning until 1980, when Hopfield and Kohonen revived the use of ANN. Since then they have been largely used in many fields of application such as image and speech recognition, economic forecasts and predictions, etc. (Wikipedia)

A simple definition of ANN is given by Haykin and describes it as “a massively parallel combination of simple processing unit which can acquire knowledge from the environment through a learning process and store the knowledge in its connections.” (Guresen, Kayakutlu Gulgun 2011)

### **2.3.1 Artificial Neural Network Components**

The main components of ANN are neurons, multiple inputs, activation functions, connection weights, and biases. The architecture of neural networks is determined by the layout of neurons in layers and the connections between them. A typical composition contains three layers as input, hidden, and output. See an example in Figure 1 below.



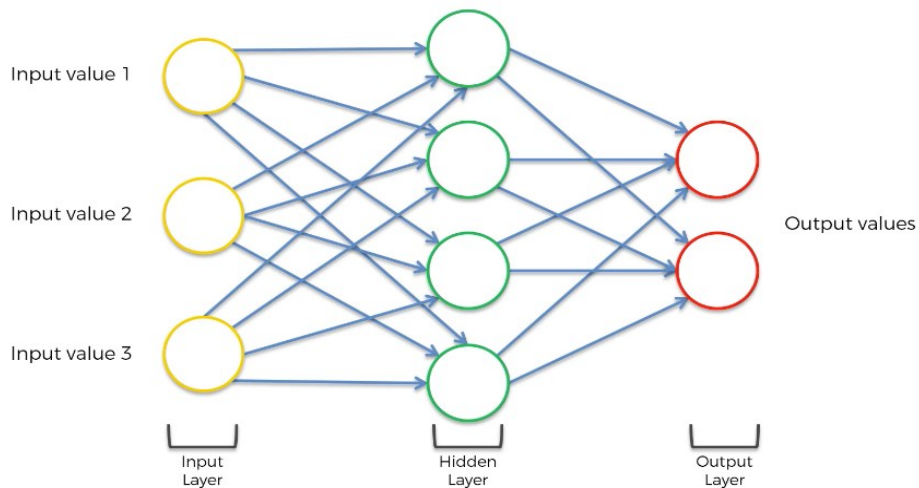


Figure 1: Artificial Neural Network Components (McCullum 2020)

### 2.3.1.1 Neurons

A single neuron represents the building block of a neural network. Figure 2 illustrates the structure of a neuron with a single input.  $X$  represents the input to the neuron;  $w$  is the weight associated with it. Weight is the essential parameter here as it is controlled by the model in order to receive a better fit for the output. When an input is passed into the neuron, it gets multiplied by the weight ( $X * w$ ). (Krishnamurthy 2021)

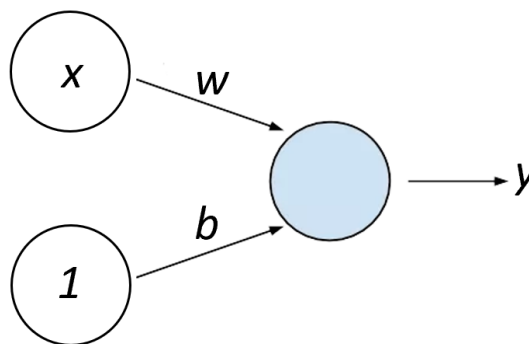


Figure 2: Single neuron structure

Bias represents the second element of the input. As it is represented in Figure 2, it is determined solely by value  $b$  as the value of the node is 1. Bias represents the unpredictable part of the model which helps to generalize it and offer the flexibility necessary for adapting to multiple unseen inputs when testing data is used.

Output occurs as a result of combining the bias and the input:  $w * x + b = y$ . This formula resembles the equation of a straight-line  $y = m * x + c$ . It is because ANN are composed of multiple interconnected neurons, each of which runs its own regression.

### 2.3.1.2 Multiple Inputs

In the real world, datasets to be analysed are complex and such a simple structure as illustrated in Figure 2 **Error! Reference source not found.** cannot be used. A combination of multiple inputs will be used to estimate the output. However, the principle is similar to the neuron with a single input.

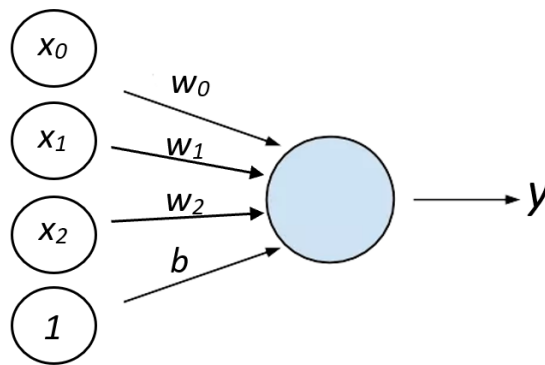


Figure 3: Neuron with multiple inputs

The formula for a network with multiple inputs illustrated in Figure 3 will be then:

$$x_0 * w_0 + x_1 * w_1 + x_2 * w_2 + b = y$$

1

### 2.3.1.3 Layers

Neurons in a neural network are organized into layers. A layer where every neuron has a connection to every other neuron in its next layer is called a dense layer. (Krishnamurthy 2021)

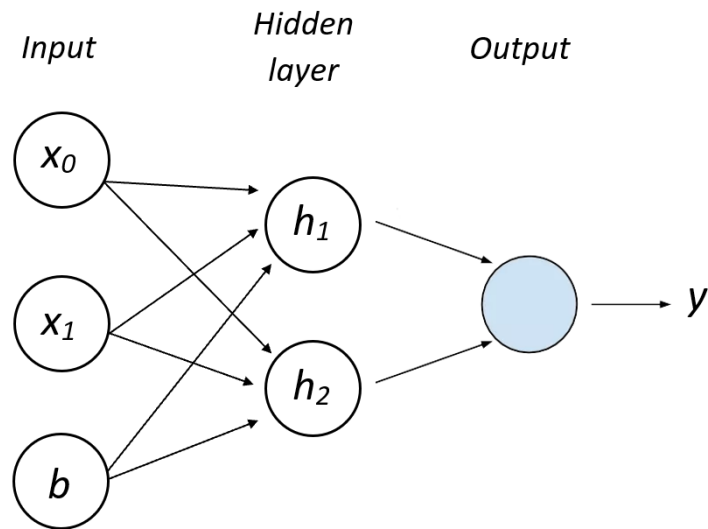


Figure 4: ANN with one hidden layer

With increasing complexity, ANNs are able to transform data and infer relationships in several complex ways. The more layers and nodes are added to the network, more complex it becomes. See Figure 4 for an example.

### 2.3.1.4 Activation Function

Previously discussed architecture can be used for predicting linear relationships. In the situation of a more complex architecture there are two options. One is to add more layers to the network between input and output layers, the so-called hidden layers. Every hidden layer has a pre-set number of nodes which will add complexity to the network and will separate it from the regression counterpart.

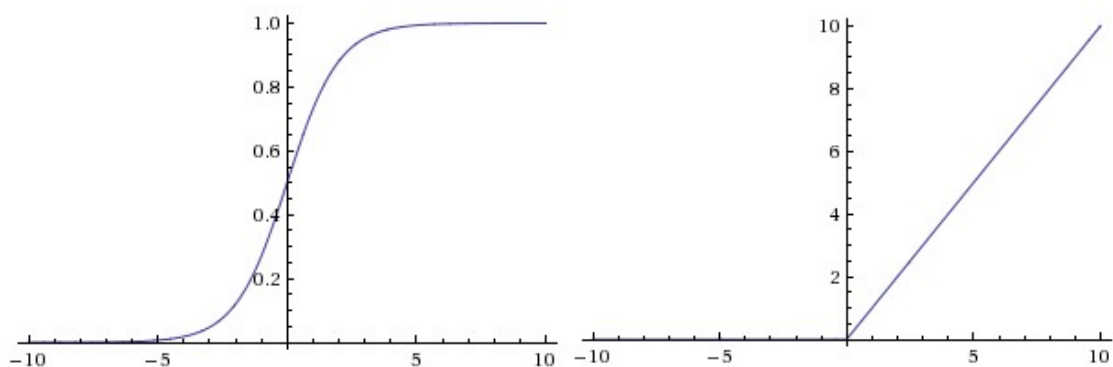


Figure 5: Activation functions (left – Sigmoid, right – ReLu)

Another way of adding complexity to a neural network is by introducing an activation function at every node that is not an input or output. An activation function is a function that transforms the input data using a non-linear method. Some of the most widely used are ReLu and Sigmoid function. See Figure 5.

$$\text{Sigmoid } f(x) = \frac{1}{1 + e^{-x}}$$

2

$$\text{ReLU } f(x) = \max(0, x)$$

3

Both models are nonlinear and as a result an extra element of adaptability is added to the model. The network will be able to predict classes that do not have linear decision boundaries or approximate nonlinear functions.

A neuron's role in a hidden layer is to pass the sum of the products of the inputs and their weights into an activation function. (Krishnamurthy 2021) The resulted value will be passed as input to the next neuron, it can be a hidden neuron or the output. See Figure 6 as an example.

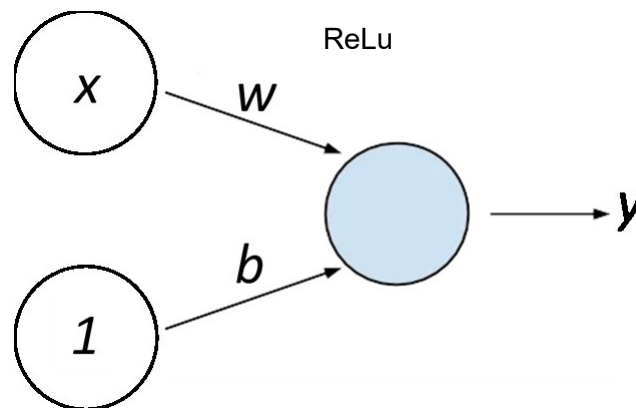


Figure 6: ANN with ReLU activation function

### 2.3.1.5 Weights Optimization

Weights are randomly assigned in a neural network when it is initialised. A neural network has control over the data by adjusting its weights. This is an iterative process, and it takes place as many times as necessary until predictions are accurate enough or another stopping criterion has been reached.

The purpose of a Loss function is to assess the exactitude of predictions. It is a function that compares the model output with the actual output and evaluates how accurate the model estimations are. A common metric for a loss function is the mean absolute error. It is a measure of the sum of absolute vertical differences between the estimated values and the actual values, see Figure 7 (Pascual 2018)

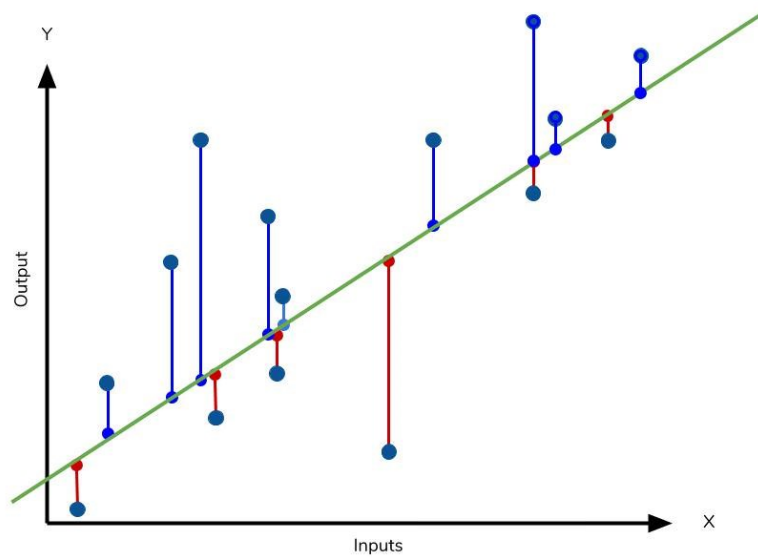


Figure 7: Mean Absolute Error

In order to find the best set of weights, an optimisation method called stochastic gradient descent is used. For every epoch, the stochastic gradient will go through the following steps:

1. Initialize a value for the weights
2. Update weights until loss function is reduced
3. Stop when the minimum error has been reached

Gradient Descent needs a differentiable algorithm because when looking for the minimum value, the gradient of the current position is calculated and then it is decided which direction to take in order to reach gradient 0. It is known that the minimum point on the curve is the point at which the error gradient is 0. See Figure 8.

Gradient Descent Algorithm

Repeat until convergence:

$$\omega_j = \omega_j - \alpha \frac{\delta}{\delta \omega_j} J(\omega) \quad \text{for } j = 1 \dots k$$

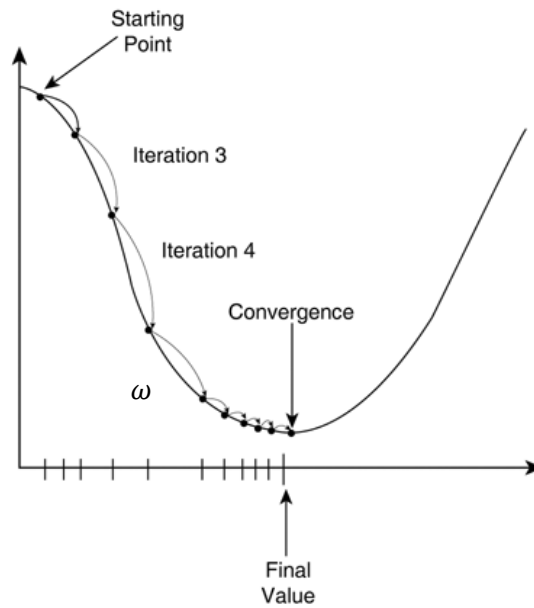


Figure 8: Gradient descent

During the iterative process, the algorithm takes the current weight and subtracts it from the loss function and multiplies it by a learning rate, which size determines how quickly is converge or divergence from the minimum value reached.

### 2.3.1.6 Fitting the Model

There are two important concepts for ML known as overfitting and underfitting. They offer an understanding about the capability of a ML algorithm and help see whether it is capable of serving its initial purpose or not.

From a mathematical point of view, overfitting represents the situation when the accuracy of the training subset is greater than the accuracy of the testing subset. Underfitting is simply the low performance of both training and testing data.

In the situation of overfitting, the model does not generalise well to unseen data. The observations are solely built on observed behaviour of the training dataset and it could not find the more complex relationships it was supposed to look for. As a result, the model cannot be used for predictions as it is not capable of adjusting to new data or different datasets.

Underfitting carries the opposite situation. As in the case of overfitting, the model was not able to find complex relationships, the difference is that the model has generated loose rules and it is not attached to any data or any concrete rules. Such a model will perform poorly on training data as it cannot see any relationships between the variables.

A way of avoiding underfitting is by adding extra layers, neurons or features to the network in order to add more complexity and increase the training time. Meanwhile, overfitting can be

prevented by diminishing the intricacy of the model and stopping the training process earlier to prevent over-analysing the data.

## 2.3.2 Artificial Neural Networks Types

There are two multiple types of neural networks based on their complexity and purposes. The most used models and their applications are going to be described in the following section.

### 2.3.2.1 Perceptron

A perceptron is a single layer neural network consisting of one input and one output layer. It has no hidden layers. The input is taken, and weighted input is calculated for each node. Further, an activation function is used for classification.

Perceptrons are used for solving simple problems such as classification, encode database, monitor access data. A typical structure is illustrated in Figure 9.

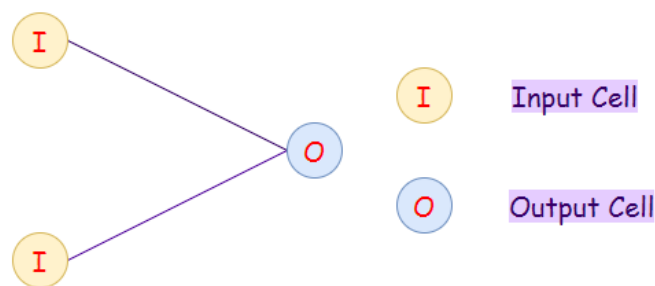


Figure 9: Perceptron architecture (Shukla and Iriondo 2020)

### 2.3.2.2 Feed-Forward Neural Networks (FFN)

In feed-forward neural networks nodes do not form a circle, all perceptrons are arranged in layers. The input layer simply takes in input and the output layer generates the output. Every perceptron from one layer is connected to the nodes in the next layers, as a result all nodes are fully connected. In a feed-forward neural network there are no back-loops, therefore, in order to minimize errors a backpropagation algorithm is generally used for updating weights.

Main applications for feed forward neural networks are pattern recognition, computer vision, speech recognition, etc. A typical example is illustrated in Figure 10 below.

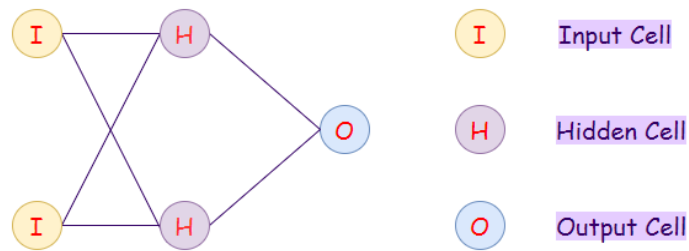


Figure 10: Feed-Forward neural network architecture

### 2.3.2.3 Radial Basis Networks

Compared to other neural networks, radial basis networks have a faster learning rate and universal approximation, therefore they are mainly used for approximation problems. As compared to feed-forward networks, radial basis ones use a radial basis function for activation. A sigmoid function that gives an output of 0 and 1, in order to identify if the answer is yes or no. They are useful when the data contains continuous values.

### 2.3.2.4 Deep Feed-Forward Networks (DFF)

This is basically a feed-forward network with multiple hidden layers, as illustrated in Figure 11. The biggest issue associated to using one hidden layer is overfitting. Therefore, by increasing the number of hidden layers, the chance of overfitting is diminished, and the generalization is improved. Most common applications for deep feed-forward neural networks are computer vision, financial predictions, pattern recognition, noise filtering, etc.

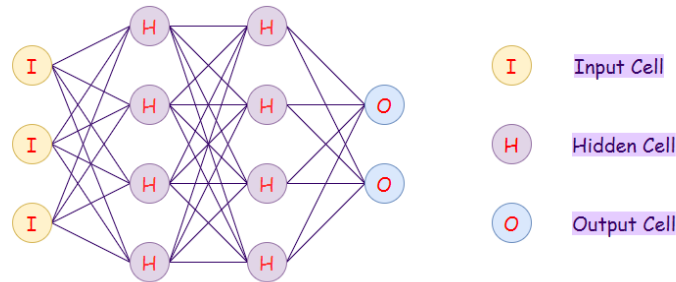


Figure 11: Deep Feed-Forward network architecture

### 2.3.2.5 Recurrent Neural Networks (RNN)

In recurrent neural networks, connections between units form a directed circle. This allows the access to previous information in current iterations. This is useful in situations like trying to predict the next word in a sentence, for that is important to know what the previously used words were. In order to achieve that, in the hidden layers every single neuron obtains inputs with a delay in time. The advantages of RNNs are the capability of processing inputs and sharing any lengths and weights over time, without increasing the size of the model. On the other side, computational time for this type of network is slow, additionally, it doesn't consider



any future input for the current state because it is not able to remember information from a long time ago. A typical architecture of RNNs is shown in **Error! Reference source not found.**

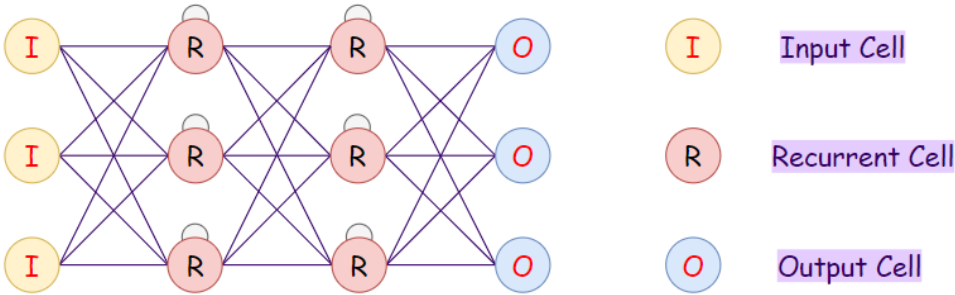


Figure 12: Recurrent neural network architecture

Typical uses for recurrent neural networks are robot control, time series prediction, time series anomaly detection, rhythm learning, music composition.

**2.3.2.6 Long / Short Term Memory (LSTM)**

It was previously mentioned that RNNs contain a time delay and it may fail when the number of data is large, and the goal is to identify most relevant values. In this case LSTM are the option to go for. These networks introduce a memory cell that aids with processing data with memory gaps and remember data from long time ago. LSTMs are usually used in speech and writing recognition. See Figure 13 for an example.

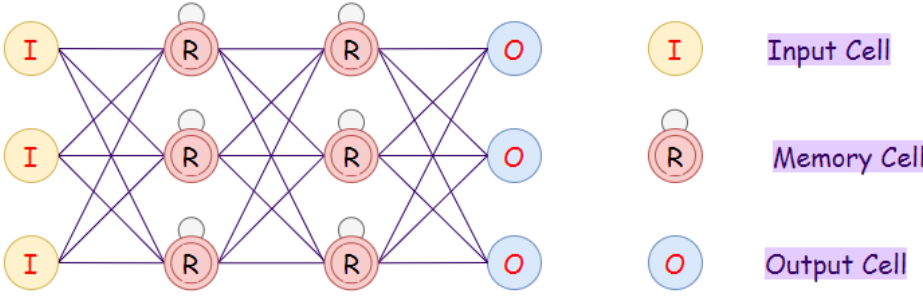


Figure 13: Long/Short Term Memory network architecture

### 2.3.2.7 Gated Recurrent Unit (GRU)

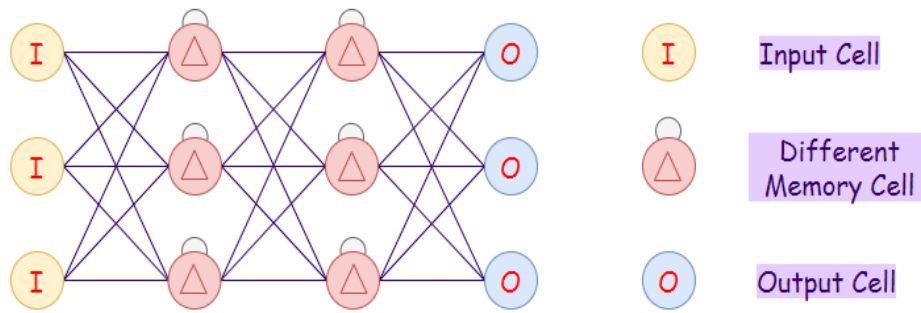


Figure 14: Gated Recurrent Unit architecture

As it can be seen in Figure 14, GRUs are similar to LSTMs as they both have similar structure, therefore produce similar results. As the name implies, they are composed of two main gates. Update gate regulates how much of the former learning needs to be passed to the future and reset gate determines how much knowledge has to be forgotten and current memory gate. It is usually used in modelling processes such as speech modelling or music modelling, as well as in language processing.

### 2.3.2.8 Auto Encoder (AE)

Auto encoders are unsupervised machine learning algorithms. The number of input cells overcomes the number of hidden cells. AEs are usually trained to display an output as close as possible to the input, based on identified common patterns. This is a relatively simple algorithm with an output identical to the input. Most often used for classification, clustering and feature compression. See Figure 15.

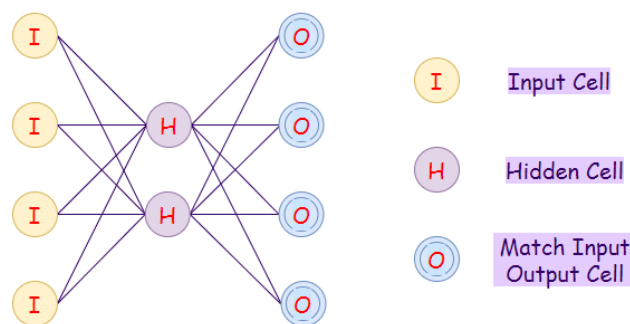


Figure 15: Auto Encoder architecture

### 2.3.2.9 Markov Chain (MC)

Markov Chains are mathematical systems that transform from one condition to another contingent on some probabilistic rules. The transitioning probability to a state depends on the

current state and the time elapsed. Some examples of possible states are letter, numbers, weather conditions, etc. MCs are applied for statistics, speech recognition, information and communication systems. See Figure 16 for an example.

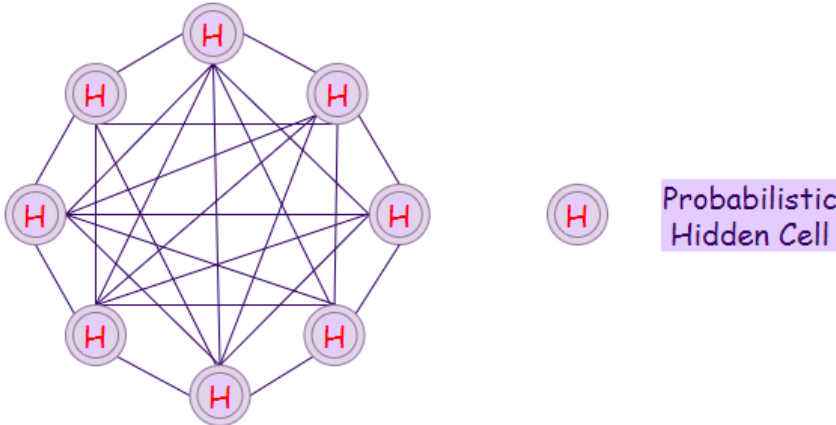


Figure 16: Markov Chain architecture

**2.3.2.10 Hopfield Network (HN)**

In Hopfield neural networks, every neuron is connected with another directly (see Figure 17). In such networks, neurons are either on or off. Their state changes based on the inputs they receive from other neurons. Usually, HNs are used for pattern storage and memories. They perform well in recognizing patterns even if the input is distorted or incomplete. Therefore, HNs are often used for image detection and recognition, especially in medicine, enhancing X-Ray images.

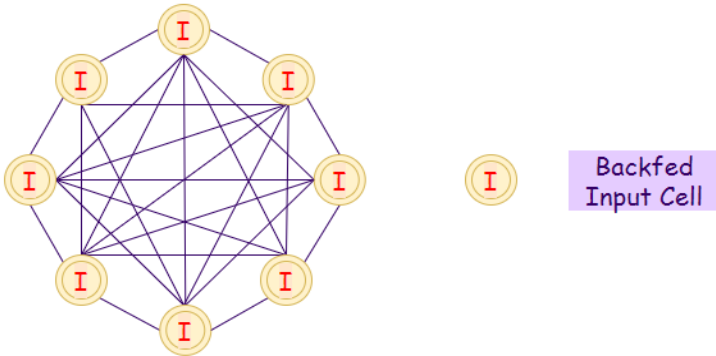


Figure 17: Hopfield Network architecture

**2.3.2.11 Boltzmann Machine (BM)**

The usual function of a Boltzmann machine is learning a probability distribution from an original dataset and use it for reasoning on unseen data. BMs are composed of input and hidden nodes. Whenever hidden nodes change their state, input nodes change into output nodes. BMs are used for classification, regression, dimensionality reduction, etc. See Figure 18.

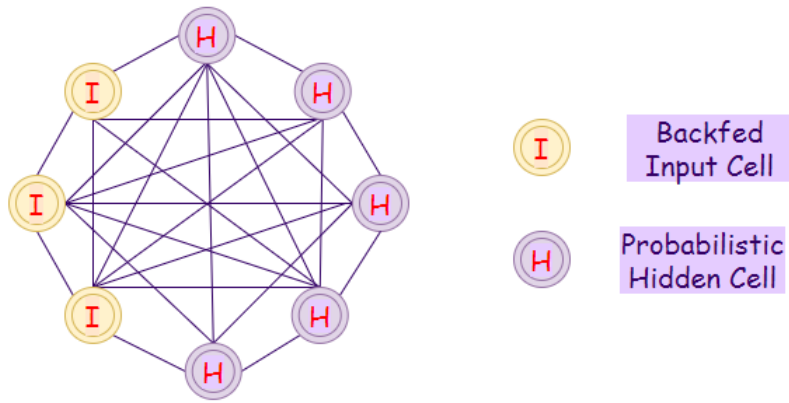


Figure 18: Boltzmann Machine architecture

### 2.3.2.12 Deep Belief Network (DBN)

A deep belief network contains multiple hidden layers and is used with an unsupervised algorithm. Features are detected by layers, then after unsupervised learning, models are trained with supervision methods to perform classification. See Figure 19.

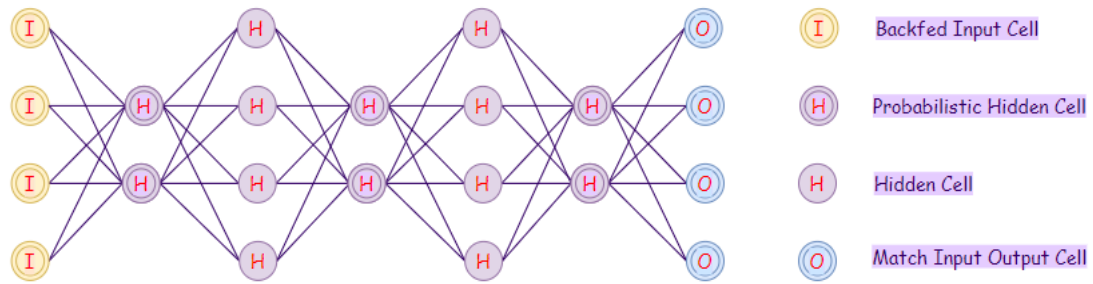


Figure 19: Deep Belief network architecture

### 2.3.2.13 Deep Convolutional Network (DCN)

Convolutional neural networks are mainly used for image classification, image clustering and object recognition. They enable unsupervised construction of hierarchical image representations. DCNs add more complex features, therefore resulted accuracies are higher. Usually they are used for image recognition, video analysis, time series forecasting, identification of face, street signs, tumours, etc. See Figure 20.

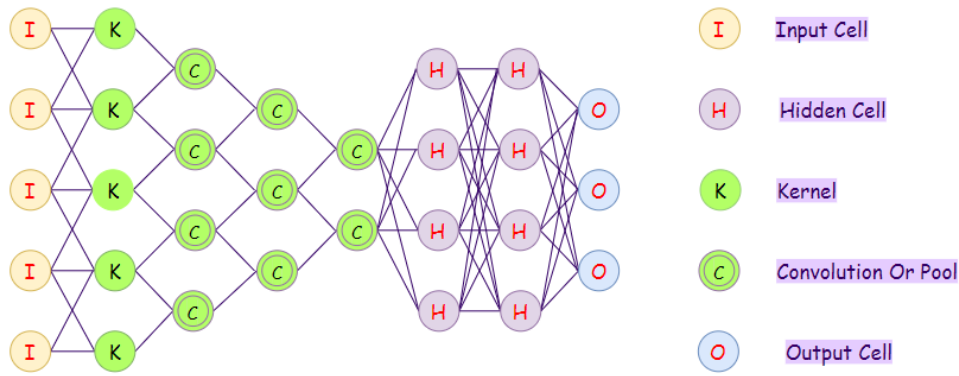


Figure 20: Deep Convolutional network architecture

### 2.3.2.14 Deconvolutional Neural Networks (DN)

These are convolutional neural networks that work in reverse and have a different application. Deconvolutional networks are used for identifying lost features or signals. A DN can make pictures out of vectors and are applied for surface depth estimation from an image for example or image super-resolution. See Figure 21.

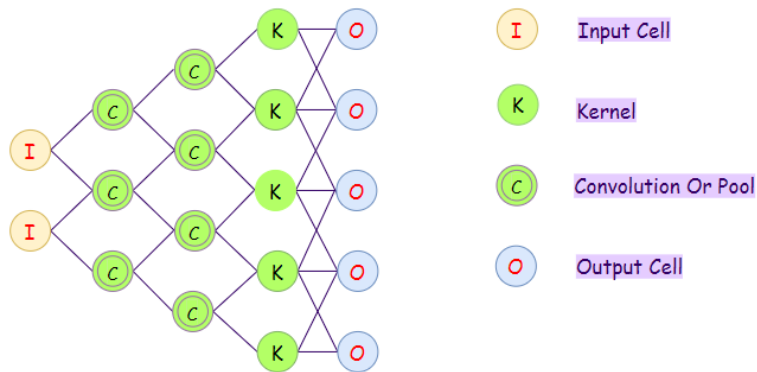


Figure 21: Deconvolutional neural network architecture

## 2.4 Trend identification in Time Series Data

Time series data represents a sequence of numerical values organized in a chronological order, often with fixed interval points in time. Such a scenario allows for an accurate prediction and forecast of future values. (Anish 2020)

Typically, time series data shows seasonal patterns, trends and other relations to external data.

In real life situations, time series as used for weather reports, earthquake prediction, finance, and many other scientific and engineering fields.

### **2.4.1 Time Series Forecasting**

Based on previous observations, time series forecasting is used to model future values. Usually, models rich in historical data are used for prediction. It is worth mentioning that time series forecast is not an exact prediction of the future but more of an estimation of what should we expect based on what happened.

There are two types of forecasting available:

- Qualitative forecasting, considered when there is no access to historical data, therefore it is highly objective
- Quantitative forecasting, used when there is a significant amount of historical data available, therefore considered more efficient

### **2.4.2 Time Series Forecasting Application**

There are two important directions for using time series forecasting, as follows:

1. To obtain and understand the factors behind produced data
2. To fit a model and use it for forecasting

In order to better understand time series datasets, it is useful to split in 4 parts: level, trend, seasonality and noise.

Level is the base value for series if it were a straight line.

Trend is linearly increasing or decreasing behaviour of series over time.

Seasonality shows patterns or cycles over time.

Noise is the variability in observations that cannot be explained by the model.

Most of the time, series have level and noise, while trend and seasonality are not mandatory

### **2.4.3 Time Series Trend Analysis**

Looking for repeated behaviour in the graphical representation of data is known as trend analysis. If the trend is constantly increasing or decreasing, then analysing it is not that difficult. However, when data shows errors, then the first step would be to smoothen in before proceeding to trend identification.

Smoothing calls for some form of local averaging of data. Most common technique is the moving average smoothing, where every element in time series is replaced with a weighted average of surrounding elements.

Fitting a function is the next step and it relates to transforming data or removing nonlinearity, should there be a nonlinear component present. Most common functions used are log, exponential, or polynomial. If we look in the example in Figure 22 below, it is clear there is an upward trend.

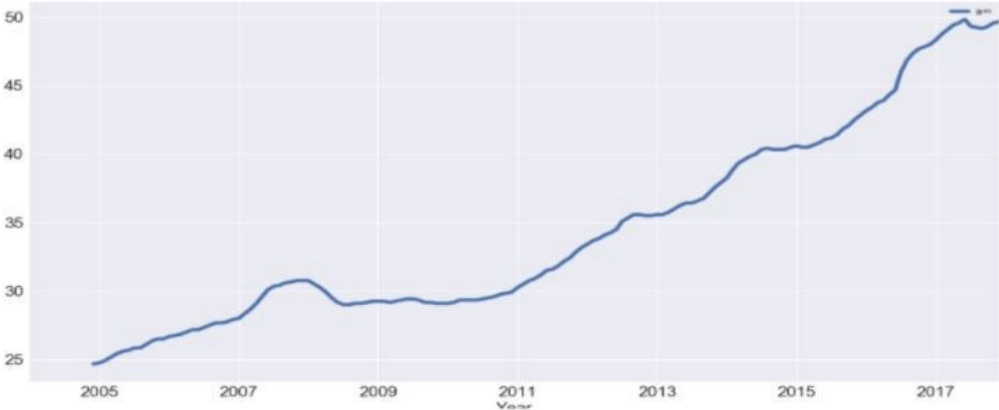


Figure 22: Example of upward trend

### 2.4.4 Time Series Seasonality Analysis

Seasonality is data reoccurrence at a determined time interval. For example, people tend to go on vacation between May and August – this is seasonality. It can be obtained by measuring the autocorrelation that follows the trend deduction from the data.

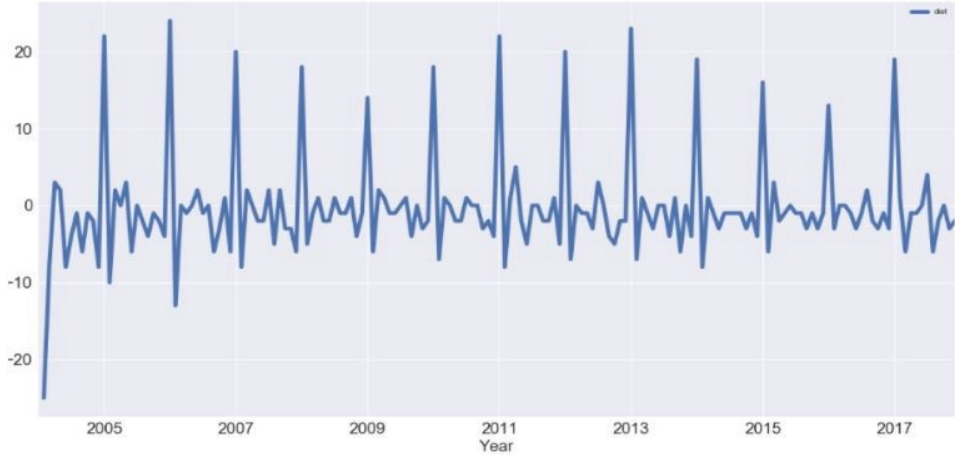


Figure 23: Seasonality example

Figure 23 above illustrates times of the year when people tend to go on a diet. It is easy to notice the spike in the beginning of every year. This means that yearly, in January, people go on diets as a resolution, compared to other months.

### 2.5 Moving Average Models

Autoregression Model (AR)

AR is a time series model which uses previous observations as input to a regression equation in order to predict the value at the next time step. The equation for a regression takes the form of:

$$\hat{y} = b_0 + (b_1 * x_1)$$

5

This method can be applied for time series where input variables are taken as observations at previous time steps, called lag variables. (Anish 2020) This would look like:

$$x_{t+1} = b_0 + (b_1 * x_t) + (b_2 * x_{t-1})$$

6

The term of auto regression comes from the fact that the regression model uses data from the same input variable at previous time steps. (Anish 2020)

### **2.5.1 Moving Average Model (MA)**

Residual errors from forecasts in a time series represent an additional source of information which can be modelled. Such an autoregression model can be used to anticipate any forecast errors, which then can be used to correct forecasts.

Trend, bias & seasonality found in the residual error can be modelled directly. It is possible to create a model of the residual error time series and predict the expected error of the model. The predicted error can then be subtracted from the model prediction & in turn provide an additional improvement in performance.

Moving Average Model is an autoregression of the residual error.

### **2.5.2 Autoregressive Integrated Moving Average (ARIMA)**

ARIMA is a forecasting technique that calculates future values entirely based on its inertia.

Autoregressive Integrated Moving Average (ARIMA) models include a clear-cut statistical model for the asymmetrical component of a time series that allows for non-zero autocorrelations in the irregular component. (Anish 2020)

ARIMA models work only for stationary time series. Therefore, if a time series is non-stationary, it first needs to be 'differenced' until stationarity is obtained

More about ARIMA can be found in Chapter 9.

### **2.5.3 ACF and PACF**

The correlation for time-series observations can be calculated with observations from previous time steps, called *lags*. Considering the fact that the correlation of the time series observations



is obtained using values of the same series at previous times, this is called a serial correlation, or an autocorrelation.

The **AutoCorrelation Function (ACF)** is a plot of the autocorrelation of a dataset of a time series by lag. This plot is sometimes called a *correlogram* or an *autocorrelation plot*.

A partial autocorrelation or *PACF* is represented by the summary of the correlation between an observation in a time series with observations at previous time steps with the correlations of in between removed observations. (Anish 2020)

## **Conclusion**

Time series analysis is one of the major aspects in data analysis for any large organization as it helps in understanding seasonality, trends, cyclicity and randomness in the sales and distribution and other attributes. These factors help companies in making a well-informed decisions which are highly crucial for any business.

## **3 Application of ML**

### **3.1 Application of ML in Oil and Gas Industry**

A big part of daily routine tasks performed in oil and gas industry require analysing large and complex datasets, a process that can be automatized in order to achieve maximum efficiency and return on investment.

Potential ML applications in oil and gas industry are as following:

- Automation
- Data collection
- Data evaluation
- Algorithms development
- Consumable analytics
- Automated recommendations
- Maximized efficiencies
- Automated adjustments

The aforementioned applications can be implemented over the entire chain of oil and gas.

#### **3.1.1 Upstream**

Machine learning is capable of assisting with locating the most efficient place to start a well, as well as improving ways of oil and gas extraction such as:

- Predictive analysis (EAG 1Source 2020)
  - Accurate modelling
  - Exploration
  - Dig sites
  - Well logging
- Oilfield operations
- Drilling efficiencies
- Rig optimization
- Risk detection
- Remote operations
- Completion

These processes are very convenient as computers can analyse large datasets faster and more efficient than any human.

### **3.1.2 Midstream**

For services such as transporting products from the field to the refinery, ML can be helpful in providing recommendations for improving the delivery system.

### **3.1.3 Downstream**

Similar to upstream and midstream applications, ML algorithms are important for the downstream in processes like refining, processing, remote system operations and risk analysis.

Operating a refinery is a challenging task due to the multitude of processes going on simultaneously. Therefore, analysing and reporting every step can be difficult for human employees. ML however, can process all that information and come up with informed decisions as a support for human experts.

### **3.1.4 Back-office Management**

The office environment can be improved with the use of ML too. By observing a variety of working elements during operations, collected data can be used to make recommendations for business stimulation in areas such as:

- Maintenance
- Market analysis
- Forecasting
- Product marketing
- etc.

Implementing machine learning in oil and gas can bring a series of advantages.

- Increased production and decreased labour costs as a result of an improved drill modelling
- Increase revenues
- More stable profit margins
- Reduced risks
- Extended meantime between failures
- Energy consumption reduction

The list of advantages can continue, what is important to mention is that ML represents a long-term enhancement for oil and gas industry. Automation and data processing bring more effectiveness and attention to detail.

### **3.2 Application of ML in Other Industries**

The importance of machine learning has been recognized by many industries that are dealing with large volumes of data. Being able to find the message behind this data, companies can work in a more efficient, sustainable and economical manner. Some examples of already existing application of ML are:

- Financial sector – companies use ML to identify valuable insights in financial data and avoid any frauds. Opportunities for investments and trade can be recognized more easily, etc.
- Marketing and sales – ML is applied in analysing customers purchase history and provide recommendations shaped for their preferences.
- Government agencies – sensor data can be used to recognize ways to reduce costs and improve efficiency for agencies like public safety or utilities.
- Healthcare system – real-time patient information is generated by the means of sensors and provide details about heartbeat, blood pressure and other vital parameters. ML is used for identifying patterns in a patient's history and predict the probability of occurrence for any illnesses, as well as in diagnostics.
- Transportation systems – based on identified patterns on multiple routes, ML algorithms are used to identify potential problems that might arise on the way and come up with an automatic recommendation for a different route.

## **4 Data Acquisition**

Data used in this work was acquired from OMV's production database GDB – Gewinnungsdatenbank. This is an internal database for OMV Austria, and it contains information about fields, horizons, produced media, workover reports, production data, surface data, equipment, etc. In this chapter, used datasets for this work are described.

### **4.1 Available Data**

All the necessary data downloaded from GDB was stored in MS Excel, MS Word, PDF, VSD, JPG and DYN files. Data corresponds to wells equipped with SRP only. The following data sets have been received:

- Digital dynamometer cards
- Analogue dynamometer cards
- Workover reports
- Production data
- Equipment data
- Well schematics
- Surface data

#### **4.1.1 Digital Dynamometer Cards**

These cards have been recorded by the means of sensors installed at the wellbore. The frequency of such measurements varies from over 5000 files a day to 1 file a day. This part of the dataset is the most valuable and widely used in this work. The challenges related to visualization and high variation in the frequency will be discussed later, in a separate chapter.

#### **4.1.2 Analogue Dynamometer Cards**

These cards have been provided as JPG files and they represent surface dynamometer measurements performed manually at the wellbore. As the frequency of such measurements is not as high as digital ones, they have been used as a mean of proofing the correctness of sensor measurements. In Figure 24 such an example is illustrated.

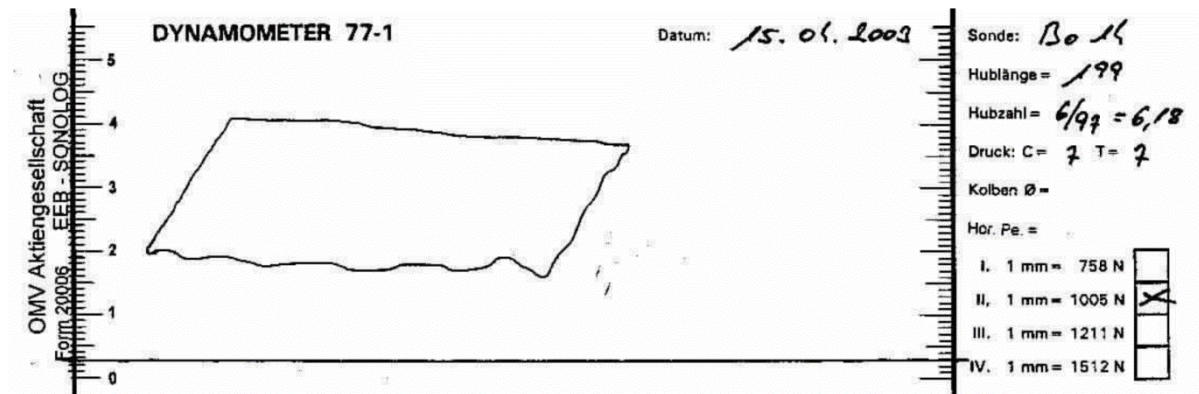


Figure 24: An example of an analogue dynamometer card downloaded from GDB

### 4.1.3 Workover Reports

Workover reports contain information about repairing, stimulation and/or maintenance activities performed in order to restore, prolong or enhance production. The reports have been retrieved in Excel files. See Figure 25 for an example.

SONDENNAME	BEHANDLUNG_BEGINN	BEHANDLUNG_ENDE	BH_CODE
ALTLICHTENWARTH 025	18.01.2018	22.01.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR-WECHSE
BOCKFLIESS 005	23.02.2018	27.02.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR KONTR
BOCKFLIESS 047	25.05.2018	06.06.2018	AUFWÄLTIGU - AUFWÄLTIGUNG; BEH. 5 - BEHANDLUNG 5
BOCKFLIESS 069	22.02.2018	23.02.2018	POLIEBRUCH - POLIERSTANGENBRUCH; BEH. 2 - BEHANDL
BOCKFLIESS 137	16.05.2018	06.06.2018	STICKSTOFF - STICKSTOFFINJEKTION; UMRÜINJEKT - UMR
BOCKFLIESS 160	27.02.2018	01.03.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR KONTR
BOCKFLIESS 176	20.12.2017	02.01.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR KONTR
BOCKFLIESS 201	13.04.2018	17.04.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR-WECHSE
EBENTHAL 014	23.01.2018	29.01.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR-WECHSE
EBENTHAL 016	23.03.2018	28.03.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR KONTR
EBENTHAL 020	18.05.2018	24.05.2018	TPWK - TIEFPUMPENWECHSEL KOMPLETT; STR-WECHSE

Figure 25: An example of workover reports file downloaded from GDB

The main challenge was reading the report itself as the entire information is contained in a single column. In order to tackle this challenge, individual reports have been extracted as Word files.

### 4.1.4 Production Data

Two different production data sets from separator measurements for 26 wells were used, containing information about the gross production volume, water cut, gas/oil ratio, gas volume, oil volume, fluid density, injected chemicals volumes, volumes of produced impurities (if any), tubing and casing pressure. Figure 26 and Figure 27 show how such files look like.

SONDENNAME	DATUM	Brutto [m³]	WC [%]	DICHTE	GÖV	Öl [t]	Gas [m³]
BERNHARDSTHAL SUED 006	08/10/2017 00:00:00	5	49	1	503	2	1,239
BERNHARDSTHAL SUED 006	10/11/2017 00:00:00	6	54	1	440	2	1,150
BERNHARDSTHAL SUED 006	01/20/2018 00:00:00	5	35	1	295	3	944
BERNHARDSTHAL SUED 006	03/04/2018 00:00:00	7	55	1	372	3	1,145
ERDPRESS 024	07/08/2017 00:00:00	30	68	1	70	9	676
ERDPRESS 024	07/28/2017 00:00:00	30	68	1	65	9	620
ERDPRESS 024	08/15/2017 00:00:00	37	67	1	90	11	1,114
ERDPRESS 024	09/04/2017 00:00:00	35	67	1	99	10	1,134
ERDPRESS 024	09/22/2017 00:00:00	33	76	1	51	7	404
ERDPRESS 024	10/09/2017 00:00:00	35	70	1	62	10	648
ERDPRESS 024	10/21/2017 00:00:00	35	68	1	65	10	704

Figure 26: An example of production data file downloaded from GDB, including produced volumes

GASSP_ENT	GASSP_EIN	CO2_INJ	N2_INJ	LPG_VOL	LPG_MASS	NGL	COND	DICHTE_BH	DICHTE	BRUTTO	OEL	H2O	H2OPROZ	GOEV	CO2	H2S	SCHWEFEL	WAERMEMENGE
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		
0.000	0.000	0.000	0.000		0.000	0.000	0.000			0.000	0.000	0.000			0.000	0.000		

SONDENNAME	SONDENKURZNAME	DATUM	FA	FAE	FAE12	FAE3	KOMMENTAR	ABSTEL	MEDIUM	ASS	BS_L	BS_G	BS	FS	SS	DUESE	TUB_DRUCK_STAT
ERDPRESS 024	ERD 24	22.07.2014	1	00N	00	N	Erstperforation 11.SH	1		0	0	0	0.000	0	24		0
ERDPRESS 024	ERD 24	31.07.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	16	16.000	0	8		0
ERDPRESS 024	ERD 24	01.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	16	16.000	0	8		0
ERDPRESS 024	ERD 24	02.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	0	0.000	0	24		0
ERDPRESS 024	ERD 24	04.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	16	16.000	0	8		0
ERDPRESS 024	ERD 24	05.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	8	8.000	0	16		0
ERDPRESS 024	ERD 24	06.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	8	8.000	0	16		0
ERDPRESS 024	ERD 24	07.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	8	8.000	0	16		0
ERDPRESS 024	ERD 24	08.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	16	16.000	0	8		0
ERDPRESS 024	ERD 24	09.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	0	0.000	0	24		0
ERDPRESS 024	ERD 24	11.08.2014	1	00G	00	G	Erstperforation 11.SH	1		0	0	8	8.000	0	16		0

Figure 27: An example of production data downloaded from GDB, including produced volumes and other information

### 4.1.5 Equipment Data

FIELD	AREA	SONDE	SONDENKURZNAME	HOR	PE	SINGLE	MEDIUM	STATION	FA	FAE12	FAE3	EINGEBAUTE	PUMPE
TEXT	TEXT	#T	#T	#			OIL/GAS/WATER	TEXT	#	#T	TEXT	TEXT	
A013 Area 1 - North		200005	BE S 5	431-11			Öl	Mühlberg	2	G0		GST-Pumpe	
A013 Area 1 - North		200006	BE S 6	431-40; 432-40			Öl	Mühlberg	2	G0		GST-Pumpe	
A006 Area 4 - Hochleiten		100024	ERD 24	111-40			Öl	Erdpress I	2	G0	W	GST-Pumpe	
A016 Area 4 - Hochleiten		200020	HL 20	111-10			Öl	Hochleiten	2	G0		GST-Pumpe	
A016 Area 4 - Hochleiten		200066	HL 66	202-11			Öl	Hochleiten	2	G0		GST-Pumpe	
A015 Area 2 - Matzen		200214	MA 214	109-35			Öl	Matzen XVII	2	G0		GST-Pumpe	
A015 Area 2 - Matzen		200568	MA 568	216-10			Öl	Matzen V	2	G0		GST-Pumpe	
A015 Area 2 - Matzen		200569	MA 569	216-10			Öl	Schönkirchen VI	1	00	F	GST-Pumpe	
A015 Area 2 - Matzen		300217	P 217	213-91			Öl	Prottes I	2	G0		GST-Pumpe	

Figure 28: An example of equipment data file downloaded from GDB

This is a single Excel file which includes all details describing horizons, pump parts installed in the wellbore, pressures, fluid levels, produced media volume. Figure 28 shows how such a file looks like.

### 4.1.6 Well Schematics

Well schematics files have been retrieved in VSD files and these contain details about geological horizon, well's trajectory, TVD, MVD, perforations, tubing and casing parameters, artificial lift system. See Figure 29 for an example.





04.10.2018  
 TPWK, STW, PGW  
 erstellt: Kaufmann

BeS 5

Endteufe: 2227,00m  
 Ges. Abweichung: 492,52m  
 Azimut: 236,75°  
 Verkürzung: 158,56m  
 KOP: 1232,74m MD  
 Neig. max: 40,97° bei MD 2123,6m  
 Zementsohle: 2184,53m  
 Horizont: 431  
 PE: 11

LK 318mm, RX 45, 3000 PSI

**Verrohrungsdaten:**

13 3/8" 8,00m zement bis zutage  
 9 5/8" 498,71m zement bis zutage  
 7" 2224,50m zement bis Plan 300m

**Einbau:**

Abg FI Übg 3 1/2" EU 8G x 3 1/2" EU (0,23m)  
 206 Si 3 1/2" EU weiß (1923,58m)  
 Übg 2 7/8" EUZ x 3 1/2" EUM (0,38m)  
 TPS id=58mm (0,23m)  
 1 Si 2 7/8" EU MR weiß (9,58m)  
 7" LSP 47B4 Nr.512 ohne Gummi (1,21m)  
 1 Si 2 7/8" EU weiß (9,53m)  
 Jaku Tub GSA (8,14m)  
 3 Si 2 7/8" EU SKR weiß (28,60m)  
 SM 2 7/8" EU (0,18m)

**TPS OK: 1924,19m**  
**LSP OK: 1934,00m**  
**GSA OK: 1944,74m**  
**Gesamteinbau: 1981,66m**

4 PS 1" Tenaris neu (6,00m)  
 mit 3 Protektoren 1" x 3 1/2" E-P-FIX-L  
 76 PG 1" Tenaris gebr (579,12m)  
 mit je 4xProtektoren 1" x 3 1/2" E-P-FIX-L  
 172 PG 7/8" Tenaris (85neu/87gebr) (1310,64m)  
 mit je 4xProtektoren 7/8" x 3 1/2" E-P-FIX-L  
 2 Sinkbar 1 3/4" gebr (12,32m)  
 mit 2 PS 7/8" Tenaris (1neu,1gebr) (1,20m)  
 Ins TP 25-150-RHAC-21 Nr.2007/18 + SRP

**Rohre: 3 1/2" EU, J55 weiß**  
 KD=72,82mm, ID=76mm, WD=6,45mm

Polierstange neu am 03.10.2018 (7,45m)

Fräser- und Scrapermarsch:  
 01.06.2015 mit Stirnfräser od=155mm  
 + Scraper Nr.17 147-170mm bis 1760m

Offene Perf.: (CE)

1769,50m – 1774,00m 1. Eggenburgsandstein

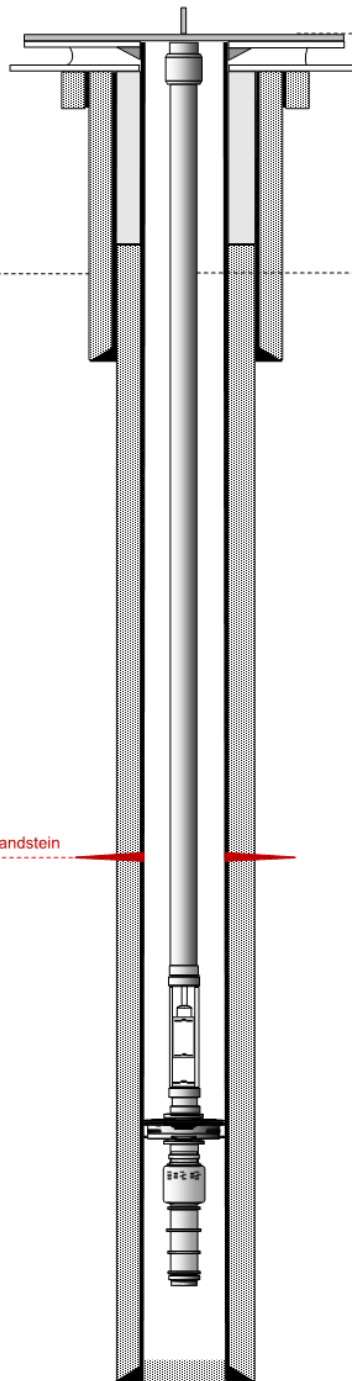


Figure 29: An example of well schematics file downloaded from GDB

### 4.1.7 Surface Data

Surface data files contain information about the energy consumption, active power of the motor, dynamic fluid level, frequency output. Figure 30 illustrates an example.

Station	Sondenname	Datum	Strom	Wirkleistung	Dyn. Spiegel MURAG	Frequenz Ist
Mühlberg	BE S 5	25.10.2017 01:00	46,37	7,76	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 02:00	46,33	7,75	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 03:00	46,37	7,73	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 04:00	46,38	7,73	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 05:00	46,38	7,74	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 06:00	46,37	7,73	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 07:00	46,39	7,74	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 08:00	46,39	7,74	1769,73	24,99
Mühlberg	BE S 5	25.10.2017 09:00	46,39	7,73	1769,74	24,99
Mühlberg	BE S 5	25.10.2017 10:00	46,39	7,73	1769,74	24,99

Figure 30: An example of surface data file downloaded from GDB

## 4.2 Data Integration

Having such a large amount of different data files, it was necessary to find a structured way for viewing the available data at once. As a result, a data matrix was created containing all available data sets, listed per well. The matrix consists of well names listed in a column, followed by the time intervals for which every data set listed above is available, number of samples/measurements. This particular way of viewing the available data allows to identify at a glance missing data interval and any patterns in data storage. See Appendix A for snapshots of the data matrix.

## 5 Data QC/QA

The data set containing digital dynamometer cards is the most extensive one, including over 7.5 million measurements. One of the main challenges in tackling such a large data set was developing an automatized process for visualization and quality check. For this purpose, Excel was used to automatically plot digital dynamometer cards altogether with multiple functions and tasks to be applied. The detailed steps of managing digital dynamometer cards are explained in this chapter.

### 5.1 DC Data Overview

Digital dynamometer cards (DCs) were extracted from GDB in DYN format. There are multiple series of apps that can be used in order to open such files. Most accessible ones are Notepad ++ or MS Excel. The DCs were received for 79 wells for the years 2017 – 2020.



Figure 31: Examples of files per day

Figure 31 illustrates a few examples of numbers of files per day over the entire duration of three years for four different wells. It can be noticed that the measurements are not stored continuously as there are some gaps in the data at the same time intervals. This is explained by the fact that during these periods some sensors changing activities took place and data

could not be recorded. Another important observation can be found in figures A and B, the number of files per day and over time is significantly higher as compared to figures C and D. In the wellbore illustrated in figure C the sensors have been installed later in time, starting September 2018. The wellbore illustrated in figure D had a defective production and was removed from active wells.

Another visible finding is the frequency of files per day. It is described by significant fluctuations over time. An explanation to this is still unclear as the measurements are stored according to some pre-set criteria defined by the data management team. The number of files vary between 5000/day to 1/day.

### 5.2 Data Visualization

The main challenge consisted not only in dealing with the high number of measurements initially received but also the continuity of this process as new data was constantly fed in. As a result, a standardized method for automatically plotting surface dynamometer cards, labelling them, and performing the visual analysis has been created using Excel.

Resulted plot represents position (m) versus load (N) as illustrated in Figure 32.

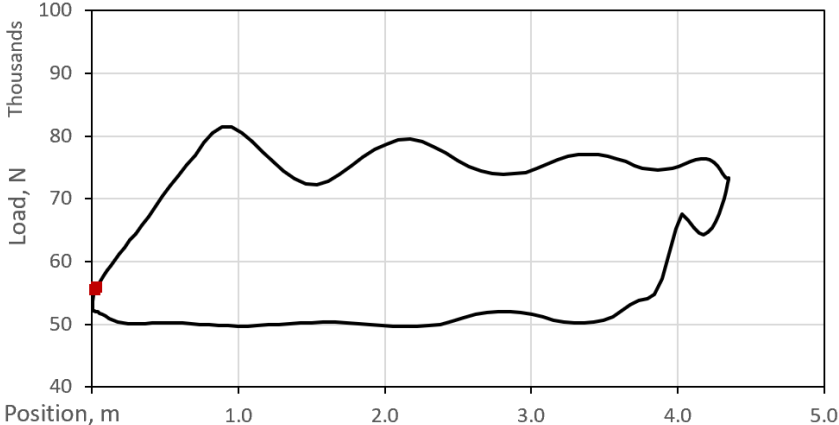


Figure 32: Surface dynamometer card

Load and position are also plotted separately for a better identification of existing errors.

An example of an erroneous measurement is illustrated in Figure 33.

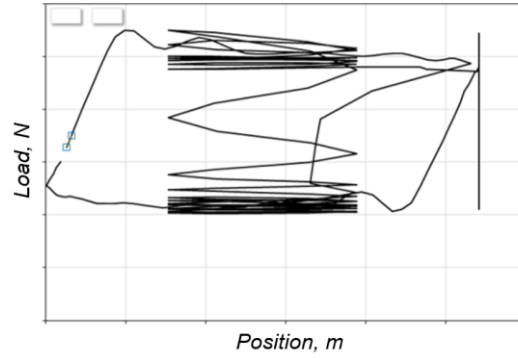


Figure 33: Example of erroneous measurement

The dynamometer card such as those plotted in Figure 33 will be labelled as “bad” and filtered out from the entire dataset.

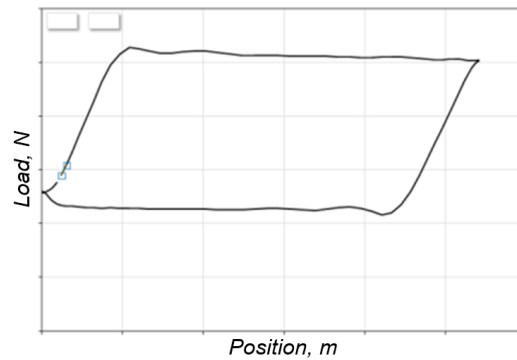


Figure 34: Example of good measurement

Figure 34 illustrates an example of a good measurement, this will receive two labels, one for the quality “good” and second for the pump state identified by the shape of the card (ex: “normal”, “fluid pound”, “gas interference”, etc.).

As the main goal in the entire project was to identify the trend in SRP behaviour, new data was constantly fed in. The advantage of the afore mentioned plot consists in its flexibility as it is linked directly to the folders containing raw data, it can automatically extract, correct, and plot the measurements.

## 6 Features for Model Building

After visualizing available dynamometer cards, as expected, a variety of shapes indicating measurements quality and SRP state were identified. Based on that, it was decided to create an intelligent model using artificial neural networks which would automatically identify dynamometer cards shapes and further use its results for performing the SRP trend analysis.

Considering the afore mentioned challenge, it was decided to divide the model's tasks into two steps. The first step was to identify the quality of the data and classify it in two classes related to quality.

After removing bad quality data, the good part was used to distinguish pump states and classify them in 10 categories which are representative of identified pump states.

A dataset composed of multiple channels including time stamps, data quality and pump states was created and used as desired output. It is worth mentioning that the input set contained manually labelled dynamometer cards. Out of 5 million measurements provided, 12.300 were manually labelled for the quality and 17.330 for the pump states. Due to the availability of labelled data sets, supervised learning was selected for the learning approach and feed-forward neural network for the model building.

Proposed model idea implies a very complex structure, therefore in order to reduce complexity and computational costs, it was necessary to select and calculate different features and use them as an improved learning process and input for the neural network. After all, feature selection has always been an exercise of great importance for machine learning, often overcoming the model selection itself.

Different features have been calculated using macros in order to describe every measurement.

Calculated features have been grouped as following:

- Statistical features
- Physical features
- Elliptic Fourier Transform features
- Fractal dimensions
- Others

**Statistical features** have been applied to physical values such as the raw sucker rod data and processed values. Processed data is data that has been converted in order to ensure that each dynamometer card contains exactly one stroke and the starting point is the beginning of the upstroke. The importance of this step lies in the fact that raw data contains examples of single dynamometer cards with multiple stroke recordings (up to five) and the starting point of the recording would not always be the beginning of an upstroke rather located at some arbitrary position between up and down strokes, seldom directly at the downstroke outset.

Raw data is exclusively used for the QC model which is further described in Chapter 7 and processed data was used for the pump sate model described in Chapter 8.

Later, statistical features have been applied to the physical features, described in 6.1.

**Physical features** have been calculated from raw data, normalized rod position and rod load and by calculating stroke integrals from full, up and down stroke, physical features of surface dynamometer cards could be integrated in the artificial neural network input. Used and calculated physical features are as following:

- Sucker rod load [N] – obtained from the raw dataset.

The surface dynamometer card records the movement of the polished rod during a pumping cycle. The rod string is following a time versus position model, reacting with the load exerted on the string by the well. As a result, surface dynamometer cards record the variation of load on the polished rod during a pumping cycle.

Load is a function of pump position at which the force is measured during a pump cycle.

- Standardized sucker rod load [N] – calculated from raw sucker rod load with the aim of having a better identification of the shape independent of the size and position on the graph.

- Sucker rod yank [N/s] – calculated from the sucker rod load as a differential of the load with respect to time, respectively the change of load over time.

This term is taken from biophysics. Biologists and biomedical engineers are suggesting to designate the term "yank" for changes in force over time, phenomenon that our muscles and nerves can perceive and respond to.

Their ideas were published on September 12, 2019 in Journal of Experimental Biology.

Scientists that study sports often use the term "rate of force development", a measure of explosive strength. Scientists who research gait and balance, in both animals and humans, further analyse how quickly the change of forces on the body takes place. They believe that existing terminology describing the time derivative of force was too unwieldy and limiting, therefore Lena Ting, Ph.D., professor of rehabilitation medicine at Emory University School of Medicine and the Wallace H. Coulter Department of Biomedical Engineering at Georgia Tech and Emory and her colleagues came up with the term "yank".

To a great extent, until now yank has been used in the study of jumping, sprinting, capturing prey and maintaining balance, in the analysis of muscles behaviour and tendons, sensory feedback and spinal reflexes, all the way down to the contributions of individual cells. (Lin, McGowan, Blum and Ting 2019)

Similar perturbations and movements to those in the musculoskeletal system can be associated with the movements happening during the pumping cycle of a sucker rod pump. As a physical phenomenon and its characteristics yank could be applied universally as a representation of changes in force over time.

- Sucker rod cycle time [spm] – obtained from the raw dataset.
- Sucker rod velocity [m/s] – calculated from the sucker rod position as a differentiation of position with respect to time.

Sucker rod position [m] – obtained from the raw dataset.

Every subchapter contains graphical representations of feature values. These visual representations embody the relationship between feature values and quality labels assigned to each measurement, calculated for four of the most representative parameters used to describe sucker rod pump performance. The feature analysis based on the graphs depicted below helps to understand their impact and importance for the automatic identification of dynamometer cards quality.



Darker areas betoken the overlap between values representing good and bad quality measurements. Brighter points (usually on the sides) illustrate unique values for the particular label and correspond to 100% separation between good and bad quality data points.

Hence comes the greatest importance of having calculated multiple features as with every feature we can have a clearer understanding about the division of values corresponding to good or bad quality measurements.

## 6.1 Statistical Features

Calculating statistical features is a common approach for identifying patterns in time series data. That is because the dataset to be used has a very high dimensionality and the number of possible features to be extracted is enormous.

A set of statistical features has been calculated in order to measure multiple properties of the variables and reduce the data dimensionality. These are as following:

- **Median** – as a measure of central tendency.

Median is the central number in an organized list of numbers. In order to find out the median value in a succession of numbers, initially, the numbers have to be sorted, or ordered from lowest to highest or highest to lowest. The median can be used to determine an approximate average, or mean, but is not to be confused with the actual mean.

- In the situation of an odd total of numbers, the median value will be the number that is in situated the middle, with an equal amount of numbers below and above.
- In the case of an even total of numbers, the middle pair has to be identified, then added together, and further divided by two in order to find the median value.

The median is occasionally applied as opposed to the mean, usually when there are outliers in the progression that could skew the average of the values. The median of a sequence is normally less afflicted by outliers than the mean. (Ganti 2021)

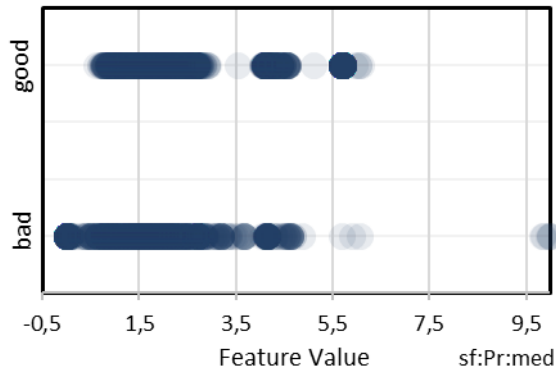


Figure 35: Raw sucker rod position, m

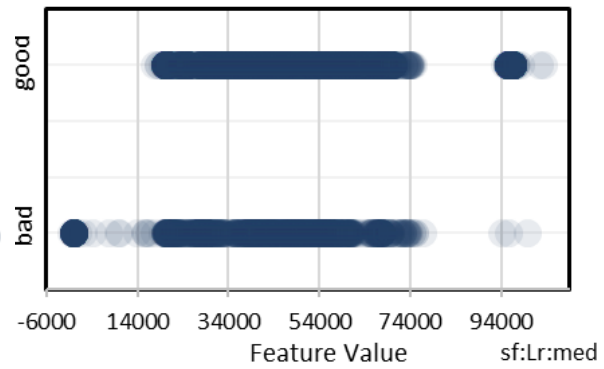


Figure 36: Raw sucker rod load, N

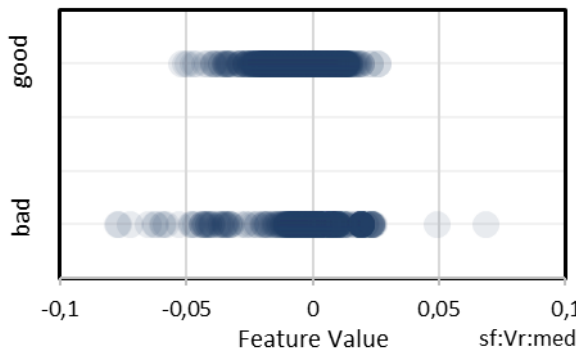


Figure 37: Raw sucker rod velocity, m/s

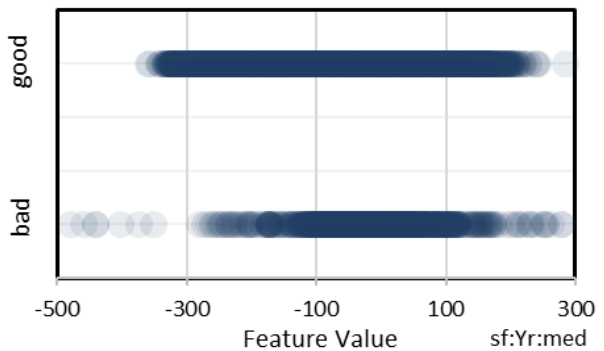


Figure 38: Raw sucker rod yank, N/s

Looking at median derived features, one can easily detect the fact that all values higher than 5,5 m for the raw sucker rod position in Figure 35 are definitely bad, whereas values between 1 m and 3 m contain data points for both good and bad labels.

In the case of raw sucker rod load in Figure 36, values lower than 20000 N are bad, values between 20000 N and 74000 N can be linked to both good and bad measurements.

Raw sucker rod velocity in Figure 37 carries a clearer distinguishment of bad cards and specifically, values lower than -0,05 m/s and higher than 0,025 m/s are certainly bad.

Values smaller than -300 N/s for the raw sucker rod yank in Figure 38 are clearly attributed to bad dynamometer cards.

By analysing the first sub-set of statistical features it is hard to make a clear separation between good and bad points, therefore it is necessary to analyse other features calculated for the same parameters.

3. **The arithmetic mean** represents the average of the variables  $\{x_1, x_2, \dots, x_m\}$  located in a time window.

4. **Span value** – it represents the difference between the minimum and maximum values of a given range. The main goal of calculating span is to make sure that every datapoint has the same scale and every feature is equally important.

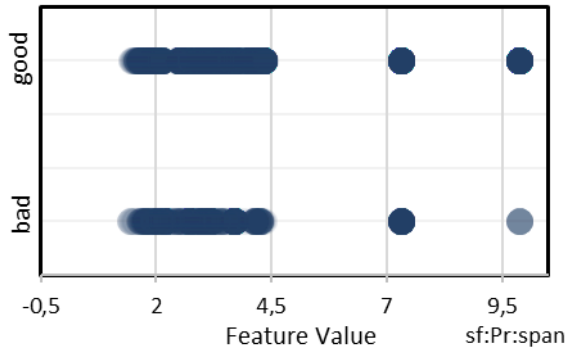


Figure 39: Raw sucker rod position, m

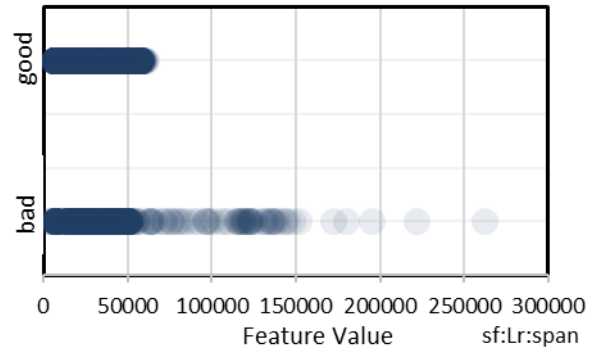


Figure 40: Raw sucker rod load, N

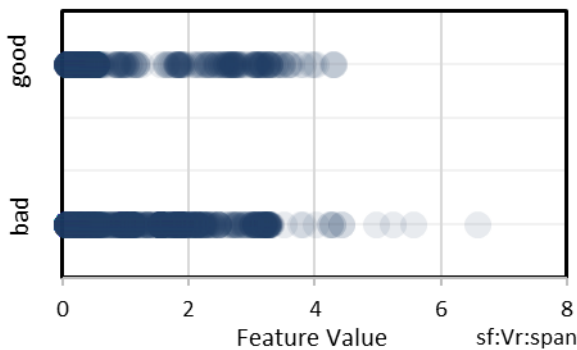


Figure 41: Raw sucker rod velocity, m/s

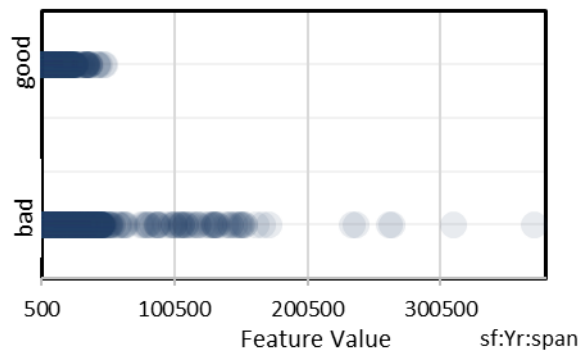


Figure 42: Raw sucker rod yank, N/s

The raw sucker rod position in Figure 39 derived using span value shows almost a complete overlap between good and bad data points.

Meanwhile, for the raw sucker rod load in Figure 40 one can very clearly see that all values higher than 50000 N are bad cards.

For velocity in Figure 41, all points exceeding 4 m/s are bad measurements.

Raw sucker rod yank in Figure 42, similar to load, illustrates very well the fact that all bad measurements are linked to values higher than 100000 N/s.

5. **Skewness** represents the degree of distortion from the normal distribution. When the distribution is symmetrical, it will have a skewness equal to zero. Positive skewness is represented by the shift to the left which means that the tail on the right side is longer. Negative skewness is represented by a shift to the right, the tail on the left side is longer (Dugar 2018).

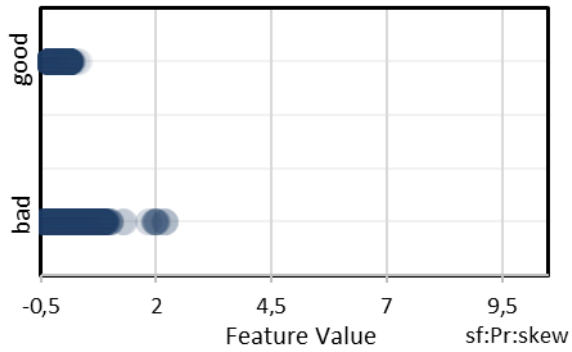


Figure 43: Raw sucker rod position, m

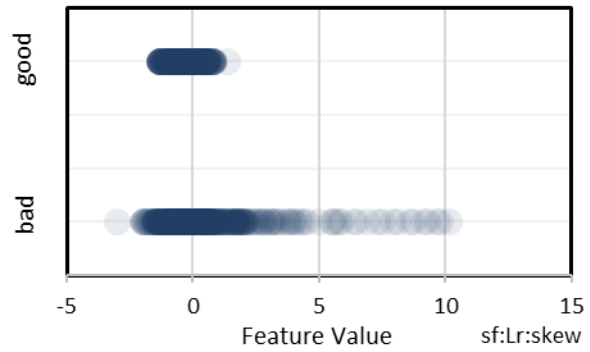


Figure 44: Raw sucker rod load, N

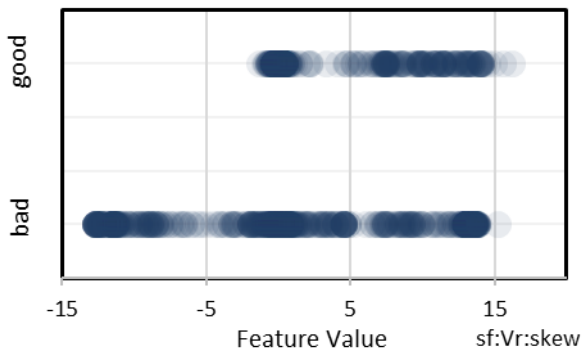


Figure 45: Raw sucker rod velocity, m/s

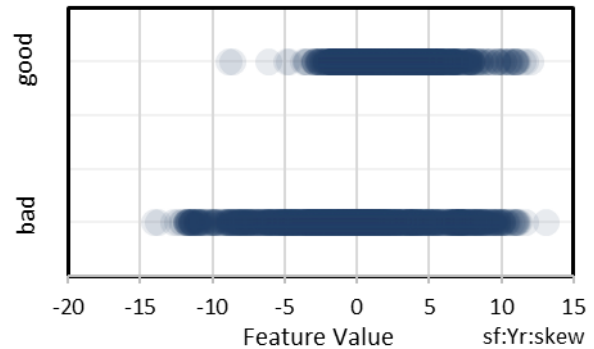


Figure 46: Raw sucker rod yank, N/s

Skewness features offer a better separation between good and bad data points. As for raw sucker rod position in Figure 43, all values higher than 0,5 m are bad.

For the raw sucker rod load in Figure 44, values lower than -2,5 N and higher than 2,5 N are bad.

Raw sucker rod velocity in Figure 45 has a clear division of good from bad values, placing all bad measurements to values lower than -3 m /s.

Raw sucker rod yank in Figure 46 has all bad quality measurements placed below -5 N/s.

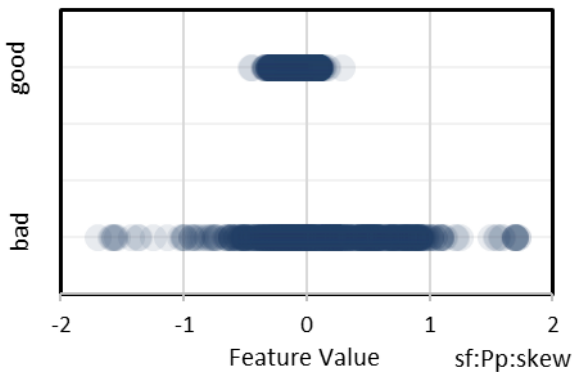


Figure 47: Processed sucker rod position, m

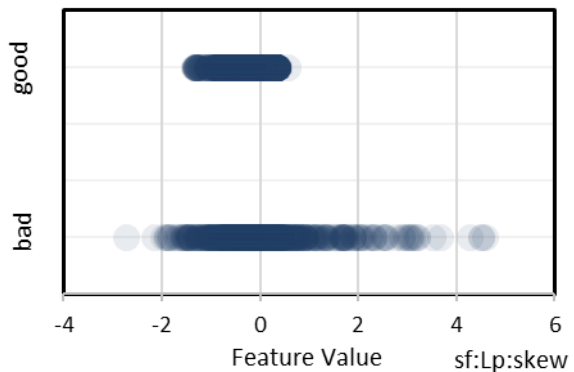


Figure 48: Processed sucker rod load, N

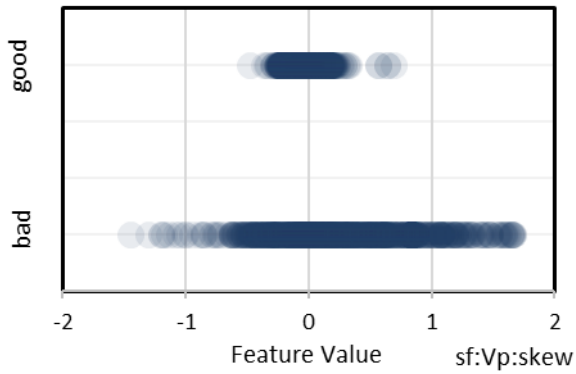


Figure 49: Processed sucker rod velocity, m/s

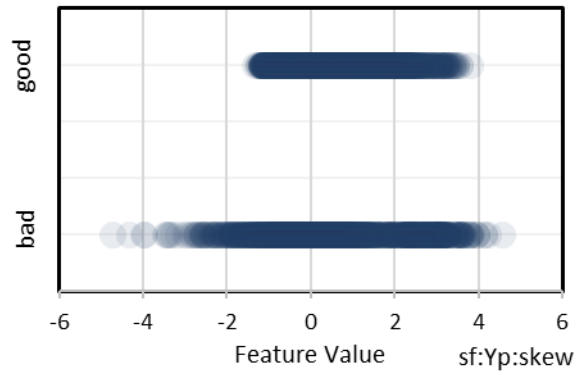


Figure 50: Processed sucker rod yank, N/s

Skewness was applied also for processed values in order to have a better understanding of these features. In the processed sucker rod position in Figure 47 one can notice that all values smaller than - 0,5 m and larger than 0,5 m are representative for the bad label.

In processed sucker rod load in Figure 48, datapoints situated lower than -2 and higher than 1 are bad measurements.

For processed sucker rod velocity in Figure 49, the distinguishment between bad and good labels is high. Datapoints under -0,5 and over 0,5 are bad.

Processed sucker rod yank in Figure 50 has bad labelled data under -1,5 N/s.

6. **Kurtosis** represents the outliers in the data distribution, it is a measure of the flatness or peakedness of data distribution. There are three categories: mesokurtic – the normal distribution, leptokurtic – longer distribution, peak is higher than the mesokurtic, narrow vertical range, platykurtic – shorter distribution, peak is lower than the mesokurtic, wider vertical range (Dugar 2018).

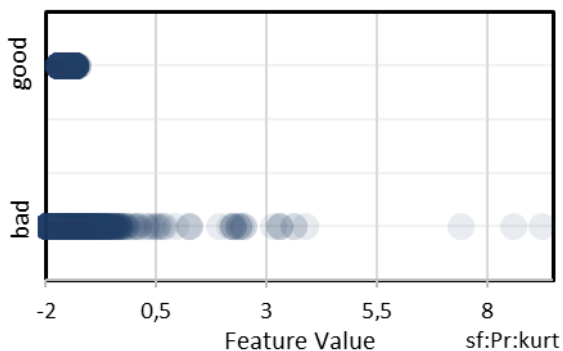


Figure 51: Raw sucker rod position, m

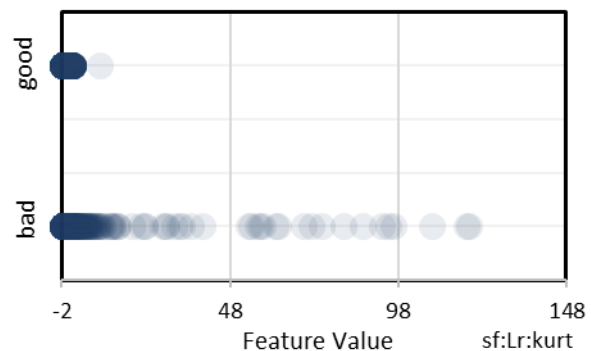


Figure 52: Raw sucker rod load, N

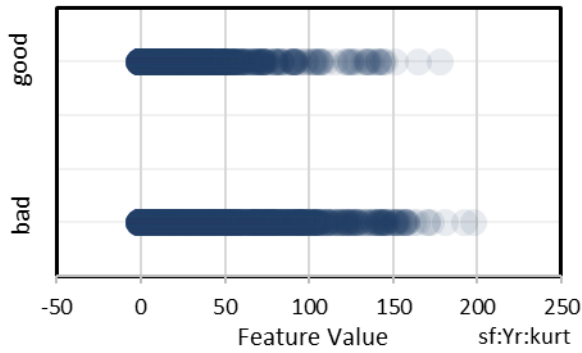


Figure 53: Raw sucker rod yank, N/s

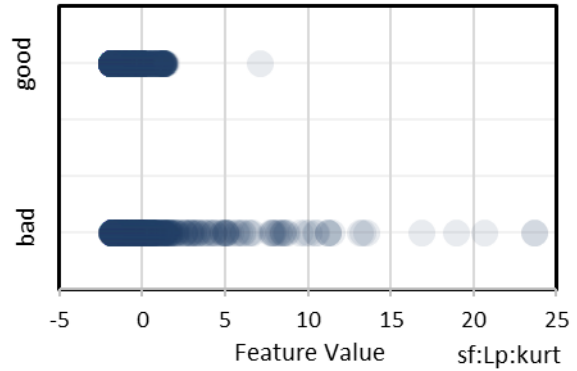


Figure 54: Processed sucker rod load, N

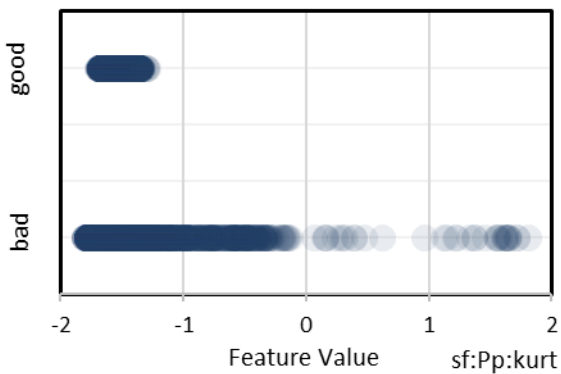


Figure 55: Processed sucker rod position, m

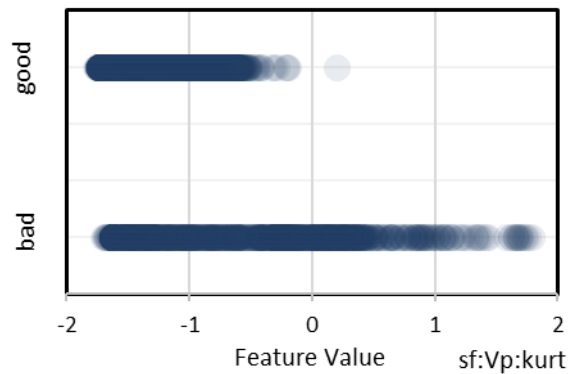


Figure 56: Processed sucker rod position, m

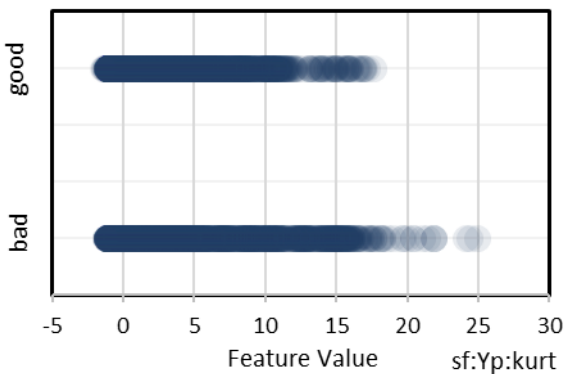


Figure 57: Processed sucker rod position, m

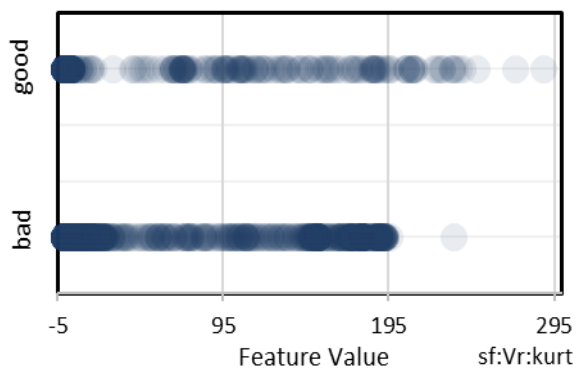


Figure 58: Raw sucker rod velocity, m/s

Analysing kurtosis derived features, raw sucker rod position (Figure 51) and load (Figure 52) have a very distinct division of bad samples, whereas for position, bad data points are located above -1 m and for load above 18 N.

For raw sucker rod yank in Figure 53 good and bad values are almost overlapping. Meanwhile, for processed sucker rod velocity (Figure 58) it is clear that all datapoints larger than 195 m/s are linked to good labelled dynamometer cards.

Processed data shows a similar behaviour for all four parameters. Processed position (Figure 55) has its bad cards over the values of -1,3 m, processed load over 2 N (Figure 54), yank (Figure 57) higher than 17 N/s, and velocity (Figure 56) over 0 m/s.

- Entropy** was calculated as a measure of impurity, or in other words, entropy is a measurement of homogeneity. It returns the information about how non-homogeneous the input dataset is. (Analytics Vidhya 2020)

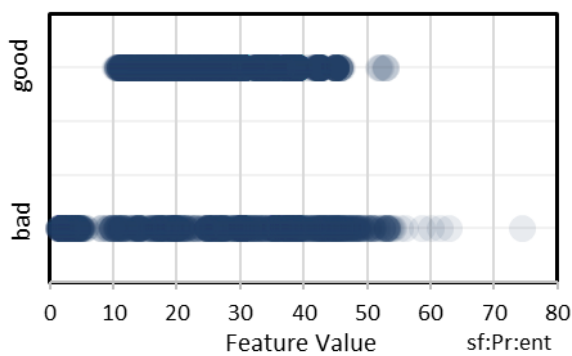


Figure 59: Raw sucker rod position, m

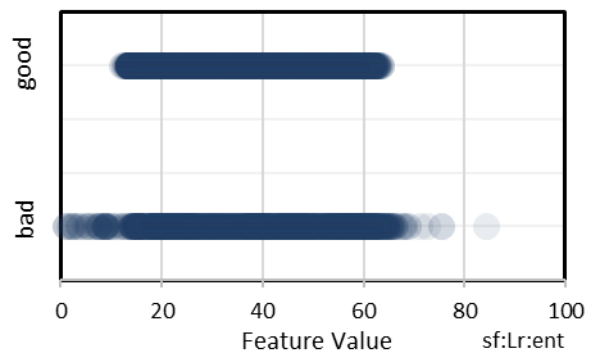


Figure 60: Raw sucker rod load, N

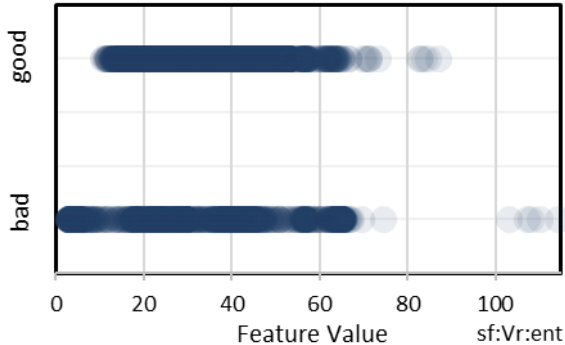


Figure 61: Raw sucker rod velocity, m/s

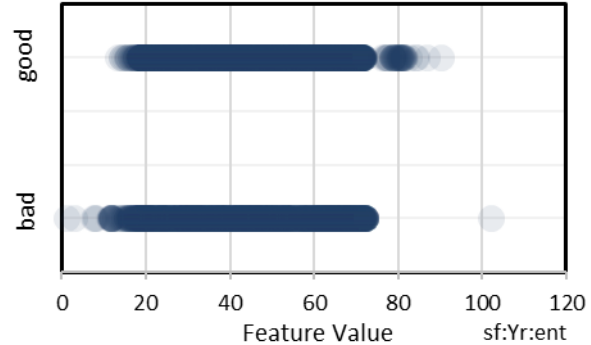


Figure 62: Raw sucker rod yank, N/s

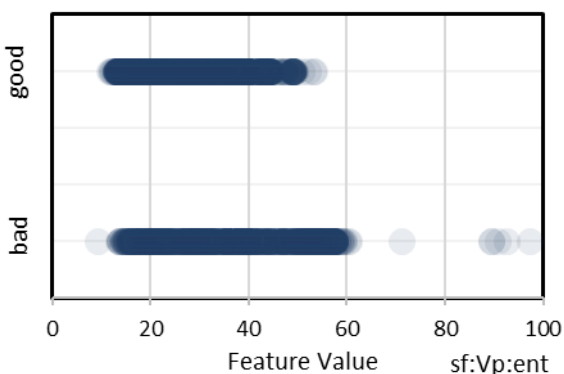


Figure 63: Processed sucker rod velocity, m/s

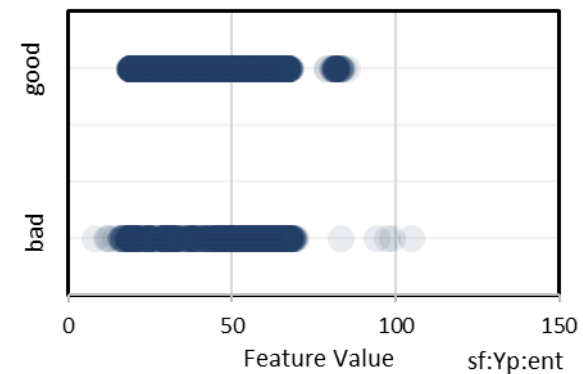


Figure 64: Processed sucker rod yank, N/s

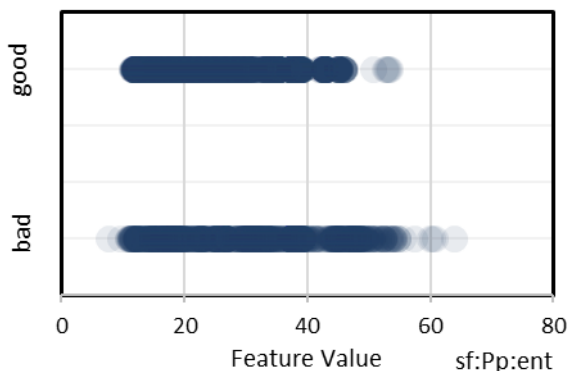


Figure 65: Processed sucker rod position, m

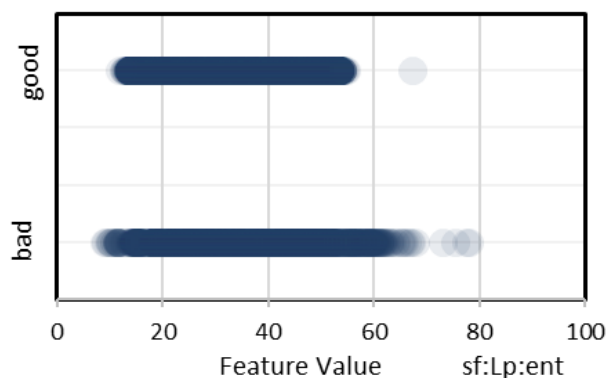


Figure 66: Processed sucker rod load, N

Raw sucker rod position in Figure 59 illustrates a separation of bad from good measurement at datapoints smaller than 10 m and larger than 55 m. For raw sucker rod load in Figure 60 this takes place at points lower than 10 N and higher than 65 N. All datapoints over 100 m/s are bad cards as of raw velocity (Figure 61) calculations.

Comparable to kurtosis derived features, the raw sucker rod yank (Figure 62) calculated for entropy features shows an overlap between good and bad data points.

Processed sucker rod velocity (Figure 63) has its bad measurements over 55 m/s, processed sucker rod yank (Figure 64) over 80 N/s, position (Figure 65) over 55 m, and load (Figure 66) over 60 N.

## 6.2 Physical Features

Before proceeding to calculating physical features, rod position had to be normalized. A normalized rod position is situated between a very small value at the beginning of the stroke up to a very large value at the end of the upstroke and turns to a very small value close to zero at the end of the downstroke, which is the beginning of the next upstroke. It makes data usable for different wells and stroke sizes and used as a feature. After normalizing all rod positions, the maximum delta position or the maximum change in position was identified. If there were some abnormal changes of stroke, then most probably, the value was larger than the usual maximum of the regular strokes.

### ***Fractal dimensions***

In order to better identify the trend, a rescaled range statistical analysis was performed. Calculated features are as mentioned earlier for raw and processed sucked rod load, position, velocity and yank, but this time as fractal dimensions. In other words, a fractal dimension describes how complicated a self-figure is and quantifies the complexity as a ratio of the change in detail to the change in scale.



Rescale range analysis helps to identify and evaluate how much persistence, randomness or mean reversion is in time series data. Strong trends are captured by the Hurst exponent, also known as long – range dependence, it is also capable of extrapolating a future value or average of the data. The Hurst exponent oscillates between zero and one. When the Hurst exponent is greater than 0.5, the data is exhibiting a strong long-term trend, as in the case of our data, and when H is less than 0.5, a trend reversal is more likely.

Rescaled range is simply calculated by dividing the span value of cumulative mean adjusted data points (sum of each data point minus the mean of the data series) by the standard deviation of the values over the duration of given time series. The rescaled range is increasing with increasing the number of observations in a times series. (Cory 2020)

Proceeding to feature plots, physical features show a better separation of good from bad dynamometer-cards.

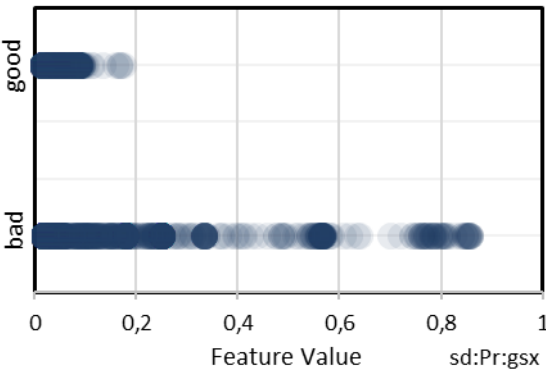


Figure 67: From stroke detector, raw sucker rod position, m  
Normalized absolute differences in the whole stroke

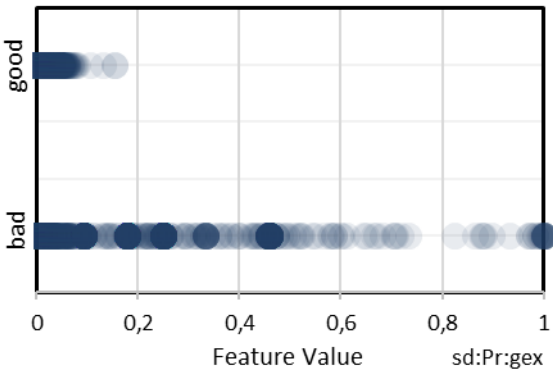


Figure 68: From stroke detector, raw sucker rod position, m  
Normalized absolute differences at the end of the stroke

Raw sucker rod position in Figure 67 derived from normalized absolute differences in the whole stroke shows that values larger than 0,2 m are bad data.

Same conclusion can be drawn by looking at raw position (Figure 68) derived from normalized absolute differences at the end of the stroke, values larger than 0,2 m are bad measurements.

For the load in Figure 69, the division between good and bad happens at values over 0,6 N for bad cards.

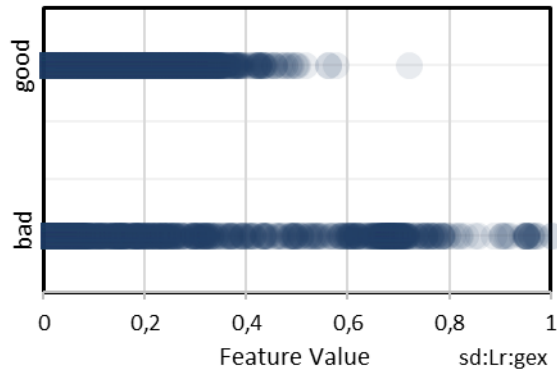


Figure 69: From stroke detector, raw sucker rod load N

Normalized absolute differences at the end of the stroke

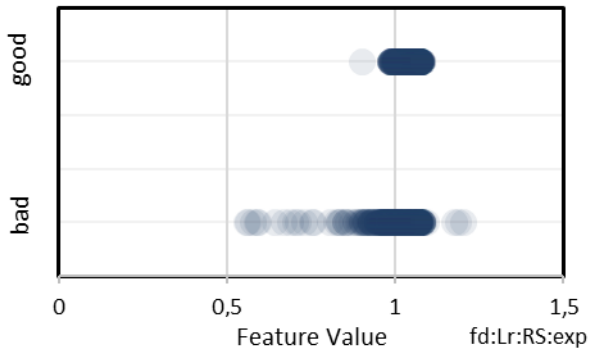


Figure 70: Fractal dimensions, raw sucker rod load, N

Rescaled range statistical analysis, Hurst exponent

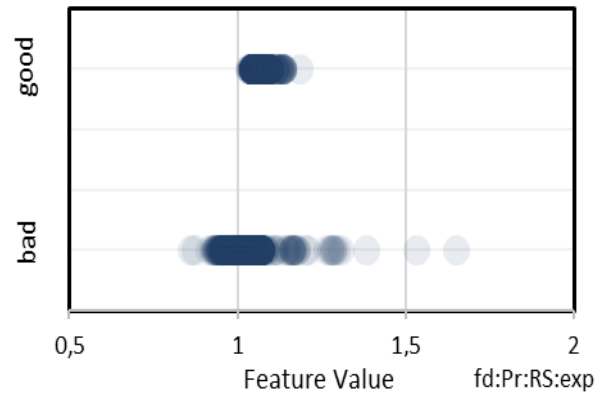


Figure 71: Fractal dimensions, raw sucker rod position, m

Rescaled range statistical analysis, Hurst exponent

Fractal dimensions features have a very good separation of bad measurements. As for raw sucker rod load (Figure 70) derived from rescaled range statistical analysis, bad datapoints are below 0,1 N and larger than 1,1 N.

For position derived features in Figure 71, values smaller than 1 m and larger than 1,25 m are linked to bad quality data.

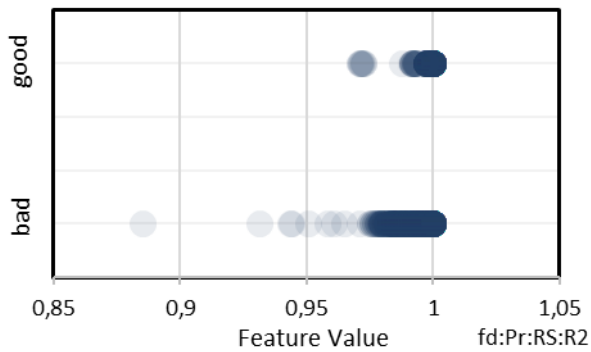


Figure 72: Fractal dimensions, raw sucker rod position, m

Rescaled range statistical analysis, log-log power function

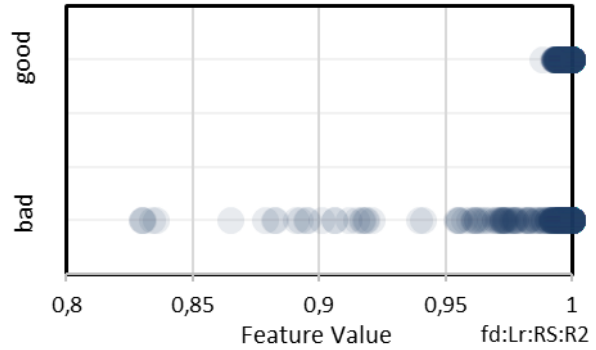


Figure 73: Fractal dimensions, raw sucker rod load, N

Rescaled range statistical analysis, log-log power function

For the same position (Figure 72) but derived using rescaled range statistical analysis and log-log power function, we see that points located under 0,925 m are bad quality datapoints.

For the sucker rod load in Figure 73, bad quality data has values smaller than 1 N.

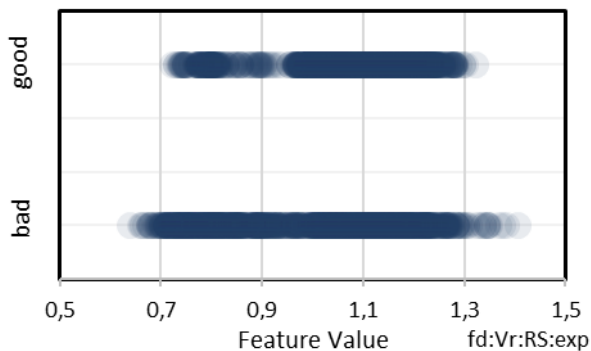


Figure 74: Fractal dimensions, raw sucker rod velocity, m/s

Rescaled range statistical analysis, Hurst exponent

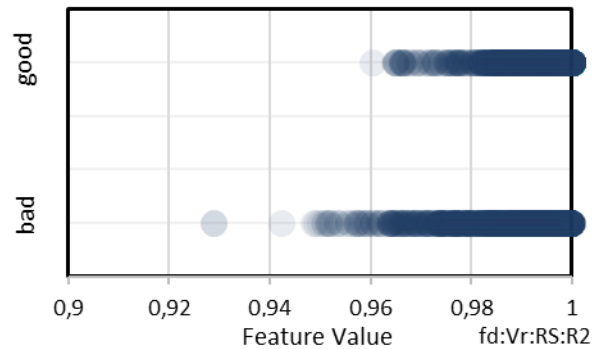


Figure 75: Fractal dimensions, raw sucker rod velocity, m/s

Rescaled range statistical analysis, log-log power function

Raw sucker rod velocity (Figure 74) derived with rescaled range statistical analysis and Hurst exponent doesn't show a good separation between good and bad values, thus practically overlapped.

Whereas, for velocity derived using log-log power function (Figure 75), it is easy to see that values below 0,96 m/s are bad measurements.

Raw sucker rod yank (Figure 76) derived with Hurst exponent has its bad measurements range for values lower than 0,3 N/s, and higher than 1,2 N/s.

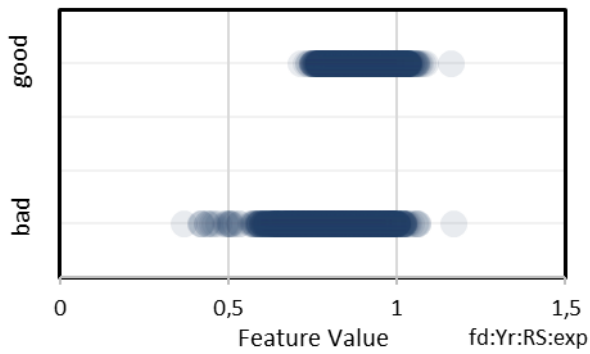


Figure 76: Fractal dimensions, raw sucker rod yank, N/s  
Rescaled range statistical analysis, Hurst exponent

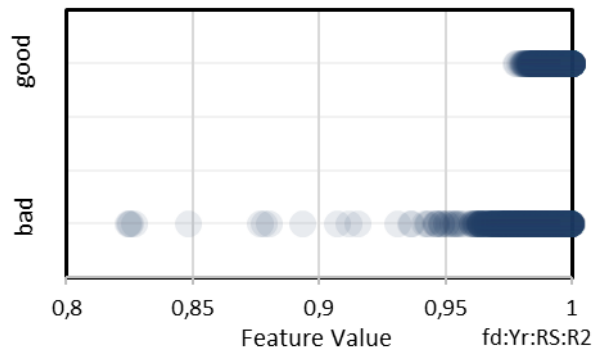


Figure 77: Fractal dimensions, raw sucker rod yank, N/s  
Rescaled range statistical analysis, log-log power function

Yank derived with log-log power function in Figure 77 has its bad quality values under 0,93 N/s.

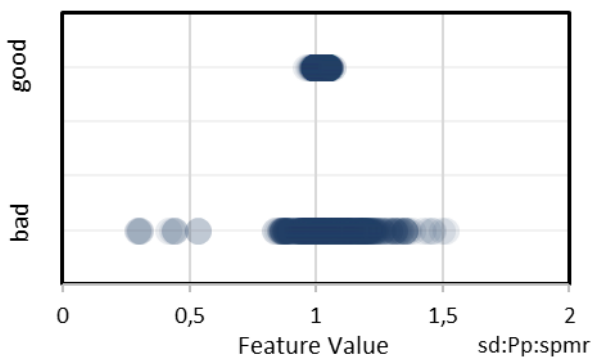


Figure 78: From stroke detector, processed sucker rod position, m  
Strokes per minute ratio

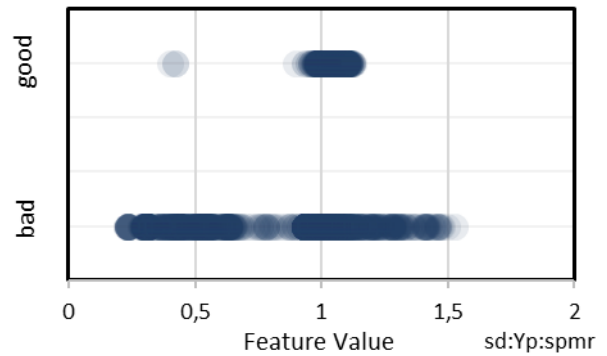


Figure 79: From stroke detector, processed sucker rod yank, N/s  
Strokes per minute ratio

Processed sucker rod position (Figure 78) derived with strokes per minute ratio has a broad range of bad measurements, and these are all values lower than 0,1 N/s, and higher than 1,1 m.

Processed sucker rod yank (Figure 79) derived with strokes per minute ratio has a distinguished range for bad measurements, and these are all values lower than 0,1 N/s, and higher than 1,2 N/s.

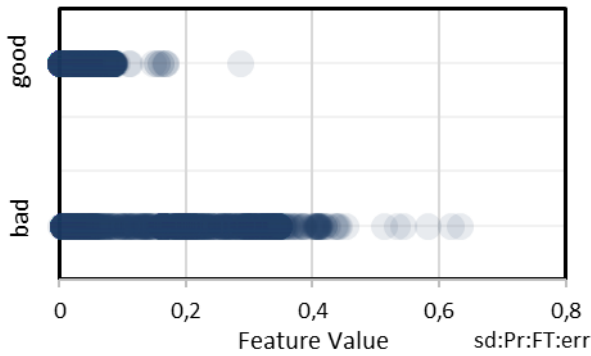


Figure 80: From stroke detector, raw sucker rod position, m

Fourier transform, root mean square error

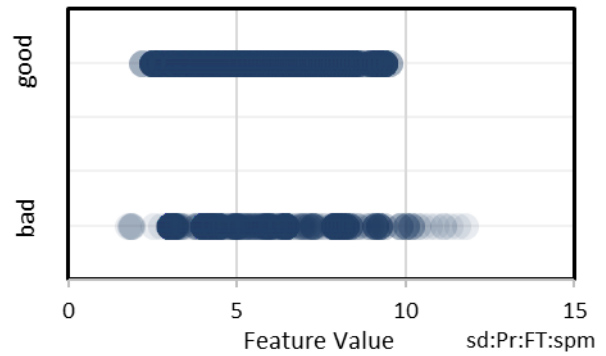


Figure 81: From stroke detector, raw sucker rod position, m

Fourier transform, strokes per minute

Raw sucker rod position in Figure 80, derived from Fourier transform with root mean square errors, has its bad measurements situated at values over 0,2 m.

Raw sucker rod position in Figure 81, derived from Fourier transform with strokes per minute, has its bad measurements situated over 10 m.

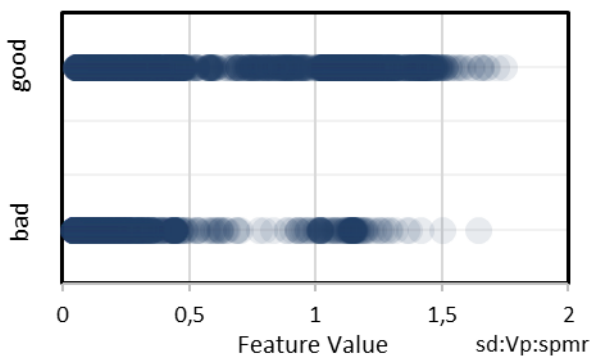


Figure 82: From stroke detector, processed sucker rod velocity, m/s

Strokes per minute ratio

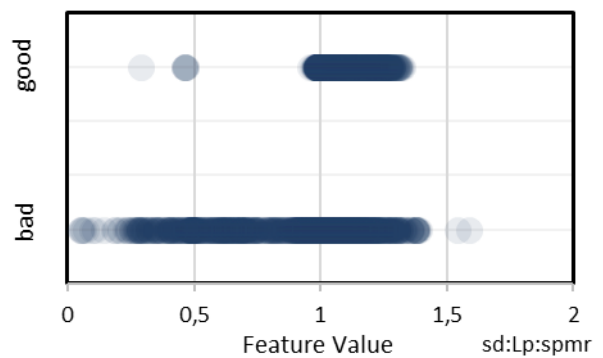


Figure 83: From stroke detector, processed sucker rod load, N

Strokes per minute ratio

Velocity derived from stroke detector (Figure 82) features with strokes per minute ratio has a poorer separation of good and bad data points, showing a slight range of values over 1,5 m/s as good.

For processed sucker rod load (Figure 83) derived from the same features, all values smaller than 1 N and larger than 1,4 N are bad.

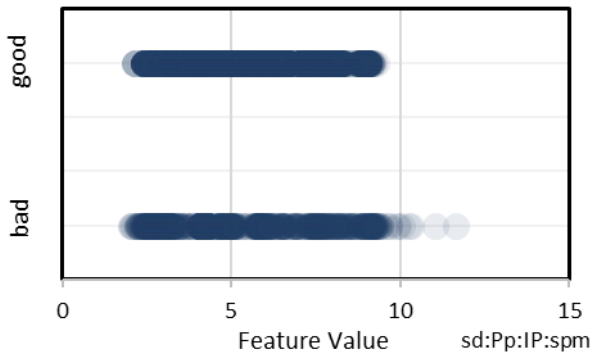


Figure 84: From stroke detector, processed sucker rod position, m

Instantaneous phase, strokes per minute

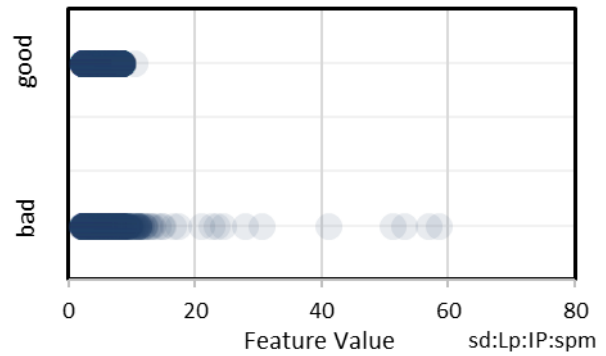


Figure 85: From stroke detector, processed sucker rod load, N

Instantaneous phase, strokes per minute

A similar case is for the processed sucker rod position (Figure 84), a few data points are located over 9 m, with few bad labelled data.

For the processed sucker rod load (Figure 85) features calculated with instantaneous phase and strokes per minute, all values larger than 12 N are bad.

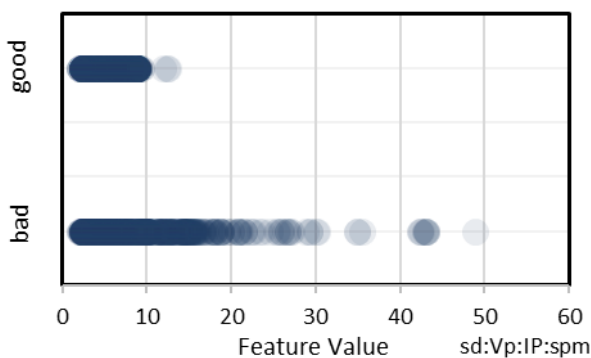


Figure 86: From stroke detector, processed sucker rod velocity, m/s

Instantaneous phase, strokes per minute

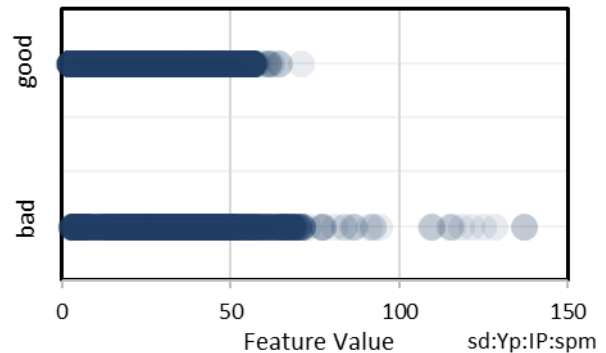


Figure 87: From stroke detector, processed sucker rod yank, N/s

Instantaneous phase, strokes per minute

Meanwhile, for the processed sucker rod velocity in Figure 86, features calculated with instantaneous phase and strokes per minute, all values larger than 10 m/s are bad.

Processed sucker rod yank (Figure 87) features obtained with instantaneous phase and strokes per minute have bad values in the range above 70 N/s.

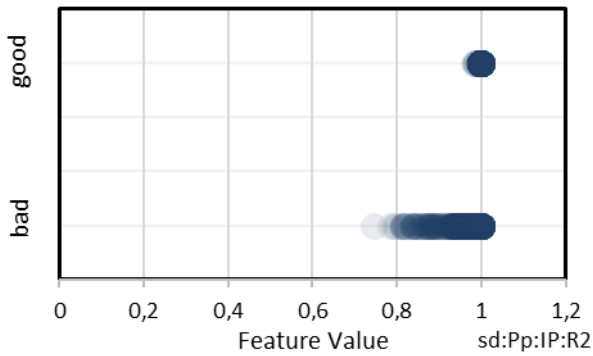


Figure 88: From stroke detector, processed sucker rod position, m

Instantaneous phase, coefficient of performance

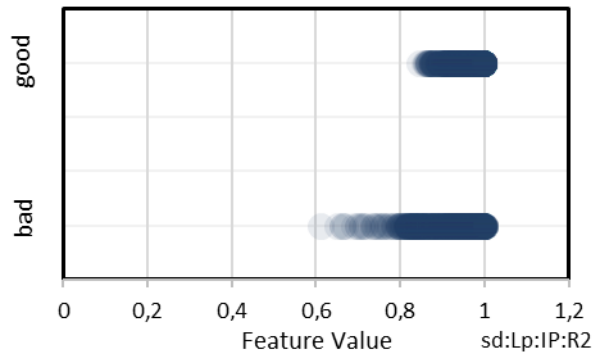


Figure 89: From stroke detector, processed sucker rod load, N

Instantaneous phase, coefficient of performance

Using the coefficient of performance, processed sucker rod position (Figure 88) has its bad values in the range below 0,9 m.

Processed sucker rod load paired (Figure 89) with the coefficient of performance has distinct bad data points under 0,7 N.

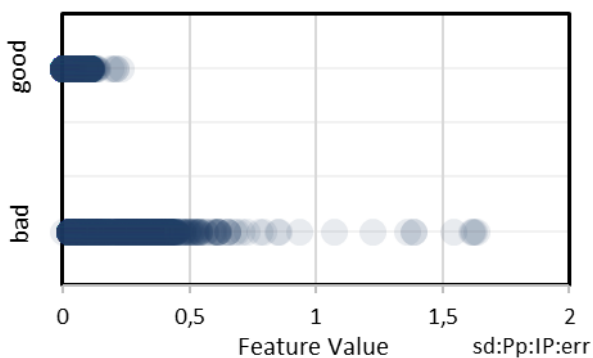


Figure 90: From stroke detector, processed sucker rod position, m

Instantaneous phase, root mean square error

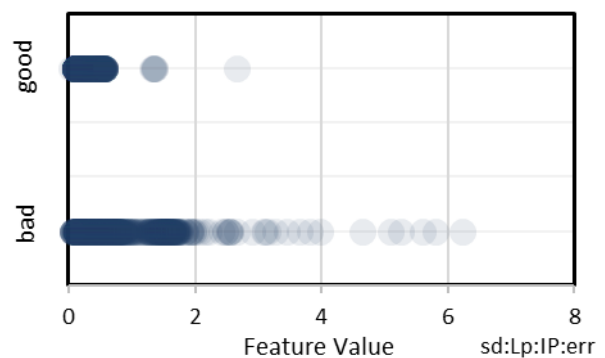


Figure 91: From stroke detector, processed sucker rod load, N

Instantaneous phase, root mean square error

Processed sucker rod position (Figure 90) features calculated with instantaneous phase and root mean square error have bad values above 0,25 m.

Whereas, processed sucker rod load (Figure 91) features obtained with instantaneous phase and root mean square error have their bad cards for values larger than 1 N.

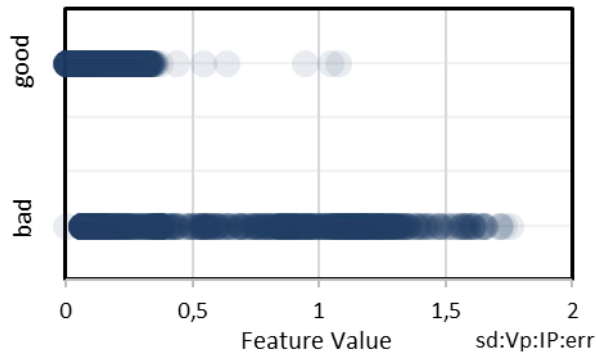


Figure 92: From stroke detector, processed sucker rod velocity, m/s

Instantaneous phase, root mean square error

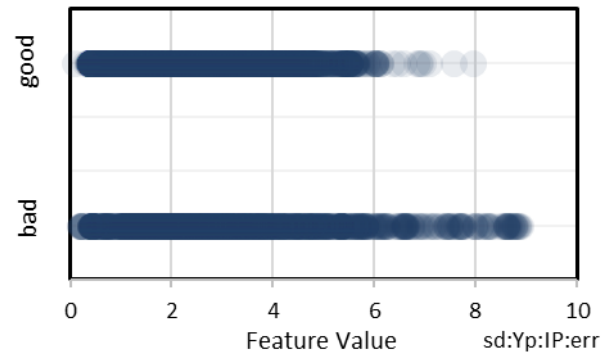


Figure 93: From stroke detector, processed sucker rod yank, N/s

Instantaneous phase, root mean square error

Processed sucker rod velocity (Figure 92) obtained with instantaneous phase and root mean square error identifies bad values for points over 0,3 m/s.

Same features for sucker rod yank (Figure 93) do not show any clear distinguishment for bad labels but there is a denser area for values over 6 N/s.

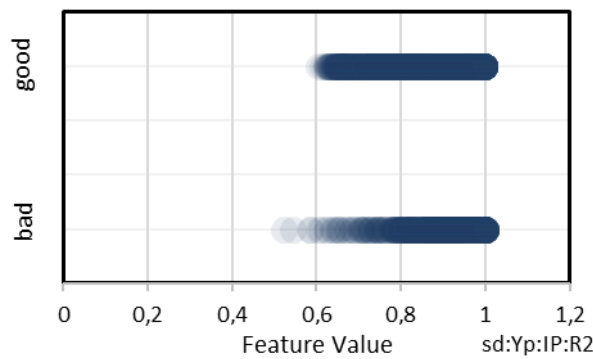


Figure 94: From stroke detector, processed sucker rod yank, N/s

Instantaneous phase, coefficient of performance

Processed sucker rod yank obtained with the coefficient of performance (Figure 94), however, doesn't show a good separation between the two labels.



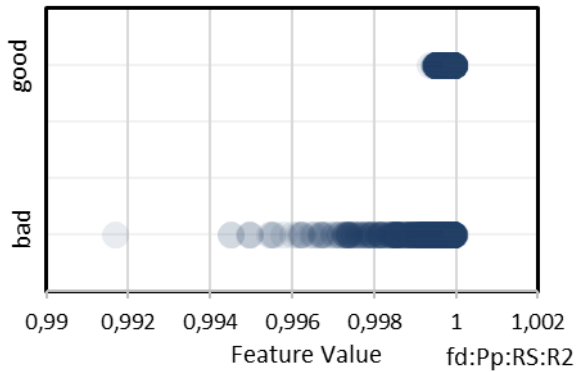


Figure 95: Fractal dimensions, processed sucker rod position, m

Rescaled range statistical analysis, coefficient of performance

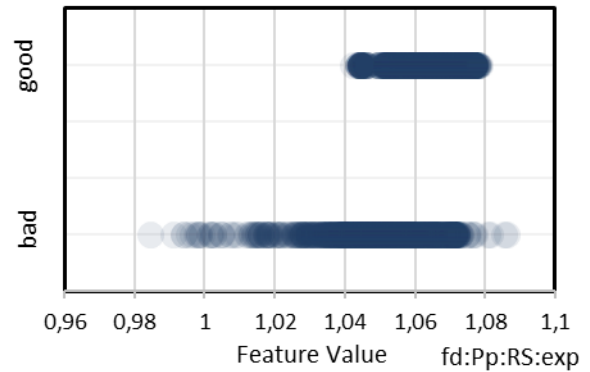


Figure 96: Fractal dimensions, processed sucker rod position, m

Rescaled range statistical analysis, Hurst exponent

Using fractal dimensions, rescaled range statistical analysis and the coefficient of performance, processed sucker rod position (Figure 95) features lower than 0,997 m are bad.

Processed sucker rod position (Figure 96) obtained with Hurst exponent has its bad data under 1,04 m.

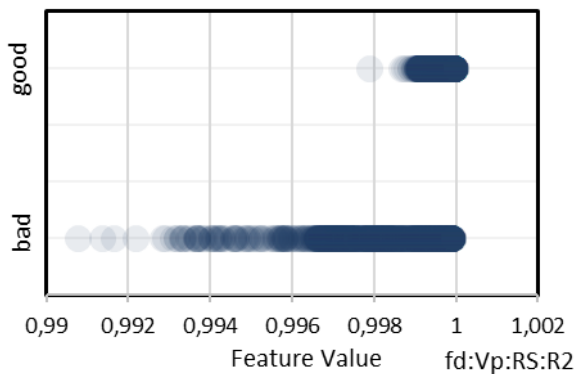


Figure 97: Fractal dimensions, processed sucker rod velocity, m/s

Rescaled range statistical analysis, coefficient of performance

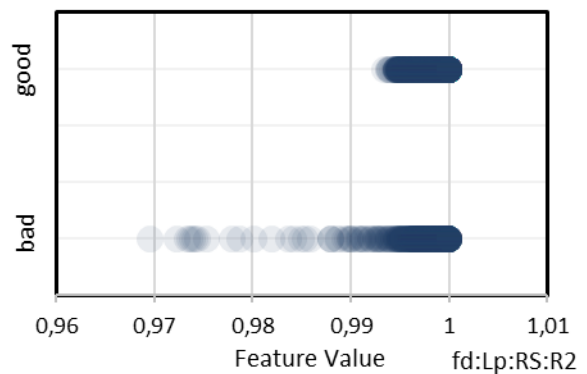


Figure 98: Fractal dimensions, processed sucker rod load, N

Rescaled range statistical analysis, coefficient of performance

Whereas, for velocity derived using rescaled range statistical analysis (Figure 97), it is easy to observe that values below 0,996 m/s are bad measurements.

Processed sucker rod load (Figure 98) features obtained with the coefficient of performance are bad if data points are smaller than 0,99 N.

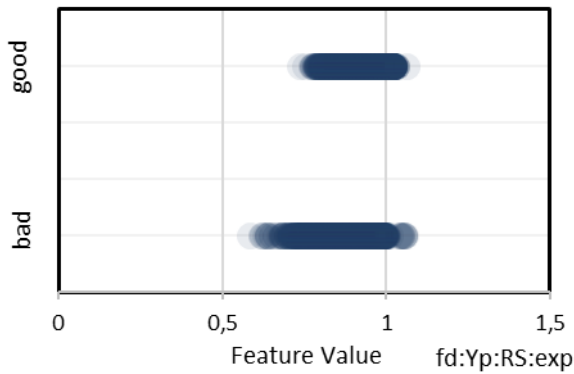


Figure 99: Fractal dimensions, processed sucker rod yank, N/s

Rescaled range statistical analysis, Hurst exponent

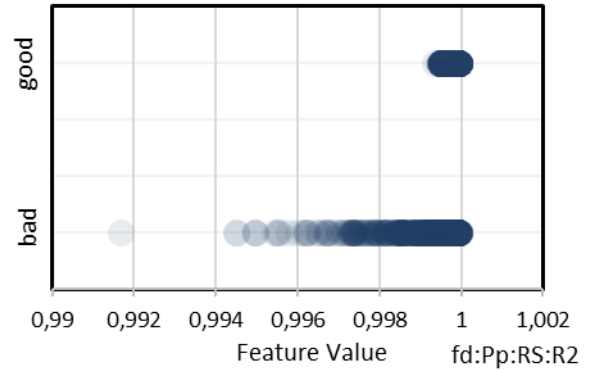


Figure 100: Fractal dimensions, processed sucker rod position, m

Rescaled range statistical analysis, coefficient of performance

For yank (Figure 99) features obtained with Hurst exponent, all values smaller than 0,35 N/s are linked to bad measurements.

Using fractal dimensions, rescaled range statistical analysis and the coefficient of performance, processed sucker rod position (Figure 100) features smaller than 0,997 m are bad.

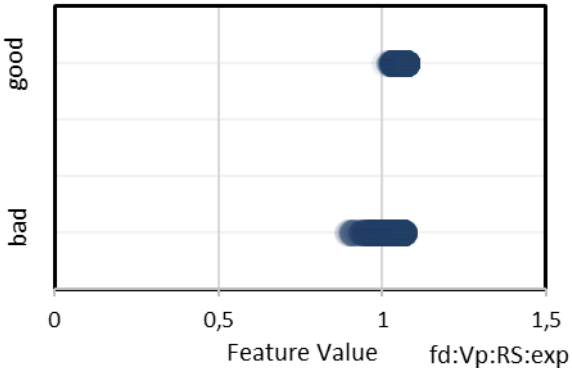


Figure 101: Fractal dimensions, processed sucker rod velocity, m/s

Rescaled range statistical analysis, Hurst exponent

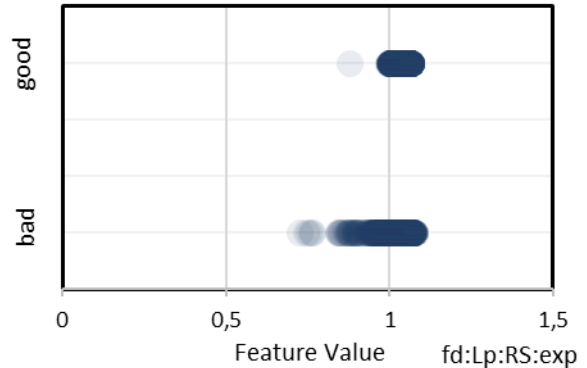


Figure 102: Fractal dimensions, processed sucker rod load, N

Rescaled range statistical analysis, Hurst exponent

Processed sucker rod velocity features (Figure 101) using Hurst exponent show that values smaller than 1 m/s are bad cards, whereas for the load (Figure 102) features all values smaller than 1, 2 N are bad.

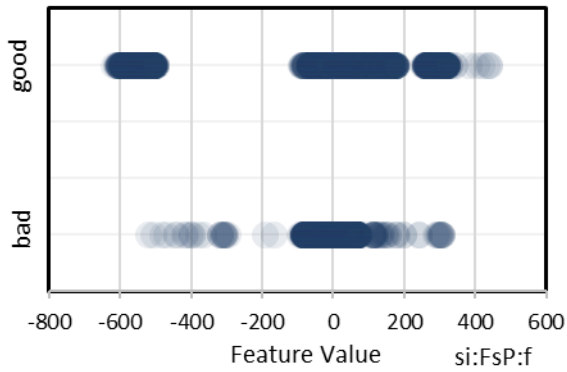


Figure 103: Full-stroke integral, processed sucker rod position, m

From standardized sucker rod load

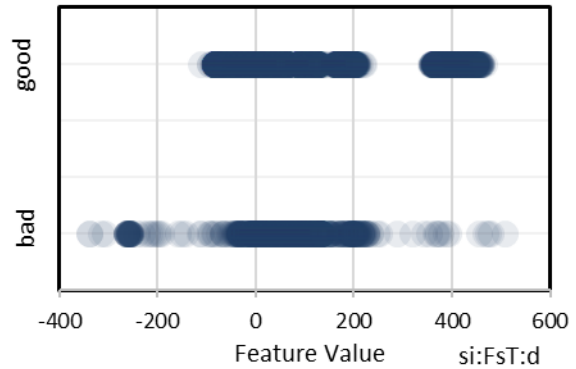


Figure 104: Full-stroke integral, sucker rod cycle time, s

From standardized sucker rod load

Using full- stroke integral and standardized sucker rod load, processed sucker rod position features (Figure 103) were calculated. Here one can see that although there are some data points for bad labels under 0 m, a denser frequency is noticed in values corresponding to the good label, as well as for values over 300 m.

Similar to processed sucker rod position obtained from down-stroke integral, sucker rod cycle time (Figure 104) also shows a good separation for good points, in particular for values larger than 220 s. Meanwhile, values smaller than -100 s are bad cards.

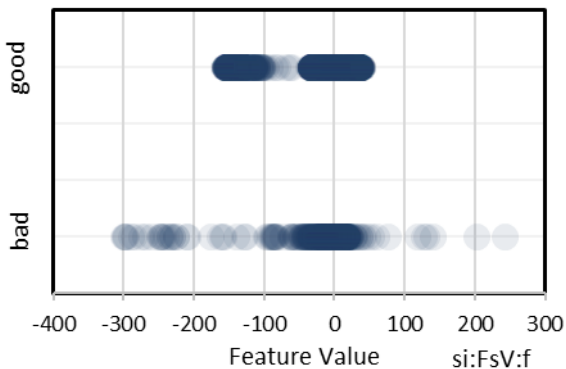


Figure 105: Full-stroke integral, processed sucker rod velocity, m/s

From standardized sucker rod load

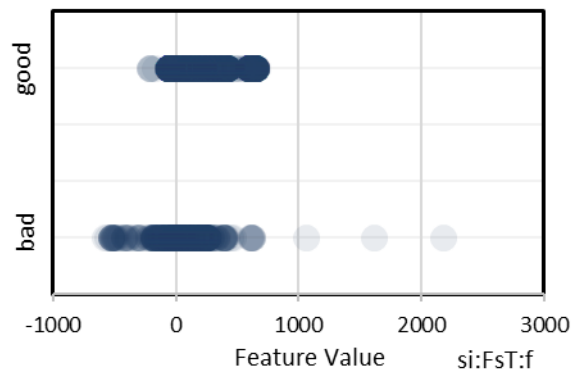


Figure 106: Full-stroke integral, sucker rod cycle time, s

From standardized sucker rod load

For processed sucker rod velocity in Figure 105, bad points are situated under -200 m/s and over 50 m/s.

The same features obtained from the full-stroke integral (Figure 106) show that bad values are represented by points smaller than -300 s and larger than 800 s.

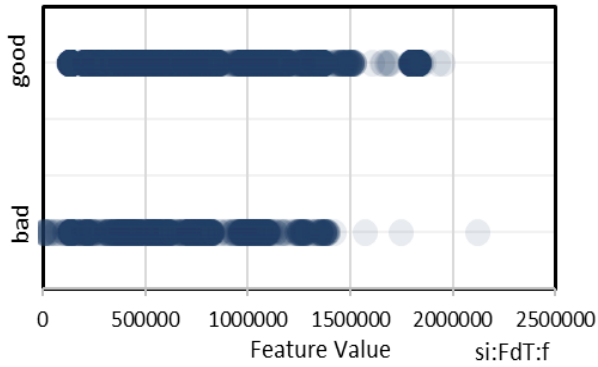


Figure 107: Full-stroke integral, sucker rod cycle time, s

From processed sucker rod load

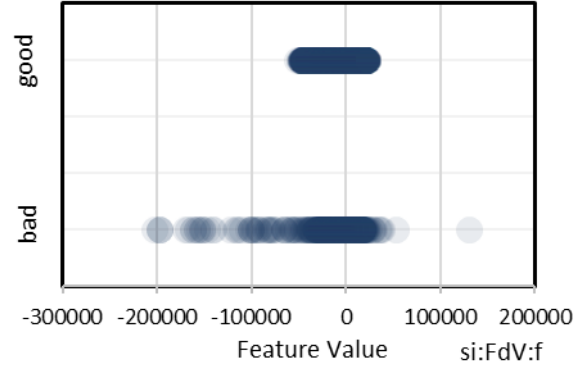


Figure 108: Full-stroke integral, processed sucker rod velocity, m/s

From processed sucker rod load

Processed sucker rod time (Figure 107) features obtained from the full- stroke integral have their good labelled data for values larger than 1500000 m/s.

For processed sucker rod velocity (Figure 108), all values smaller than -500000 m/s are bad.

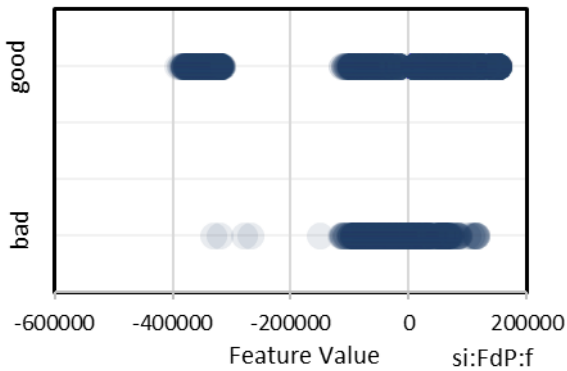


Figure 109: Full-stroke integral, processed sucker rod position, m

From processed sucker rod load

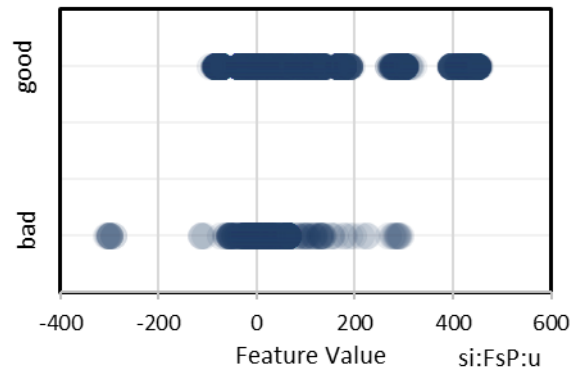


Figure 110: Up-stroke integral, processed sucker rod position, m

From standardized sucker rod load

For the processed sucker rod position (Figure 109), good values are situated under -300000 m.

For upstroke, the integral derived processed sucker rod position (Figure 110) features with values larger than 300 m are good and smaller than -100 m are bad.

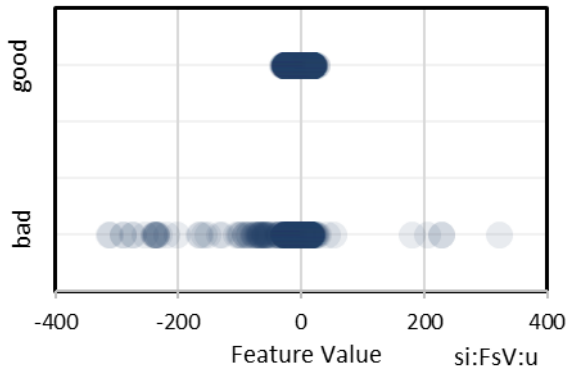


Figure 111: Up-stroke integral, processed sucker rod velocity, m/s

From standardized sucker rod load

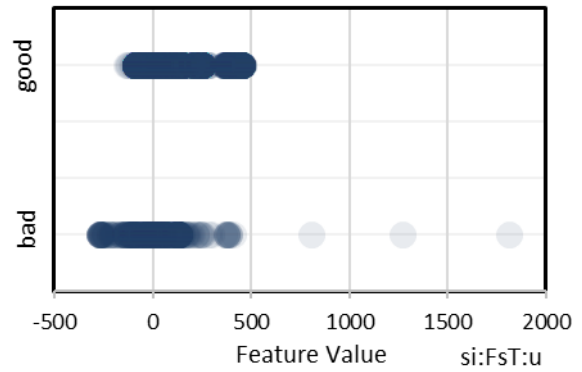


Figure 112: Up-stroke integral, sucker rod cycle time, s

From standardized sucker rod load

Processed sucker rod velocity (Figure 111) features obtained with the up-stroke integral and standardized sucker rod load, locate bad labels for values smaller than - 50 m/s and larger than 50 m/s.

For the rod cycle time (Figure 112) there are a few bad points for values above 500 s.

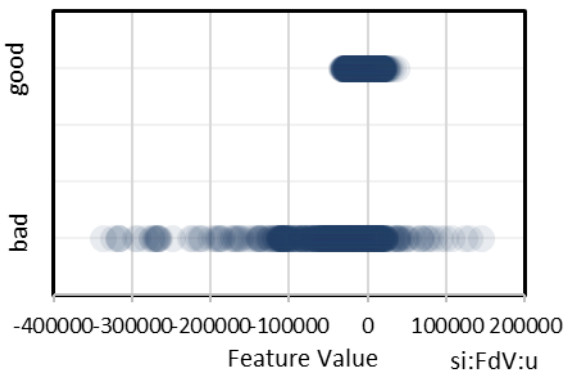


Figure 113: Up-stroke integral, processed sucker rod velocity, m/s

From processed sucker rod load

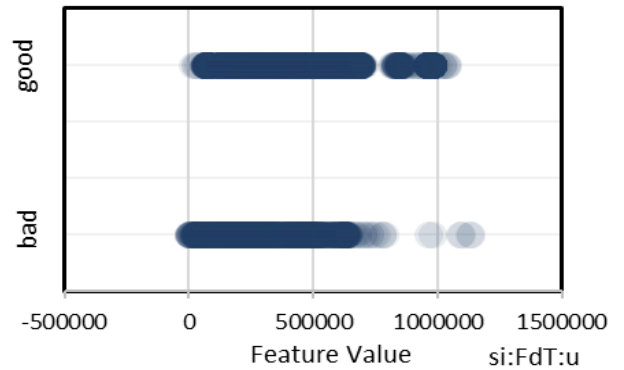


Figure 114: Up-stroke integral, sucker rod cycle time, s

From processed sucker rod load

Processed sucker rod velocity (Figure 113) obtained from processed sucker rod load places the bad values under -50000 m/s and over 50000 m/s.

A similar situation can be observed for the sucker rod cycle time (Figure 114); however, the density of bad values is very low over 1000000 s, therefore, it can be assumed that good values are larger than 7500000 s.

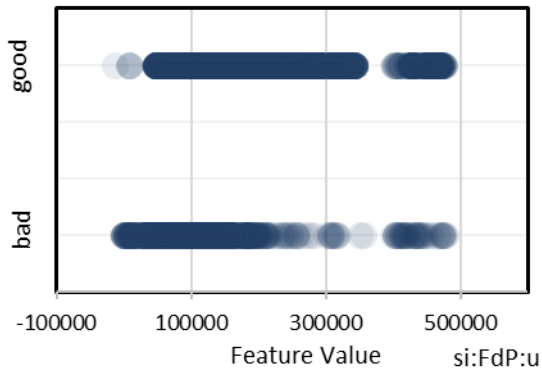


Figure 115: Up-stroke integral, processed sucker rod position, m

From processed sucker rod load

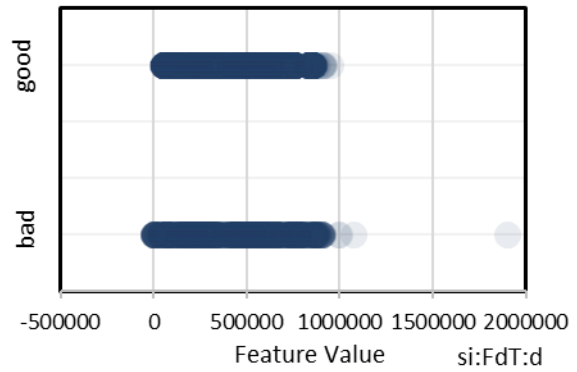


Figure 116: Down-stroke integral, sucker rod cycle time, s

From processed sucker rod load

For the processed sucker rod position (Figure 115) features, the distinguishment between bad and good related features is hard to define.

For sucker rod cycle time (Figure 116) features obtained using the down-stroke integral, values are overlapping, having a single bad data point around 2000000 s.

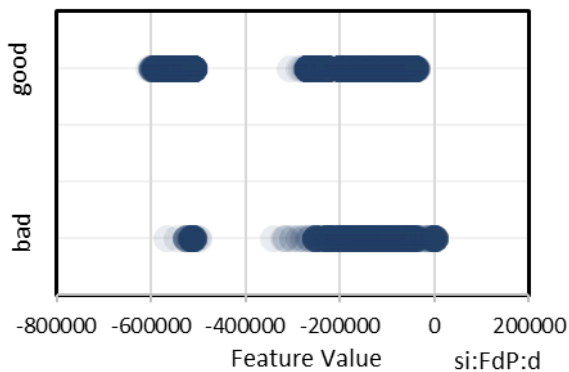


Figure 117: Down-stroke integral, processed sucker rod position, m

From processed sucker rod load

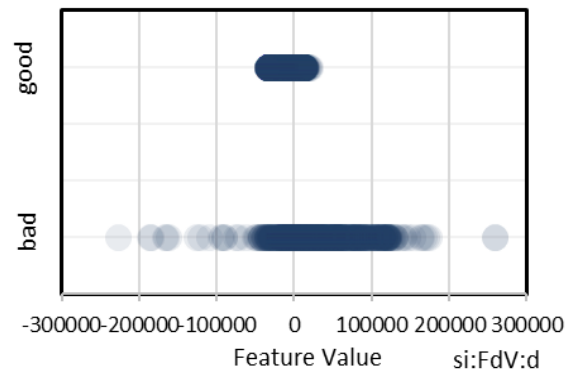


Figure 118: Down-stroke integral, processed sucker rod velocity, m/s

From processed sucker rod load

When using the down-stroke integral and processed sucker rod load, the values for processed sucker rod position (Figure 117) features are overlapping.

Sucker rod velocity features (Figure 118) obtained using down-stroke integral have their bad values smaller than -50000 m/s and larger than 4500000 m/s.

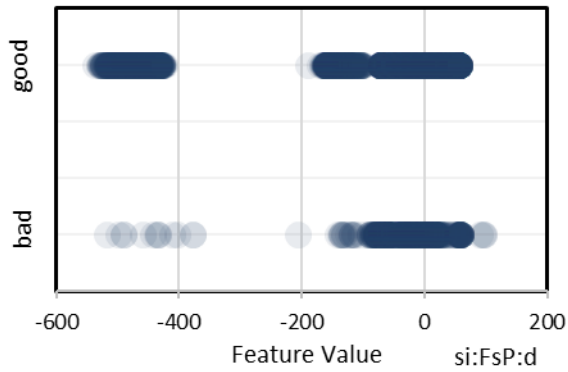


Figure 119: Down-stroke integral, processed sucker rod position, m

From standardized sucker rod load

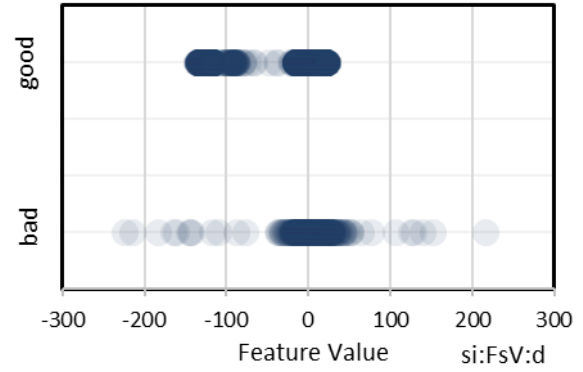


Figure 120: Down-stroke integral, processed sucker rod velocity, m/s

From standardized sucker rod load

For processed sucker rod position (Figure 119) obtained from standardized sucker rod load, all values smaller than -400 m are good.

For processed sucker rod velocity (Figure 120), all values smaller than -150 m/s and larger than 50 m/s are features for data labelled as bad quality.

### 6.3 Elliptic Fourier Transform Features

In order to recreate the shape of dynamometer cards, Elliptic Fourier Transform (EFT) was applied and used as another category of inputs to the neural network.

Fourier analysis converts a signal its original domain or space to frequency domain or the other way round.

The choice was made in favour of EFT as dynamometer cards are a closed contour and Fourier descriptors have a success story of use for closed contours characterization. They are invariant with contours' rotation, dilation and translation so no information about the shape is lost (Kuhl and Giardina 1982)

First step is to locate first two points, respectively:

$$x_p = \sum_{i=1}^p \Delta x_i \quad 8$$

$$y_p = \sum_{i=1}^p \Delta y_i \quad 9$$

Fourier series expansion for  $x$  projection is:

$$x(t) = A_o + \sum_{n=1}^{\infty} a_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T} \quad 10$$

Where:

$$A_o = \frac{1}{T} \int_0^T x(t) dt \quad 11$$

Fourier coefficients for the  $n$ th harmonic are:

$$a_n = \frac{2}{T} \int_0^T x(t) \cos \frac{2n\pi t}{T} dt \quad 12$$

$$b_n = \frac{2}{T} \int_0^T x(t) \sin \frac{2n\pi t}{T} dt \quad 13$$

After derivation:

$$a_n = \frac{T}{2n^2\pi^2} \sum_{p=1}^K \frac{\Delta x_p}{\Delta t_p} \left[ \cos \frac{2n\pi t_p}{T} - \cos \frac{2n\pi t_{p-1}}{T} \right] \quad 14$$

$$b_n = \frac{T}{2n^2\pi^2} \sum_{p=1}^K \frac{\Delta x_p}{\Delta t_p} \left[ \sin \frac{2n\pi t_p}{T} - \sin \frac{2n\pi t_{p-1}}{T} \right] \quad 15$$

Fourier series for  $y$  projection is:

$$y(t) = C_o + \sum_{n=1}^{\infty} c_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T} \quad 16$$

$$c = \frac{T}{2n^2\pi^2} \sum_{p=1}^K \frac{\Delta y_p}{\Delta t_p} \left[ \cos \frac{2n\pi t_p}{T} - \cos \frac{2n\pi t_{p-1}}{T} \right] \quad 17$$

$$d_n = \frac{T}{2n^2\pi^2} \sum_{p=1}^K \frac{\Delta y_p}{\Delta t_p} \left[ \sin \frac{2n\pi t_p}{T} - \sin \frac{2n\pi t_{p-1}}{T} \right] \quad 18$$

$A_o$  and  $C_o$  are DC components of Fourier series:

$$A_o = \frac{1}{T} \sum_{p=1}^K \frac{\Delta x_p}{2\Delta t_p} (t_p^2 - t_{p-1}^2) + \varepsilon_p (t_p - t_{p-1}) \quad 19$$



$$C_o = \frac{1}{T} \sum_{p=1}^K \frac{\Delta y_p}{2\Delta t_p} (t_p^2 - t_{p-1}^2) + \delta_p (t_p - t_{p-1}) \quad 20$$

Where:

$$\varepsilon_p = \sum_{j=1}^{p-1} \Delta x_j - \frac{\Delta x_p}{\Delta t_p} \sum_{j=1}^{p-1} \Delta t_j \quad 21$$

$$\delta_p = \sum_{j=1}^{p-1} \Delta y_j - \frac{\Delta y_p}{\Delta t_p} \sum_{j=1}^{p-1} \Delta t_j \quad 22$$

$$\varepsilon_1 = \delta_1 = 0 \quad 23$$

Number of harmonics in the Fourier approximation can be specified as:

$$X_N = A_0 + \sum_{n=1}^N a_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T} \quad 24$$

$$Y_N = C_0 + \sum_{n=1}^N c_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T} \quad 25$$

A truncated Fourier approximation of a closed contour can be written as

$$x(t) = A_0 + \sum_{n=1}^N X_n \quad 26$$

$$y(t) = C_0 + \sum_{n=1}^N Y_n \quad 27$$

$X_n, Y_n$  ( $1 \leq n \leq N$ ) are the components of the projections

$$X_n(t) = a_n \cos \frac{2\pi n t}{T} + b_n \sin \frac{2\pi n t}{T} \quad 28$$

$$Y_n(t) = c_n \cos \frac{2\pi n t}{T} + d_n \sin \frac{2\pi n t}{T} \quad 29$$

Points  $(X_n, Y_n)$  have elliptic loci and the Fourier conjecture to the original contour can be considered an inclusion in proper phase relationship of rotating phasors, which are described by the projections. Every rotating phasor has an elliptic locus and it rotates faster than the first

harmonic by its harmonic number  $n$ . The same elliptic loci will be obtained for the points  $(X_n, Y_n)$  despite the starting point of the contour (Kuhl and Giardina 1982).

We are reconstructing the shape of dynamometer cards from Fourier descriptors. Starting from the origin, the shape is reconstructed by adding all spatial components. We can go back to original shape by inversion. When DC and the first descriptor is included, we recreate a circle and by adding all descriptors, the circle becomes very close to the original dynamometer card shape.

Elliptic Fourier descriptors for absolute, real and imaginary parts represent the main part of calculated features. They maintain the ellipsis description in a two-dimensional space by considering that the image space defines a complex plane. Real part is represented by the first coordinate system, the imaginary part is the second co-ordinate system.

The amplitude and the phase have been identified. Amplitude is the square root of the squared real plus the squared imaginary values, phase is the arccos tangent of the imaginary value over the real value.

Figure 121 below illustrates the plotted output of calculated Fourier descriptors. Harmonic representations can be found in red. The recreation capability of the red contour can be increased by increasing the number of harmonics.

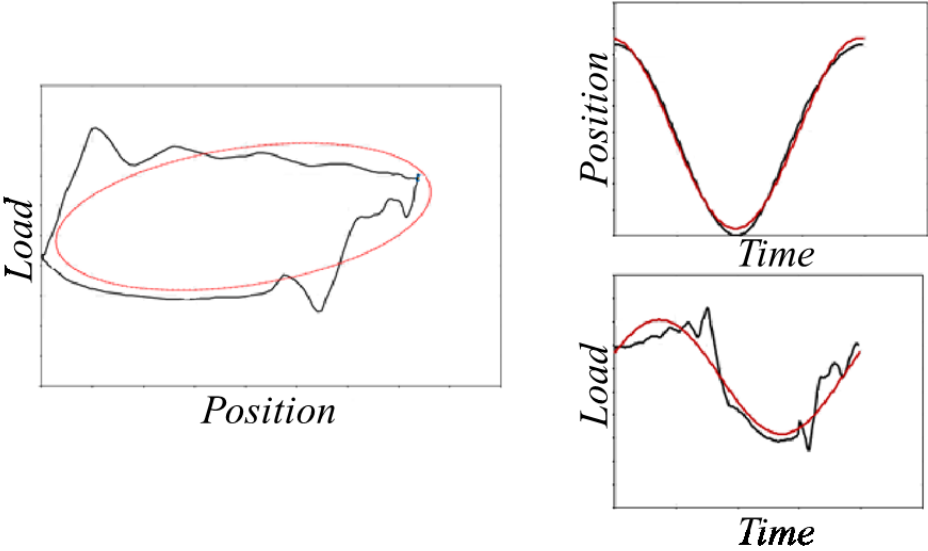


Figure 121: DC card representation with the first harmonic EFT component

The relationship between features' values related to each quality label has been plotted for Elliptic Fourier descriptors Figures 122 - 229. Each graph represents feature values for a single descriptor.  $Pr$  is the raw sucker rod position,  $Lr$  is the raw sucker rod load,  $lnA$  is the natural log of amplitude spectrum,  $ph$  is the unwrapped phase spectrum,  $Re$  is the real part of the spectrum,  $Im$  is the imaginary part of the spectrum.

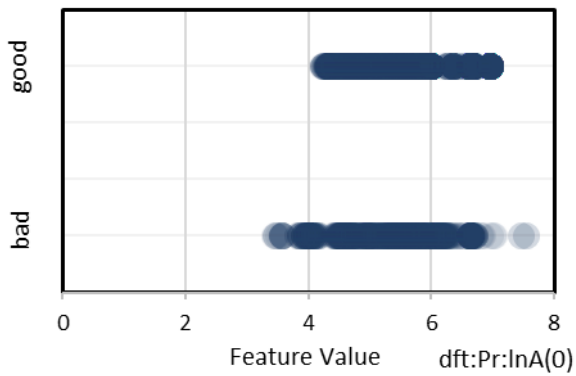


Figure 122: Descriptor (0), raw sucker rod position, natural log of amplitude spectrum

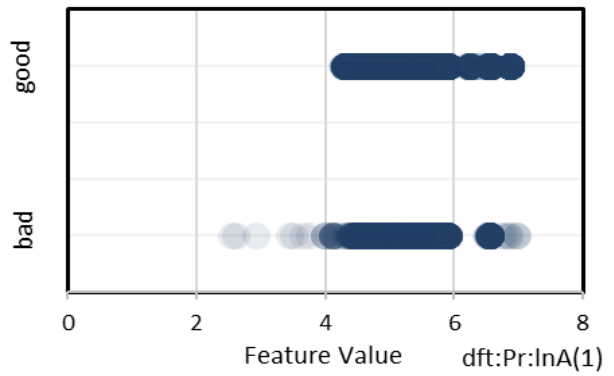


Figure 123: Descriptor (1), raw sucker rod position, natural log of amplitude spectrum

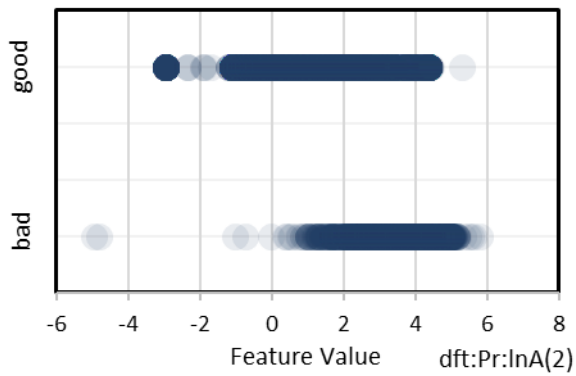


Figure 124: Descriptor (2), raw sucker rod position, natural log of amplitude spectrum

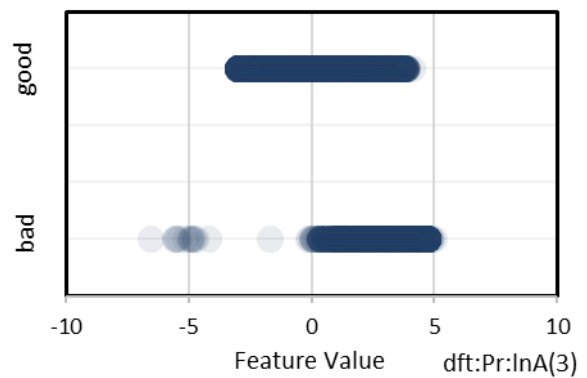


Figure 125: Descriptor (3), raw sucker rod position, natural log of amplitude spectrum

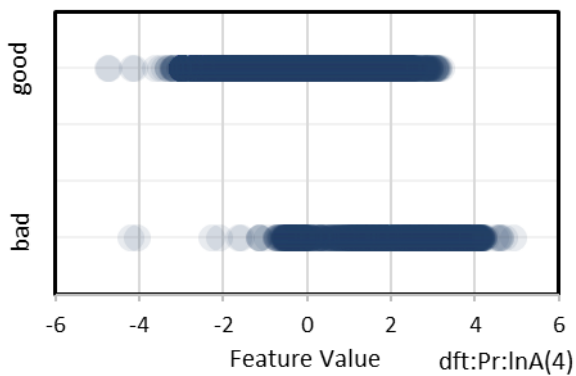


Figure 126: Descriptor (4), raw sucker rod position, natural log of amplitude spectrum

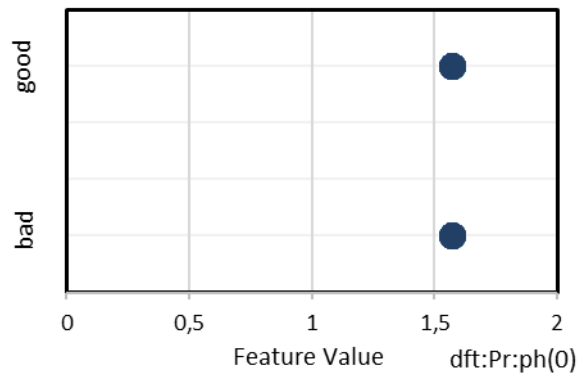


Figure 127: Descriptor (0), raw sucker rod position, unwrapped phase spectrum

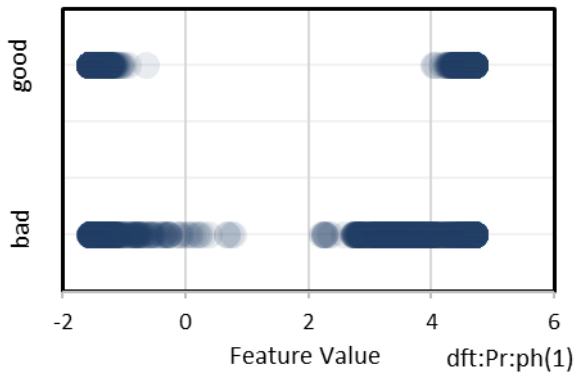


Figure 128: Descriptor (1), raw sucker rod position, unwrapped phase spectrum

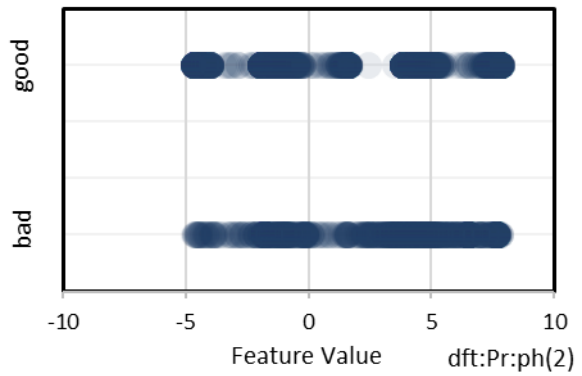


Figure 129: Descriptor (2), raw sucker rod position, unwrapped phase spectrum

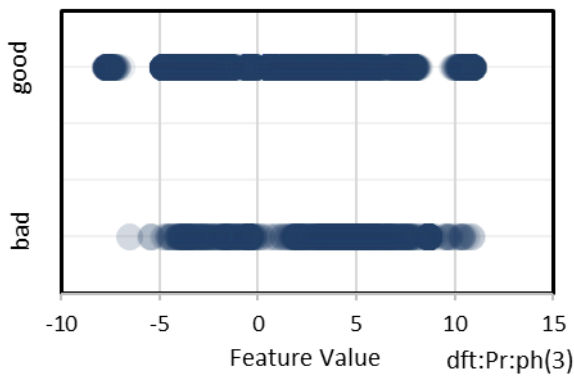


Figure 130: Descriptor (3), raw sucker rod position, unwrapped phase spectrum

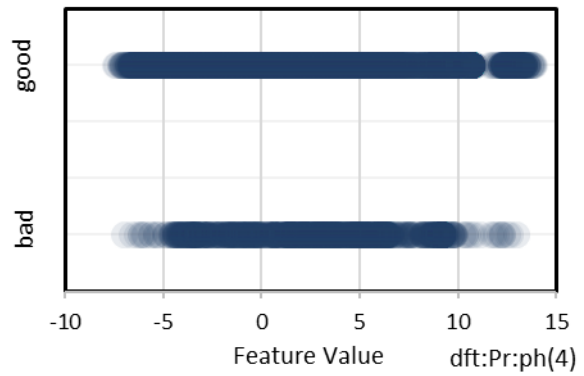


Figure 131: Descriptor (4), raw sucker rod position, unwrapped phase spectrum

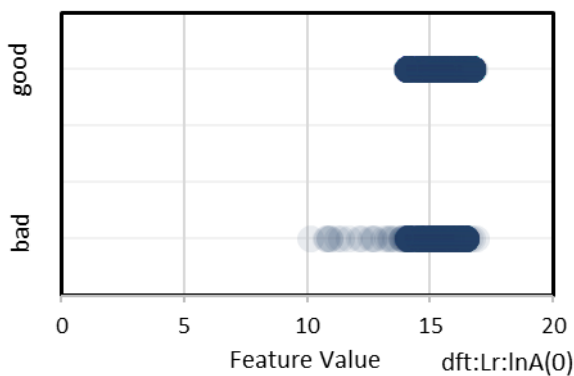


Figure 132: Descriptor (0), raw sucker rod load, natural log of amplitude spectrum

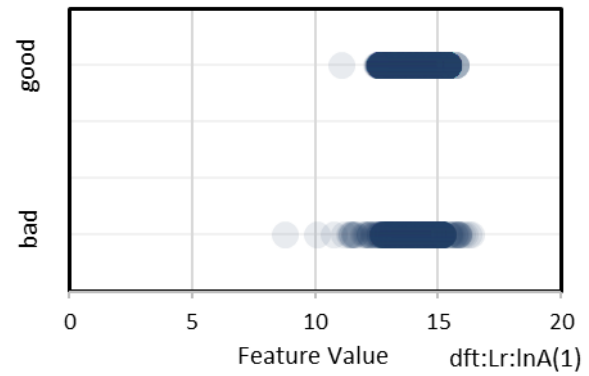


Figure 133: Descriptor (1), raw sucker rod load, natural log of amplitude spectrum

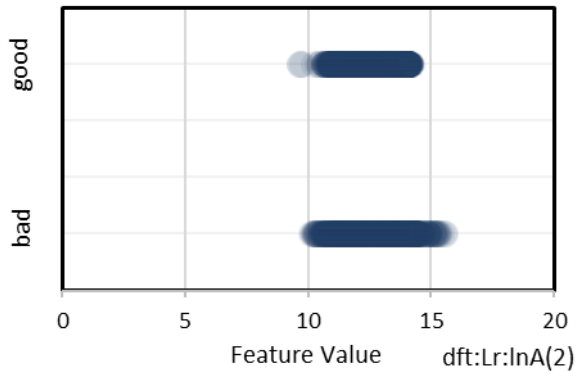


Figure 134: Descriptor (2), raw sucker rod load, natural log of amplitude spectrum

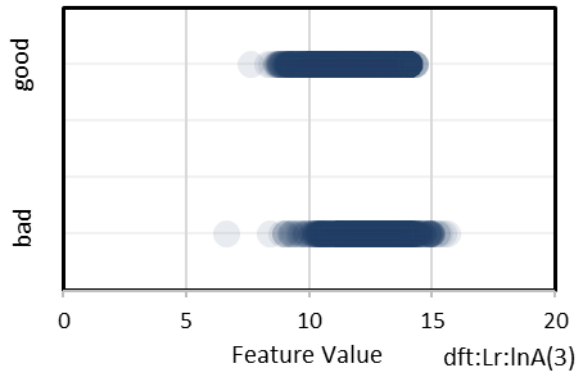


Figure 135: Descriptor (3), raw sucker rod load, natural log of amplitude spectrum

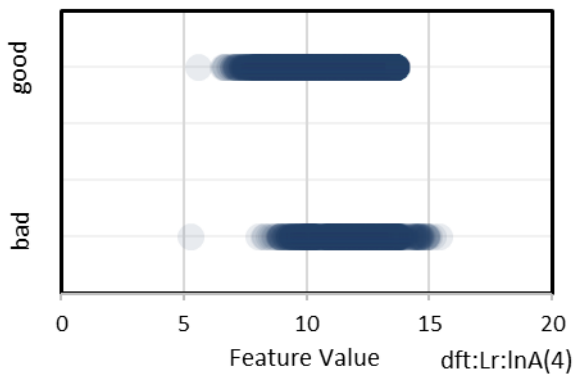


Figure 136: Descriptor (4), raw sucker rod load, natural log of amplitude spectrum

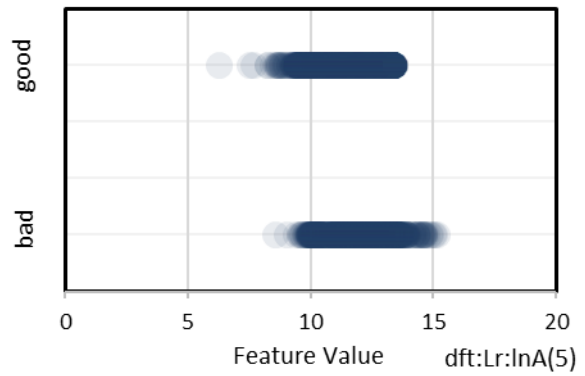


Figure 137: Descriptor (5), raw sucker rod load, natural log of amplitude spectrum

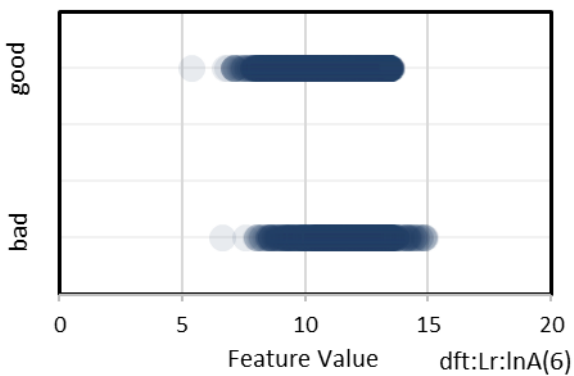


Figure 138: Descriptor (6), raw sucker rod load, natural log of amplitude spectrum

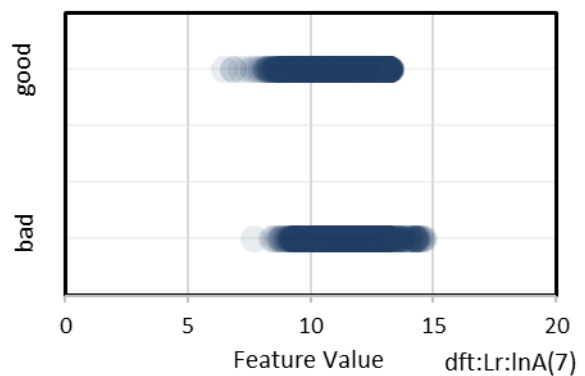


Figure 139: Descriptor (7), raw sucker rod load, natural log of amplitude spectrum

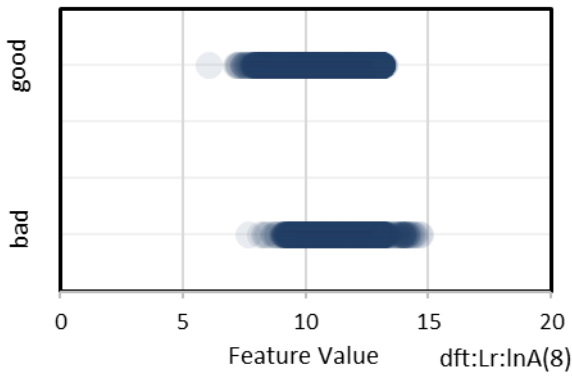


Figure 140: Descriptor (8), raw sucker rod load, natural log of amplitude spectrum

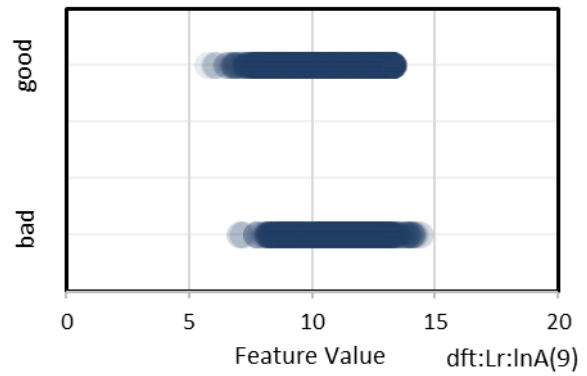


Figure 141: Descriptor (9), raw sucker rod load, natural log of amplitude spectrum

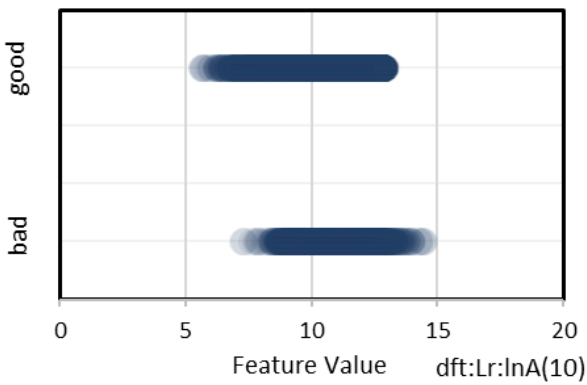


Figure 142: Descriptor (10), raw sucker rod load, natural log of amplitude spectrum

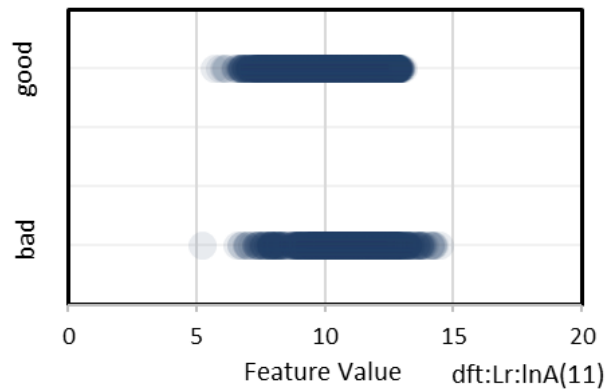


Figure 143: Descriptor (11), raw sucker rod load, natural log of amplitude spectrum

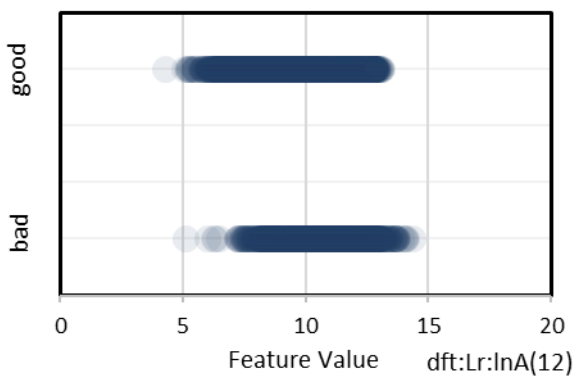


Figure 144: Descriptor (12), raw sucker rod load, natural log of amplitude spectrum

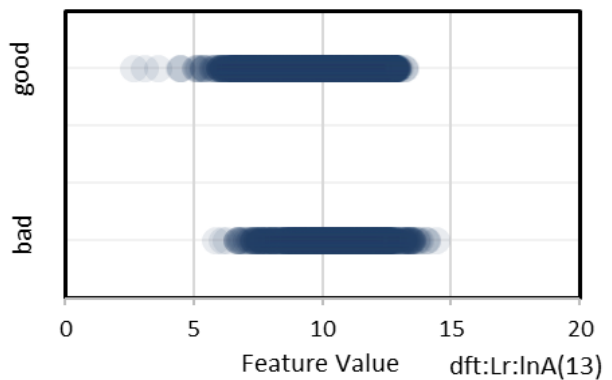


Figure 145: Descriptor (13), raw sucker rod load, natural log of amplitude spectrum

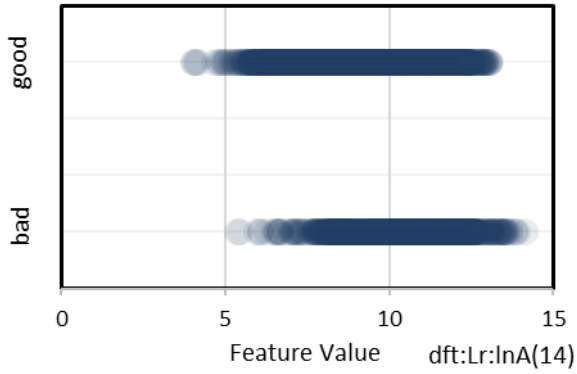


Figure 146: Descriptor (14), raw sucker rod load, natural log of amplitude spectrum

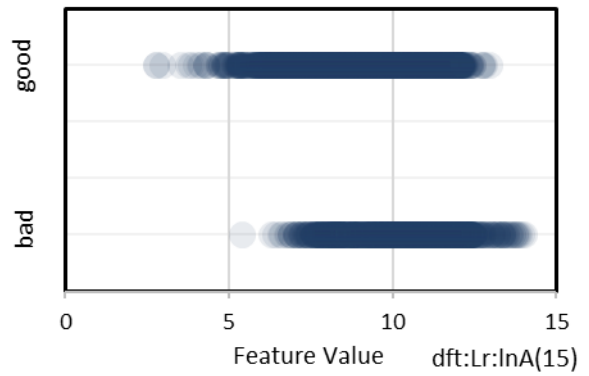


Figure 147: Descriptor (15), raw sucker rod load, natural log of amplitude spectrum

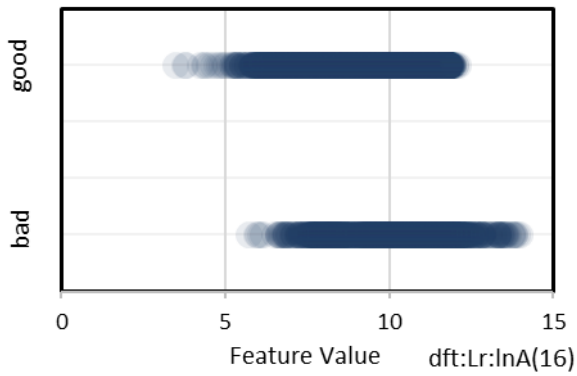


Figure 148: Descriptor (16), raw sucker rod load, natural log of amplitude spectrum



Figure 149: Descriptor (17), raw sucker rod load, natural log of amplitude spectrum

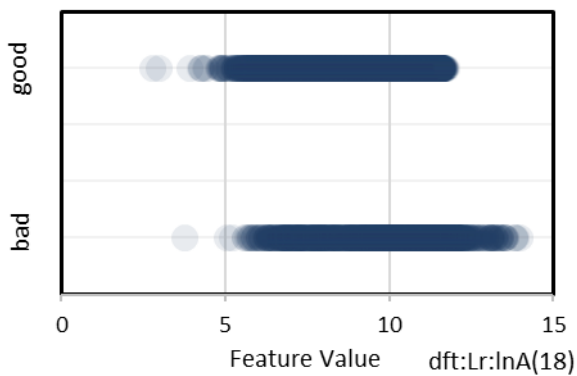


Figure 150: Descriptor (18), raw sucker rod load, natural log of amplitude spectrum

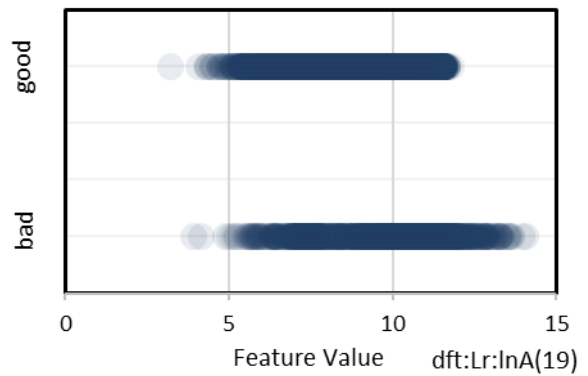


Figure 151: Descriptor (19), raw sucker rod load, natural log of amplitude spectrum

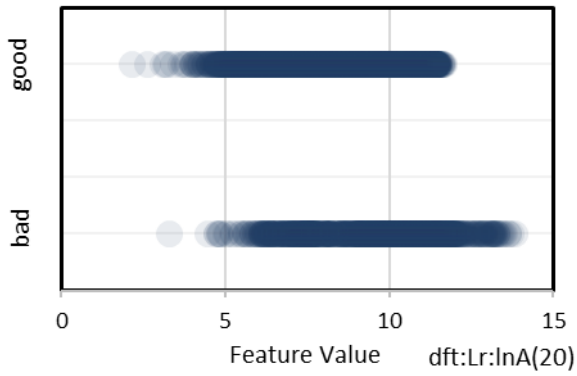


Figure 152: Descriptor (20), raw sucker rod load, natural log of amplitude spectrum

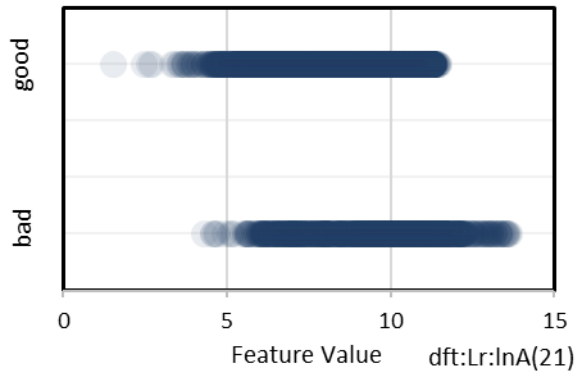


Figure 153: Descriptor (21), raw sucker rod load, natural log of amplitude spectrum

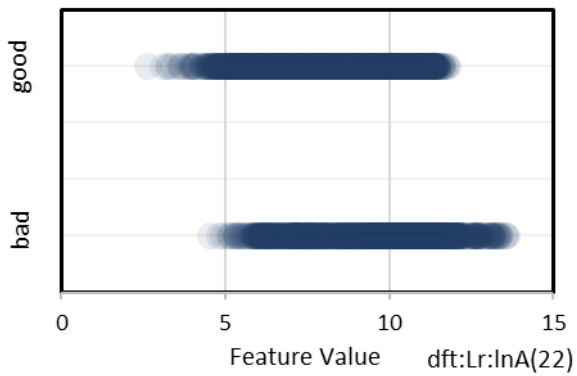


Figure 154: Descriptor (22), raw sucker rod load, natural log of amplitude spectrum

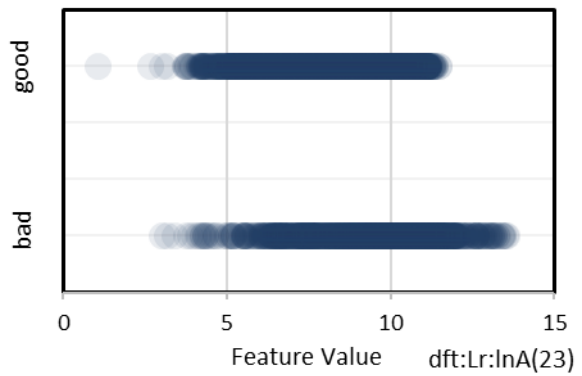


Figure 155: Descriptor (23), raw sucker rod load, natural log of amplitude spectrum



Figure 156: Descriptor (24), raw sucker rod load, natural log of amplitude spectrum



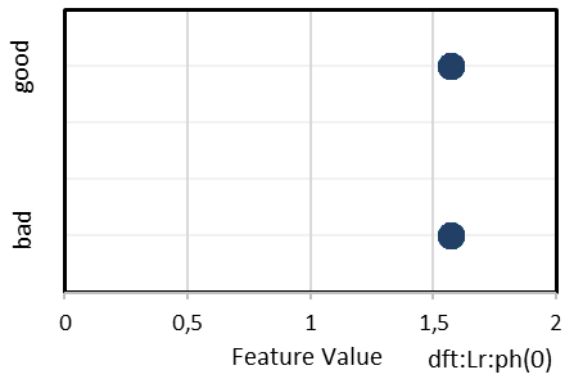


Figure 157: Descriptor (0), raw sucker rod load, unwrapped phase spectrum

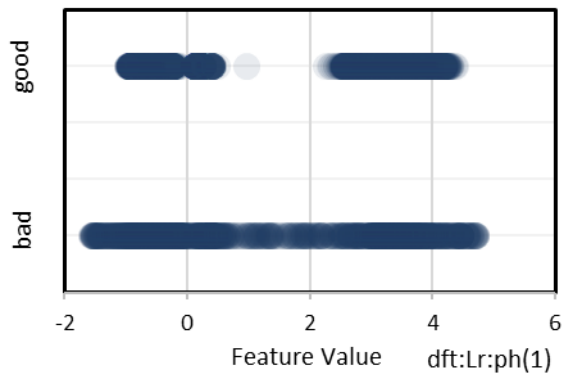


Figure 158: Descriptor (1), raw sucker rod load, unwrapped phase spectrum

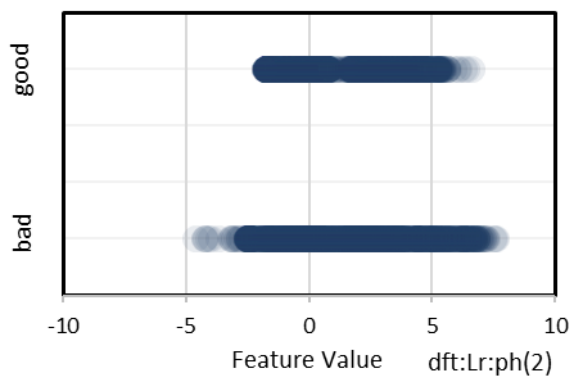


Figure 159: Descriptor (2), raw sucker rod load, unwrapped phase spectrum

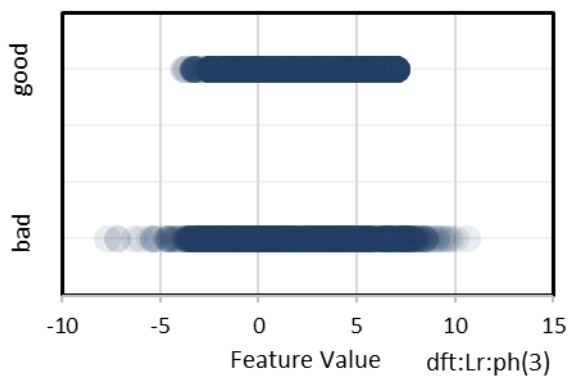


Figure 160: Descriptor (3), raw sucker rod load, unwrapped phase spectrum

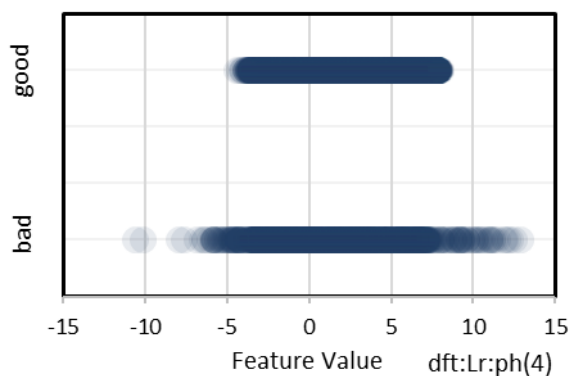


Figure 161: Descriptor (4), raw sucker rod load, unwrapped phase spectrum

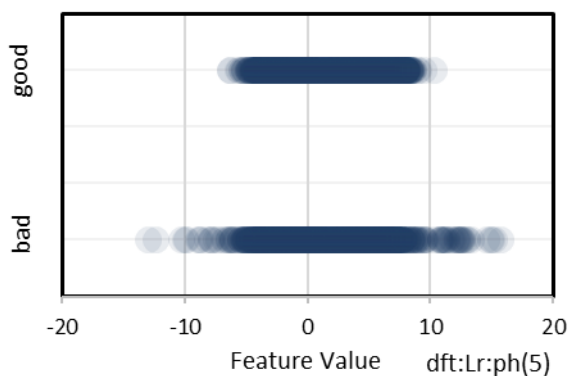


Figure 162: Descriptor (5), raw sucker rod load, unwrapped phase spectrum

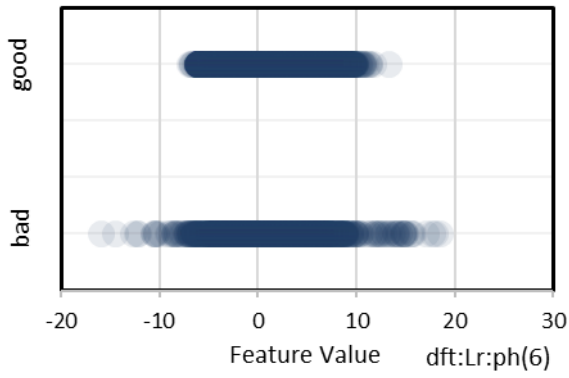


Figure 163: Descriptor (6), raw sucker rod load, unwrapped phase spectrum

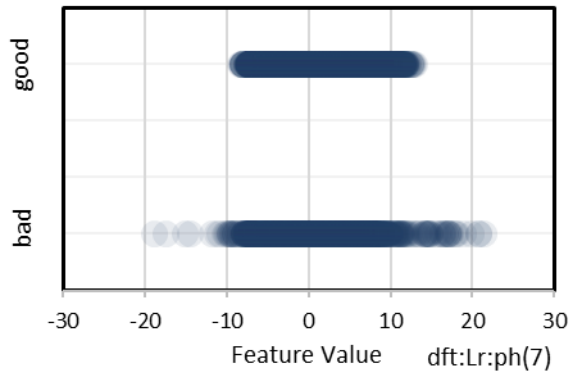


Figure 164: Descriptor (7), raw sucker rod load, unwrapped phase spectrum

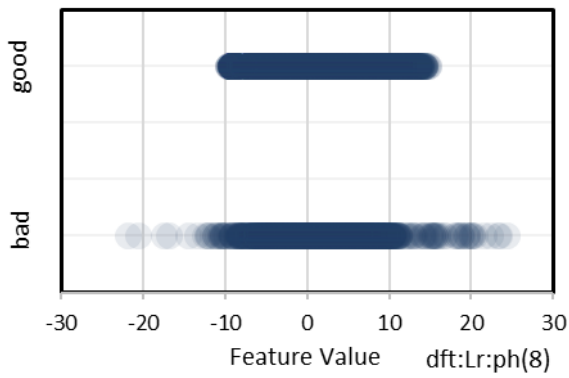


Figure 165: Descriptor (8), raw sucker rod load, unwrapped phase spectrum

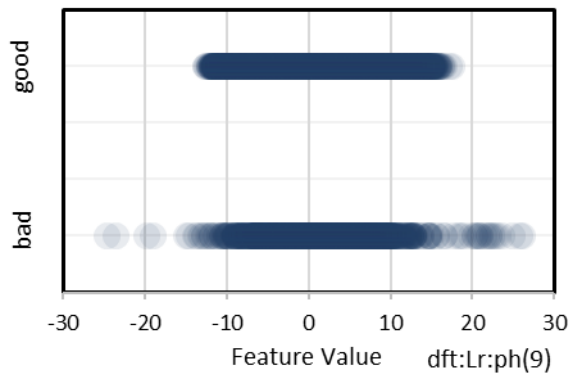


Figure 166: Descriptor (9), raw sucker rod load, unwrapped phase spectrum

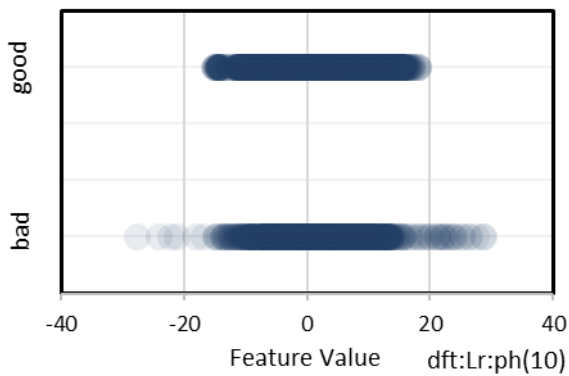


Figure 167: Descriptor (10), raw sucker rod load, unwrapped phase spectrum

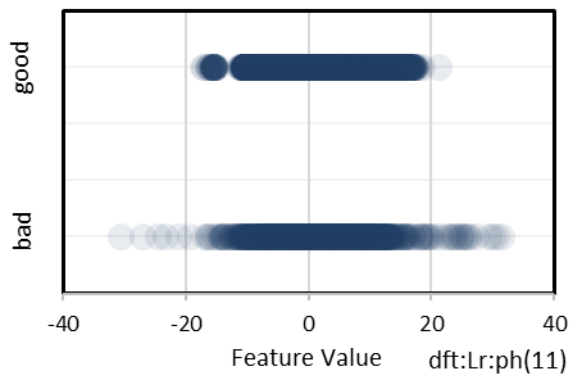


Figure 168: Descriptor (11), raw sucker rod load, unwrapped phase spectrum

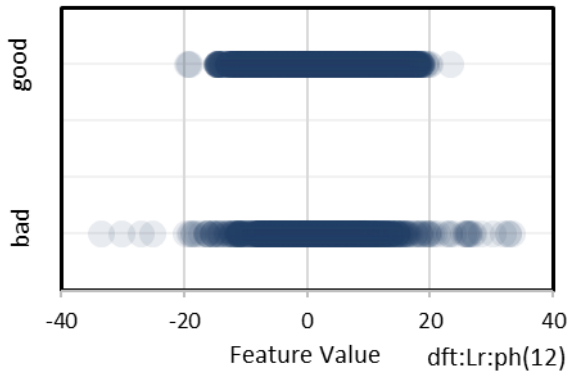


Figure 169: Descriptor (12), raw sucker rod load, unwrapped phase spectrum

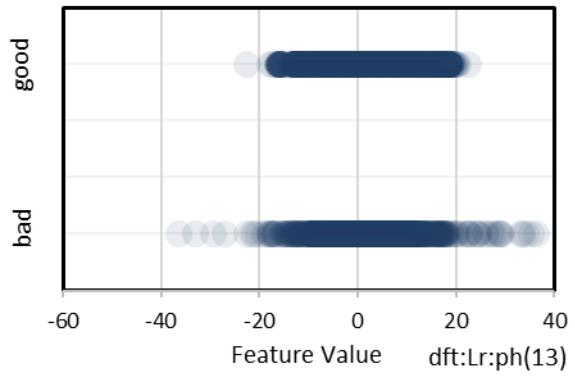


Figure 170: Descriptor (13), raw sucker rod load, unwrapped phase spectrum

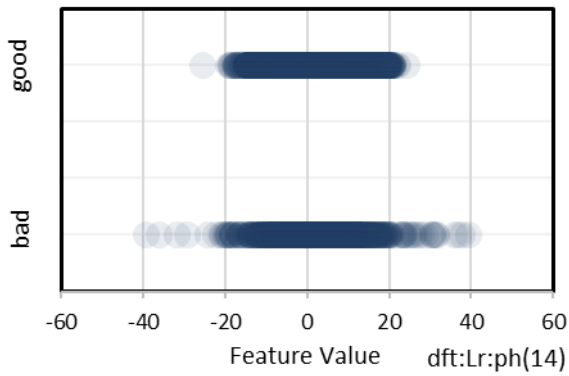


Figure 171: Descriptor (14), raw sucker rod load, unwrapped phase spectrum

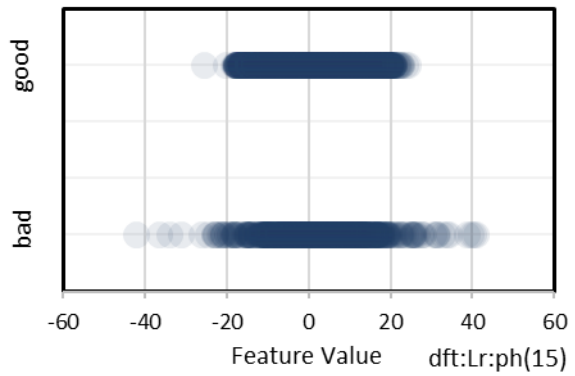


Figure 172: Descriptor (15), raw sucker rod load, unwrapped phase spectrum

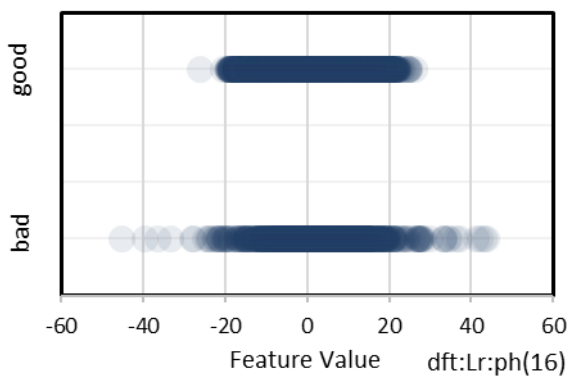


Figure 173: Descriptor (16), raw sucker rod load, unwrapped phase spectrum

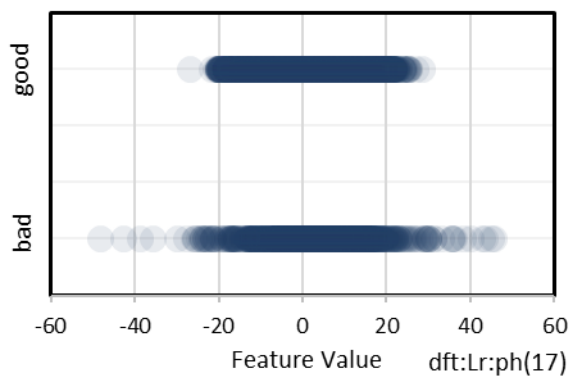


Figure 174: Descriptor (17), raw sucker rod load, unwrapped phase spectrum

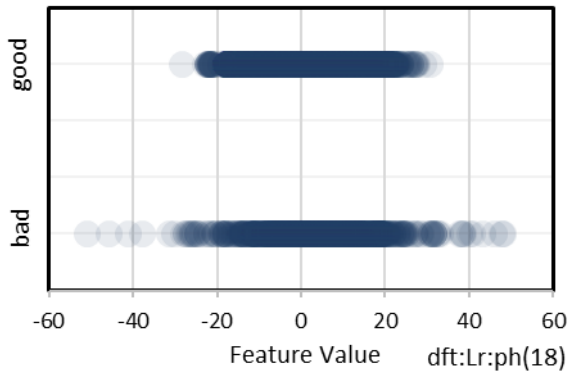


Figure 175: Descriptor (18), raw sucker rod load, unwrapped phase spectrum

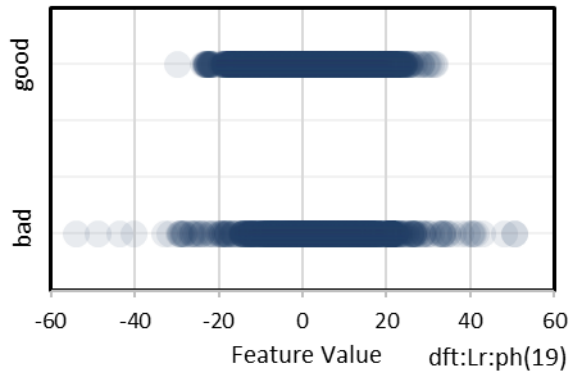


Figure 176: Descriptor (19), raw sucker rod load, unwrapped phase spectrum

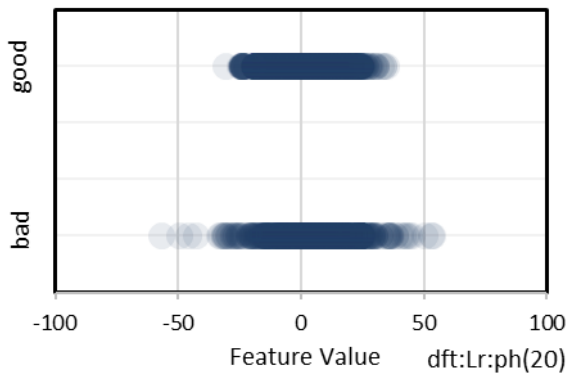


Figure 177: Descriptor (20), raw sucker rod load, unwrapped phase spectrum

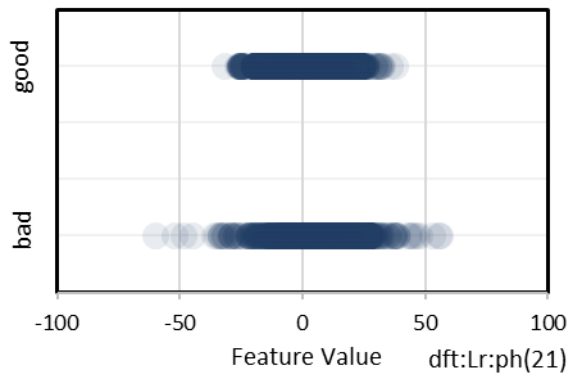


Figure 178: Descriptor (21), raw sucker rod load, unwrapped phase spectrum

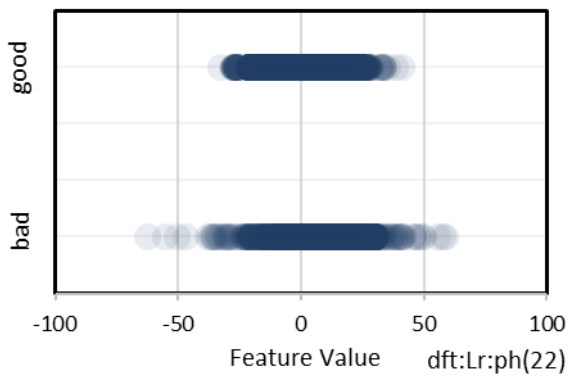


Figure 179: Descriptor (22), raw sucker rod load, unwrapped phase spectrum

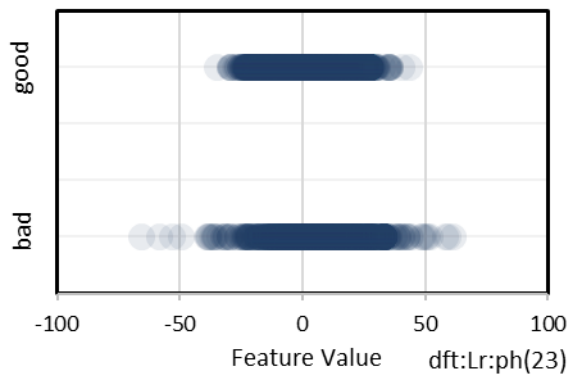


Figure 180: Descriptor (23), raw sucker rod load, unwrapped phase spectrum

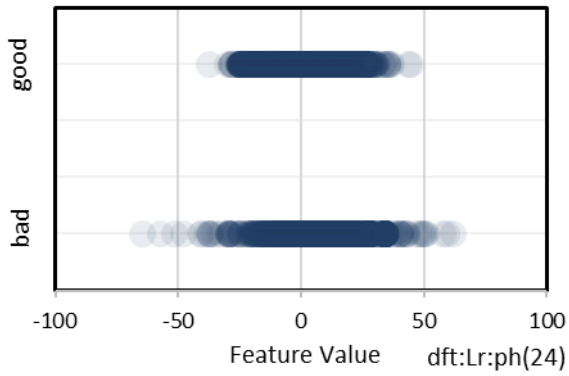


Figure 181: Descriptor (24), raw sucker rod load, unwrapped phase spectrum

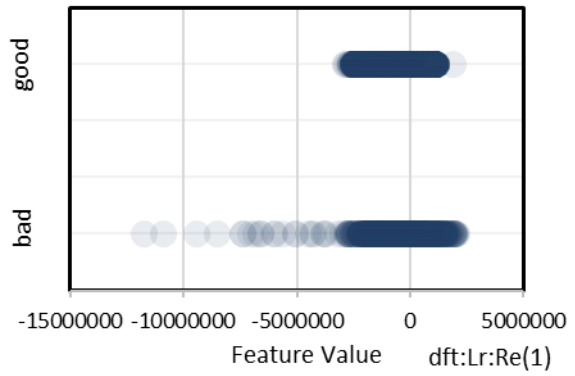


Figure 182: Descriptor (1), raw sucker rod load, real part of the spectrum

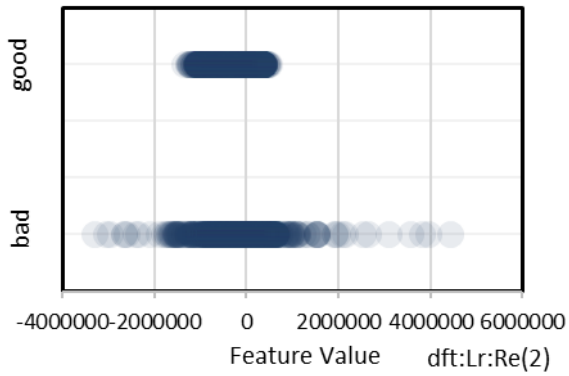


Figure 183: Descriptor (2), raw sucker rod load, real part of the spectrum

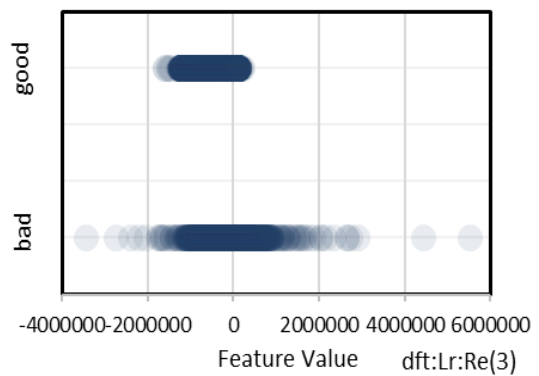


Figure 184: Descriptor (3), raw sucker rod load, real part of the spectrum

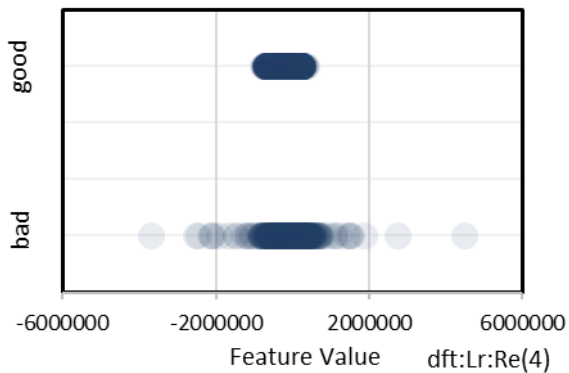


Figure 185: Descriptor (4), raw sucker rod load, real part of the spectrum

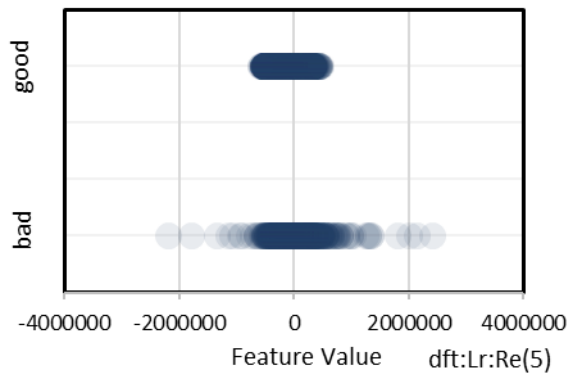


Figure 186: Descriptor (5), raw sucker rod load, real part of the spectrum

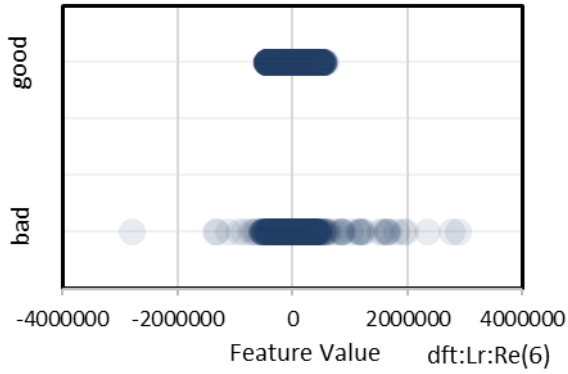


Figure 187: Descriptor (6), raw sucker rod load, real part of the spectrum

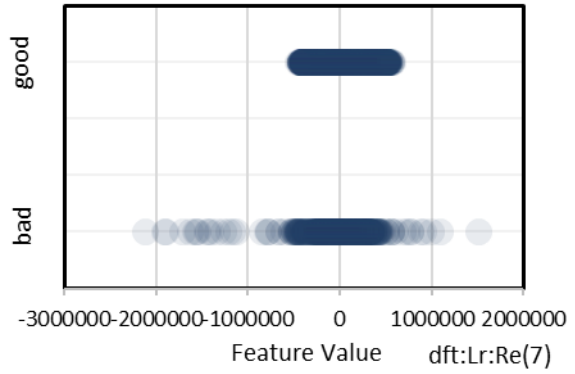


Figure 188: Descriptor (7), raw sucker rod load, real part of the spectrum

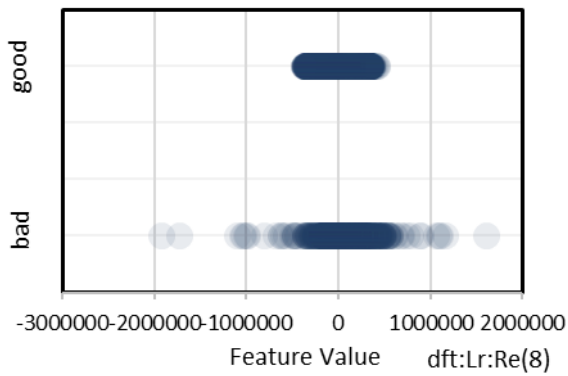


Figure 189: Descriptor (8), raw sucker rod load, real part of the spectrum

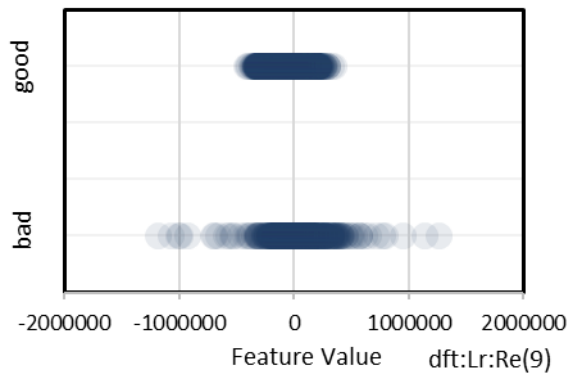


Figure 190: Descriptor (9), raw sucker rod load, real part of the spectrum

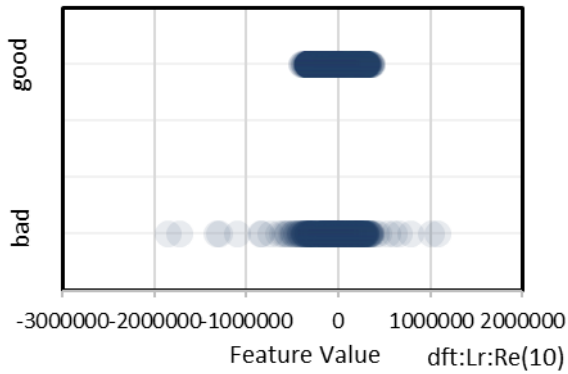


Figure 191: Descriptor (10), raw sucker rod load, real part of the spectrum

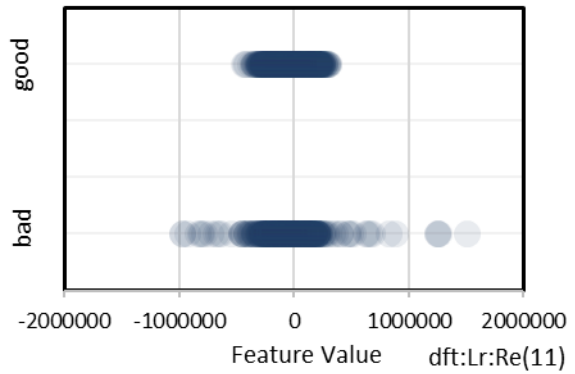


Figure 192: Descriptor (11), raw sucker rod load, real part of the spectrum

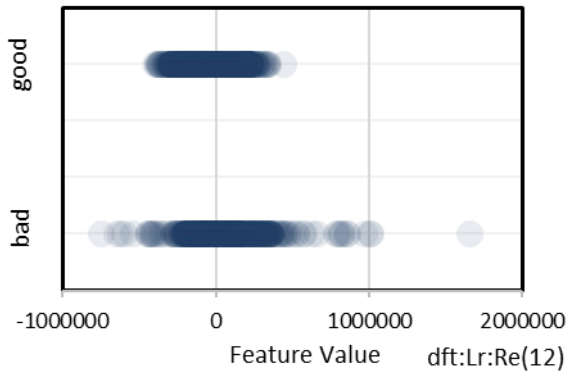


Figure 193: Descriptor (12), raw sucker rod load, real part of the spectrum

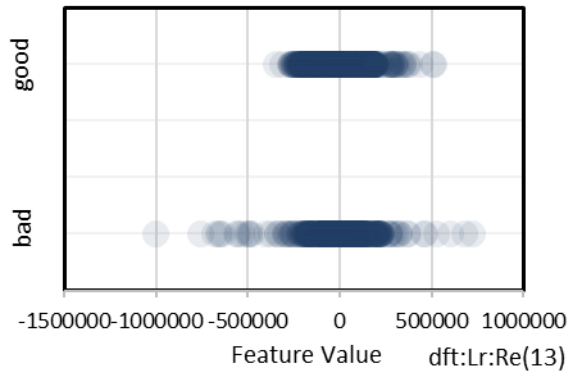


Figure 194: Descriptor (13), raw sucker rod load, real part of the spectrum

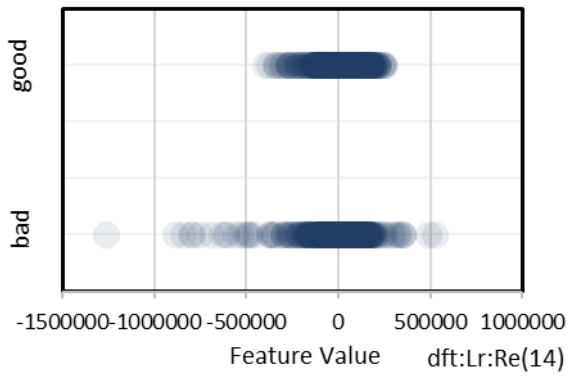


Figure 195: Descriptor (14), raw sucker rod load, real part of the spectrum

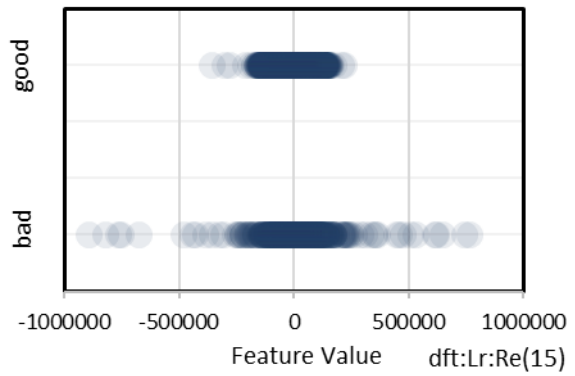


Figure 196: Descriptor (15), raw sucker rod load, real part of the spectrum

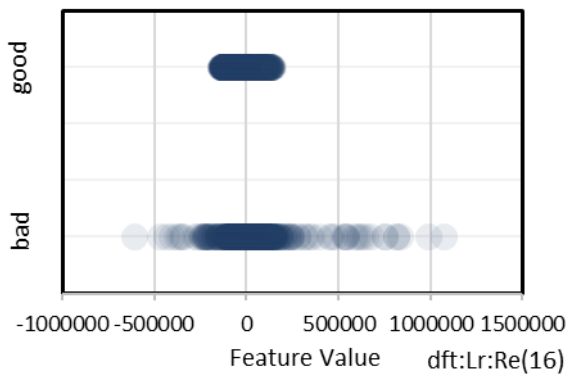


Figure 197: Descriptor (16), raw sucker rod load, real part of the spectrum

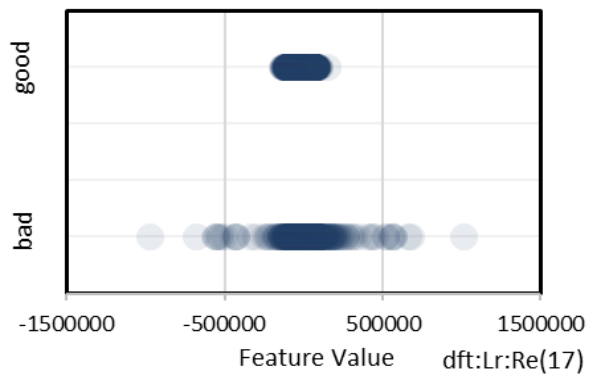


Figure 198: Descriptor (17), raw sucker rod load, real part of the spectrum

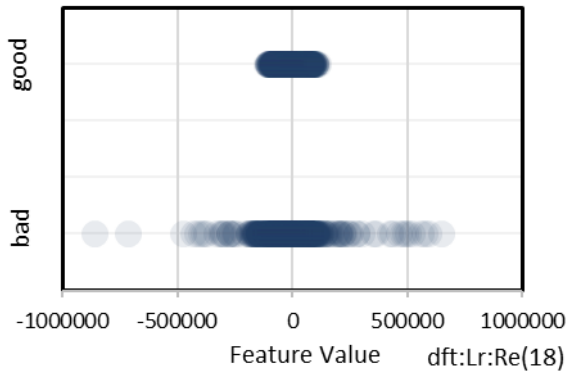


Figure 199: Descriptor (18), raw sucker rod load, real part of the spectrum

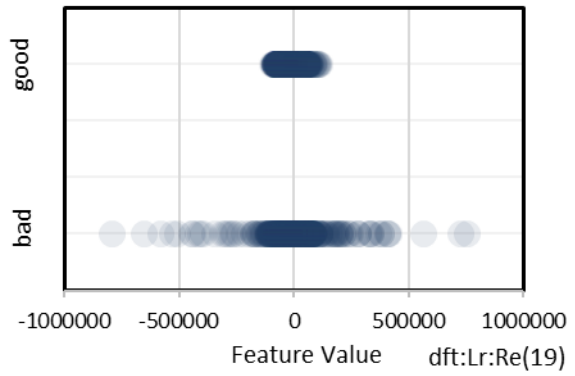


Figure 200: Descriptor (19), raw sucker rod load, real part of the spectrum

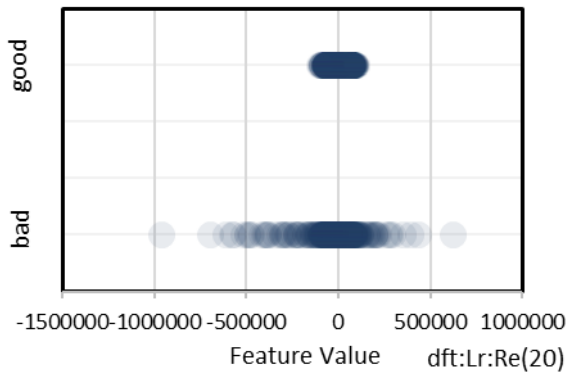


Figure 201: Descriptor (20), raw sucker rod load, real part of the spectrum

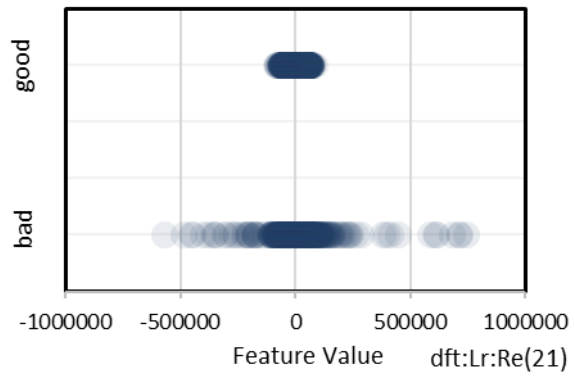


Figure 202: Descriptor (21), raw sucker rod load, real part of the spectrum

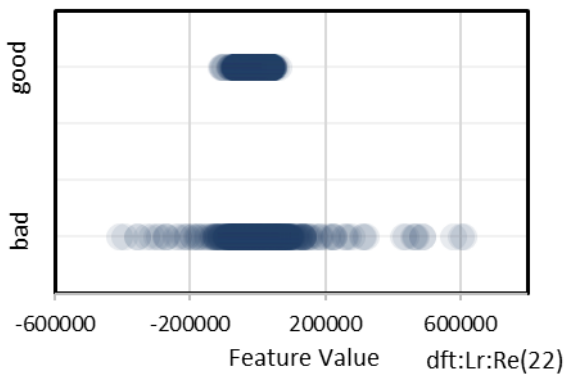


Figure 203: Descriptor (22), raw sucker rod load, real part of the spectrum

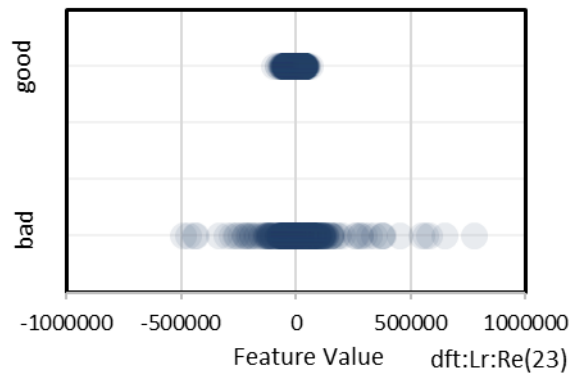


Figure 204: Descriptor (23), raw sucker rod load, real part of the spectrum



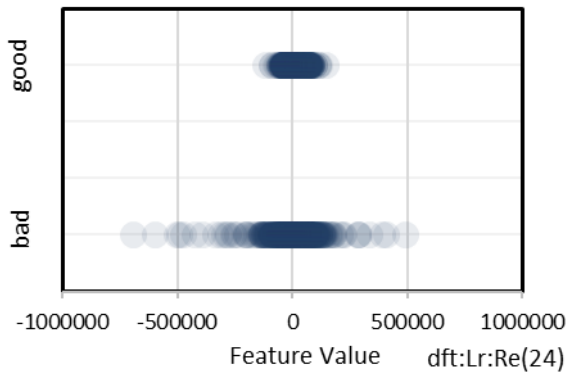


Figure 205: Descriptor (24), raw sucker rod load, real part of the spectrum

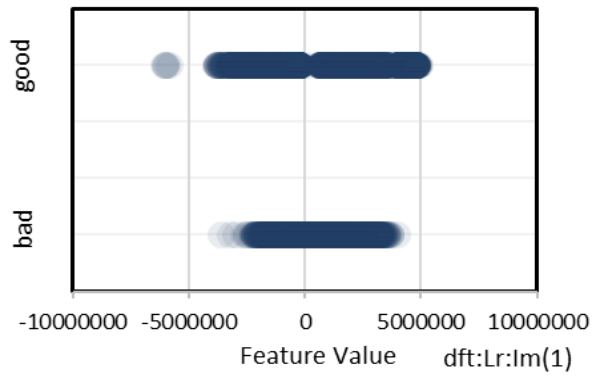


Figure 206: Descriptor (1), raw sucker rod load, imaginary part of the spectrum

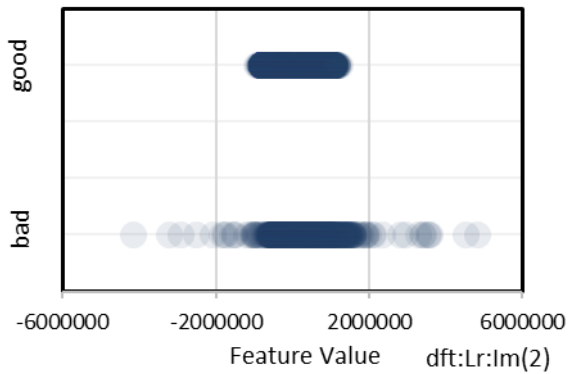


Figure 207: Descriptor (2), raw sucker rod load, imaginary part of the spectrum

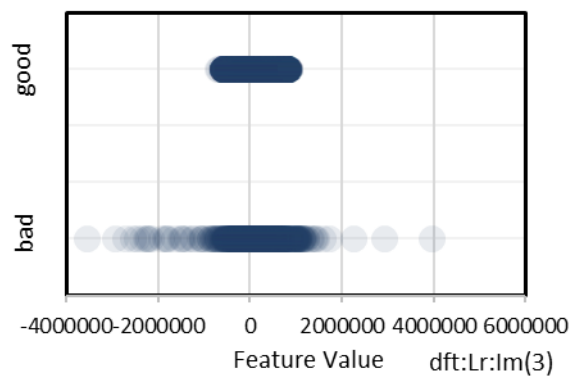


Figure 208: Descriptor (3), raw sucker rod load, imaginary part of the spectrum

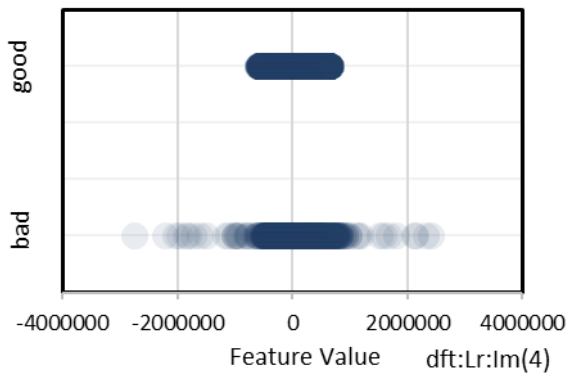


Figure 209: Descriptor (4), raw sucker rod load, imaginary part of the spectrum

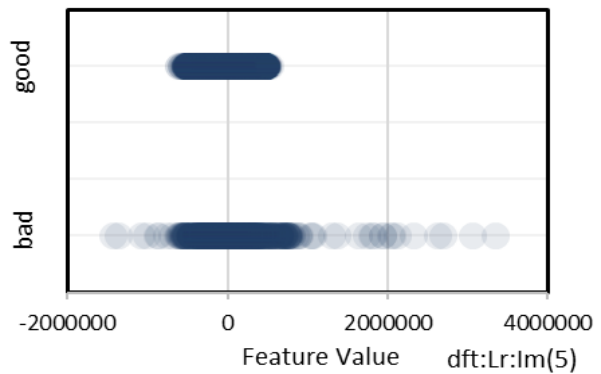


Figure 210: Descriptor (5), raw sucker rod load, imaginary part of the spectrum

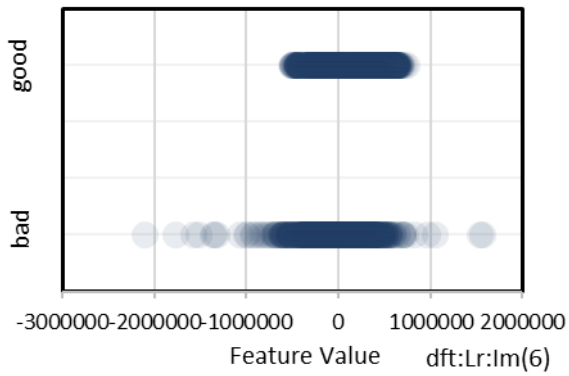


Figure 211: Descriptor (6), raw sucker rod load, imaginary part of the spectrum

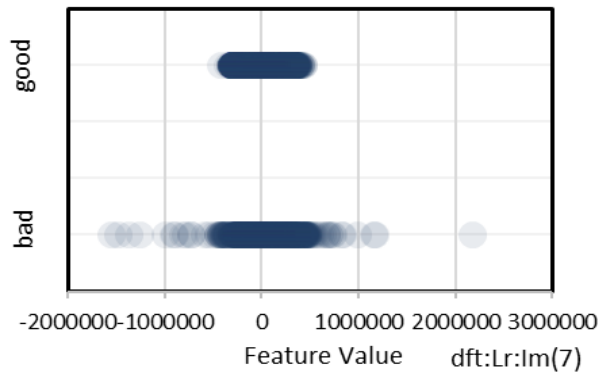


Figure 212: Descriptor (7), raw sucker rod load, imaginary part of the spectrum

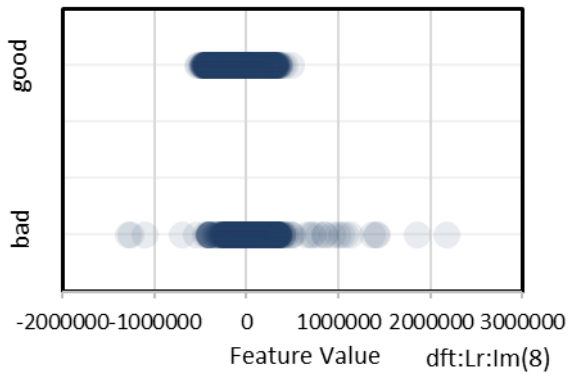


Figure 213: Descriptor (8), raw sucker rod load, imaginary part of the spectrum

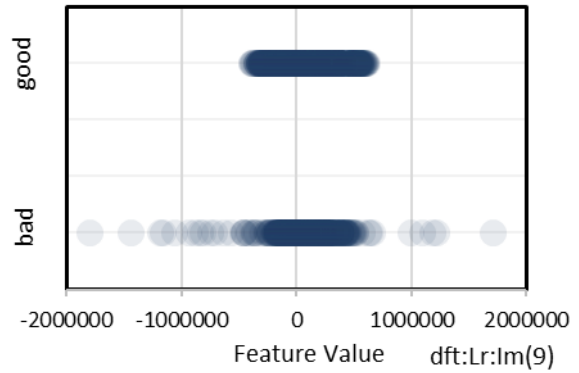


Figure 214: Descriptor (9), raw sucker rod load, imaginary part of the spectrum

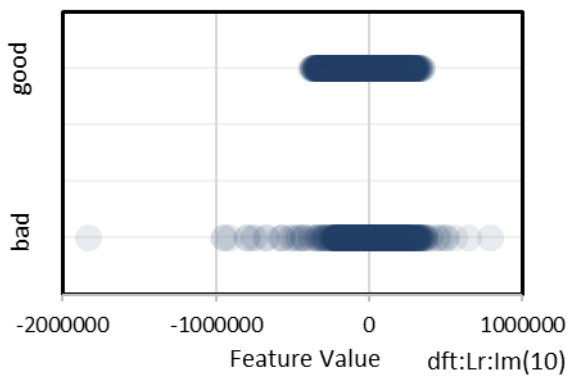


Figure 215: Descriptor (10), raw sucker rod load, imaginary part of the spectrum

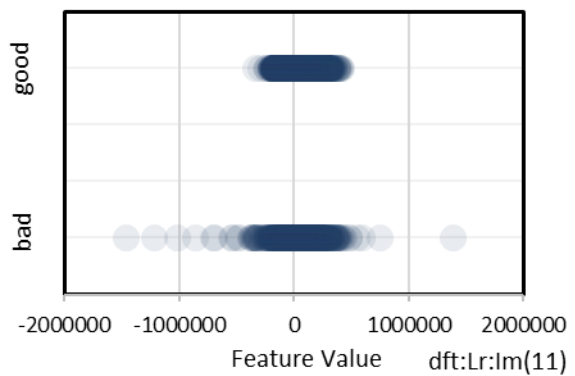


Figure 216: Descriptor (11), raw sucker rod load, imaginary part of the spectrum

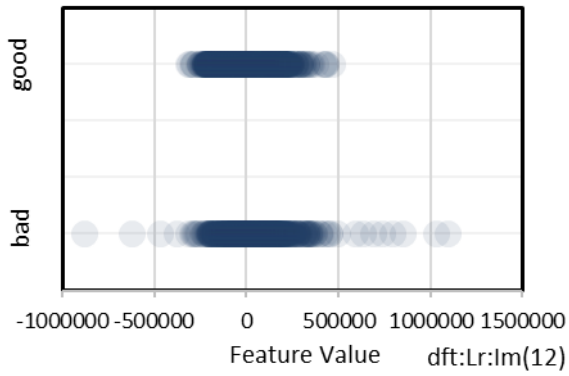


Figure 217: Descriptor (12), raw sucker rod load, imaginary part of the spectrum

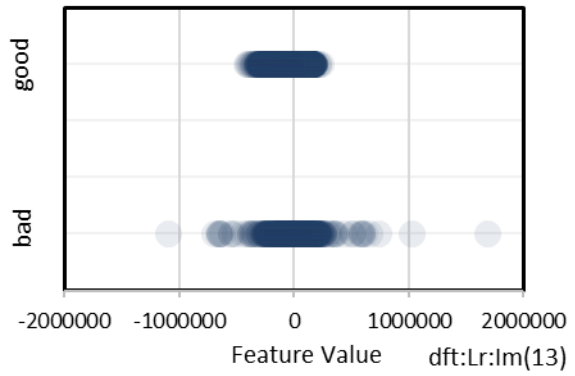


Figure 218: Descriptor (13), raw sucker rod load, imaginary part of the spectrum

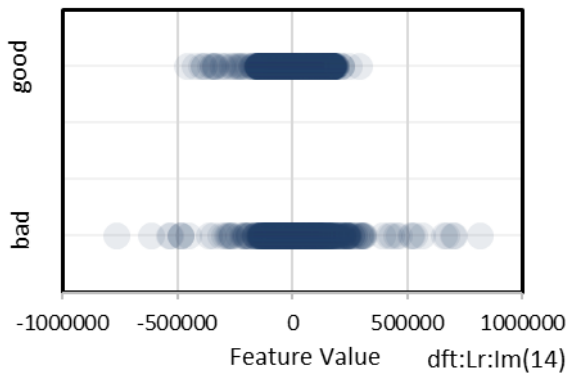


Figure 219: Descriptor (14), raw sucker rod load, imaginary part of the spectrum

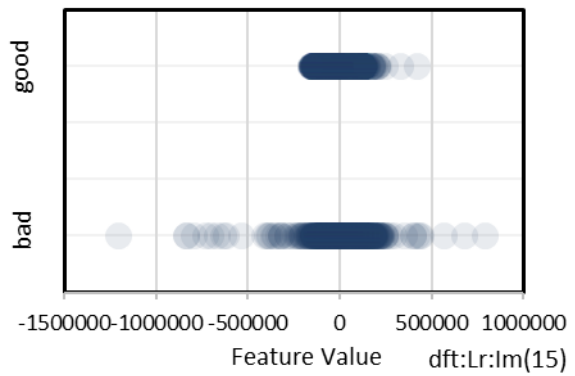


Figure 220: Descriptor (15), raw sucker rod load, imaginary part of the spectrum

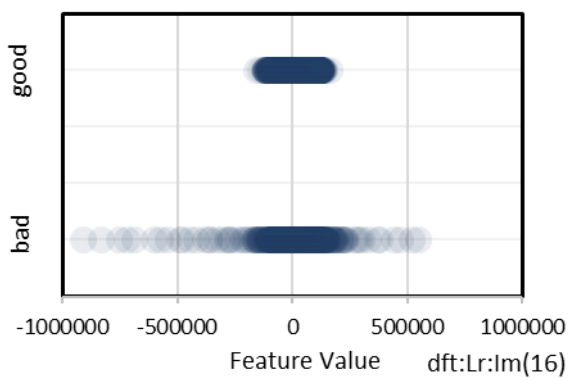


Figure 221: Descriptor (16), raw sucker rod load, imaginary part of the spectrum

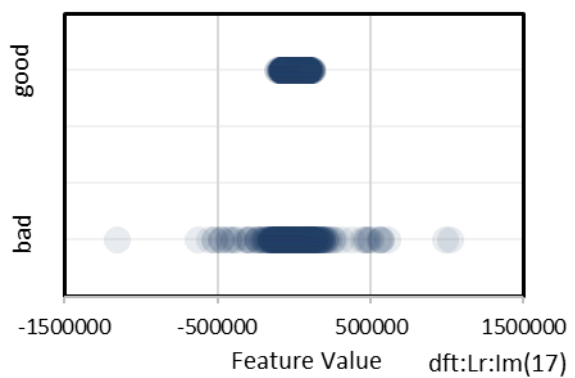


Figure 222: Descriptor (17), raw sucker rod load, imaginary part of the spectrum

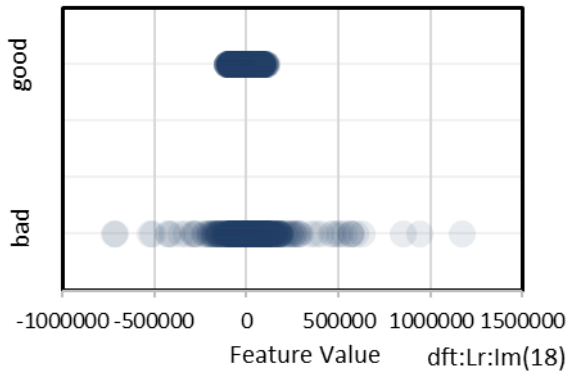


Figure 223: Descriptor (18), raw sucker rod load, imaginary part of the spectrum

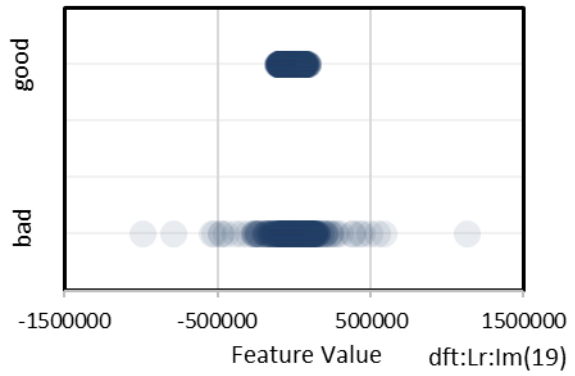


Figure 224: Descriptor (19), raw sucker rod load, imaginary part of the spectrum

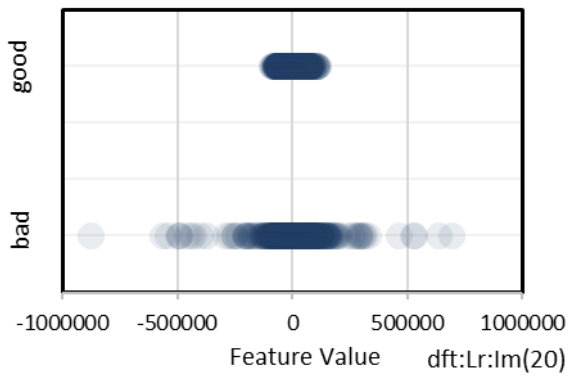


Figure 225: Descriptor (20), raw sucker rod load, imaginary part of the spectrum

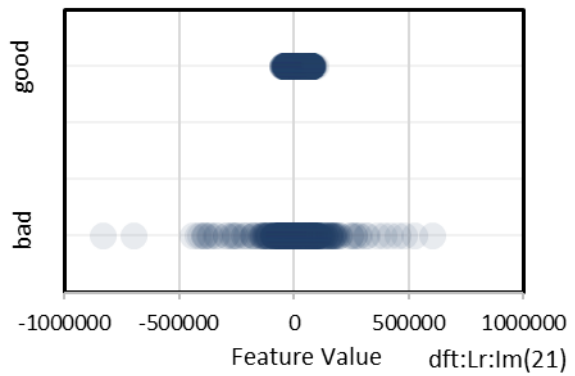


Figure 226: Descriptor (21), raw sucker rod load, imaginary part of the spectrum

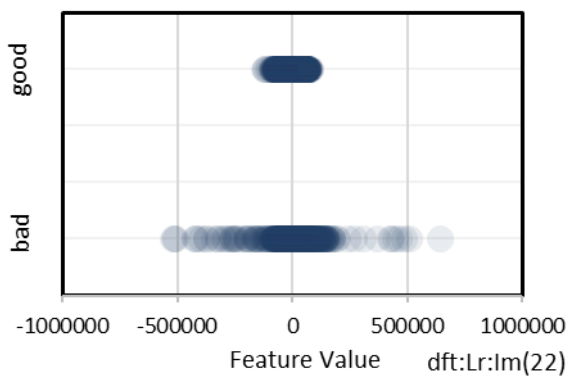


Figure 227: Descriptor (22), raw sucker rod load, imaginary part of the spectrum

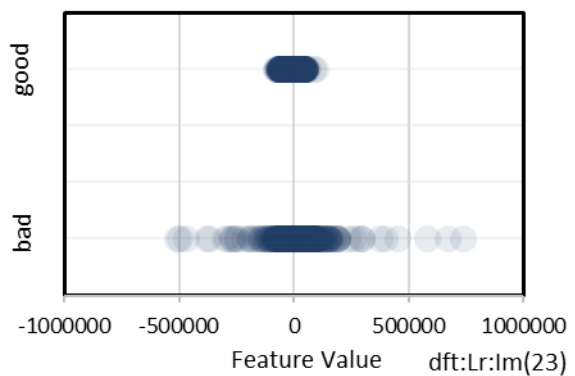


Figure 228: Descriptor (23), raw sucker rod load, imaginary part of the spectrum

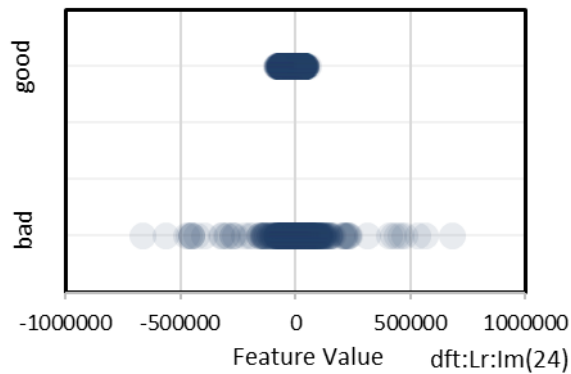


Figure 229: Descriptor (24), raw sucker rod load, imaginary part of the spectrum

Elliptic Fourier Transform features are very similar in quality and do not show a distinctive split between two labels apart from a few descriptors from the imaginary part of the spectrum., therefore, these will be omitted for the feature selection and will only be used as an input for the neural networks.

All above mentioned features, were used as inputs to the neural network with the same configuration as the previous approaches to classify the resampled data into different classes determined by the labelled data.

## 6.4 Feature Selection

In order to minimize the classification error, understand the performance of calculated features and identify most representative candidates, a feature selection process was initiated.

A few methods have been used such as sequential features selection method with two variants, sequential forward selection, and sequential backward selection; and search strategies with random search extension and exhaustive search extension.

**Sequential feature selection** has two components:

1. An objective function – criterion. In this method, we seek to reduce the criterion to its minimum over all subsets.
2. A sequential search algorithm – it removes or adds features from a predefined subset, meanwhile evaluating the criterion. A sequential search is only capable of moving in one direction by constantly expanding or shrinking the size of the candidate set.

### 6.4.1. Sequential Forward Selection (SFS)

SFS implies the addition of features to an empty candidate set until a point when adding more features do not increase the criterion. Initially, the best single feature is designated, then couples of features are created with the use of a single remaining feature and then, this leading

feature, and the best match is selected. The next step, triplets of features are formed using one of the remaining features, together with remaining two best features, and as a result, the best triplet is selected. This procedure takes place until all available features are selected. (Siddheshwar 2018)

From the mathematical point of view, when the input data in the algorithm is:

$$Y = \{y_1, y_2, \dots, y_d\}$$

Then the output will be:

$$X_k = \{x_j \mid j = 1, 2, \dots, k; x_j \in Y\}, \text{ where } k = (0, 1, 2, \dots, d)$$

Where the selected features are  $k$  and  $k < d$ .

In the initialization  $X$  is a null set and  $k = 0$  (where  $k$  is the size of the subset).

In the termination, the size is  $k = p$  where  $p$  is the number of desired features. (Verma 2021)

#### **6.4.2. Sequential Backward Selection (SBS)**

SBS is similar to the forward selection with the difference that in this case features are removed from the set until the highest criterion has been reached. So first, the criterion is calculated for all features and then, one by one, features are deleted until a predefined number of features are left.

Mathematically if the input data is:

$$Y = \{y_1, y_2, \dots, y_d\}$$

The output of the variant will be:

$$X_k = \{x_j \mid j = 1, 2, \dots, k; x_j \in Y\}, \text{ where } k = (0, 1, 2, \dots, d)$$

In the initialization  $X$  is a subset of features and  $k = d$  (where  $k$  is the size of the subset).

In the termination, the size is  $k = p$  where  $p$  is the number of desired features. (Verma 2021)

#### **6.4.3. Random Search Extension (RSX)**

Assuming we have a multi-dimensional grid, and we are looking for a point in this grid which minimizes or maximizes an objective function, a random point is taken, and the objective function value is measured. If the obtained value is better than the one already achieved, this point is kept in the memory. This procedure is repeated for a certain number of times. So, it

basically chooses features randomly, measures the model performance, and repeats it more times, the feature combination with the best performance is selected.

### 6.4.4. Exhaustive Search Extension (ESX)

ESX evaluates every possible combination of features and returns the best-performing subset.

### 6.4.5 Results Evaluation

#### Sequential Forward Selection

The analysis started with 30 features continuously adding one by one additional features.

#	Lent	Lacc	LF1	chans	chan	sd:pr:gsx	sd:lr:gsx	sd:pr:gex	sd:lr:gex	sf:pr:med	sf:pr:span	sf:pr:skew	sf:pr:kurt	sf:pr:ent	sf:lr:med	sf:lr:span	sf:lr:skew	sf:lr:kurt	sf:lr:ent	sf:vr:med	sf:vr:span	sf:vr:skew	
1	36,8%	95,2%	78,7%	30	30 Chn																		
2	36,8%	96,2%	84,0%	31	sd:vp:ip:err																		
3	36,8%	96,8%	86,7%	32	sd:lr:gsx	c																	
4	36,8%	97,4%	89,3%	33	sd:pr:gex	o	c																
5	36,8%	97,5%	89,5%	34	sd:lr:gex	o	o	c															
6	36,8%	97,7%	90,5%	35	sf:vp:ent	o	o	o															
7	36,8%	97,7%	90,8%	36	sf:vr:skew	o	o	o															c
8	36,8%	97,7%	90,7%	37	sf:vr:span	o	o	o														c	o
9	36,8%	97,8%	90,8%	38	sf:yr:ent	o	o	o														o	o
10	36,8%	97,7%	90,8%	39	fd:pp:rs:exp	o	o	o														o	o
11	36,8%	97,8%	91,0%	40	sd:lp:ip:r2	o	o	o														o	o
12	36,8%	97,8%	90,9%	41	sf:yp:ent	o	o	o														o	o
13	36,8%	97,8%	91,1%	42	sf:yr:span	o	o	o														o	o
14	36,8%	97,8%	91,1%	43	sd:lp:spmr	o	o	o														o	o
15	36,8%	97,8%	91,0%	44	sf:lr:skew	o	o	o									c					o	o
16	36,8%	97,8%	91,1%	45	si:fsp:f	o	o	o									o					o	o
17	36,8%	97,9%	91,3%	46	sd:vp:ip:r2	o	o	o									o					o	o
18	36,8%	97,9%	91,7%	47	sf:yr:skew	o	o	o									o					o	o
19	36,8%	98,0%	91,8%	48	sf:yp:skew	o	o	o									o					o	o

Figure 230: SFS results

Figure 230 illustrates a screenshot of the results derived from the SFS analysis. LF1 represents the F1-Score (criterion function) and the following column represents the number of input channels. Every channel is a feature.

By observing the LF1 column, we can see that F1 score value increases with the addition of each feature. The graphic illustration of overall results is presented in the chart in Figure 231.

First point on the graph represents the initial 30 channels, giving the highest error and lowest F1-score value. As features are sequentially added, the value of error decreases and F1-score increases.

Going back to Figure 230, columns to the right (marked 1-17...) contain letters “c” and “o” which indicate which features have been selected in order to get the F1-score value for the

particular number of channels. For example, in row number 15, we have a F1- score value of 91%, 44 features, out of which features under number 2, 3, 4, 12, 16, 17 have been kept as the best set.

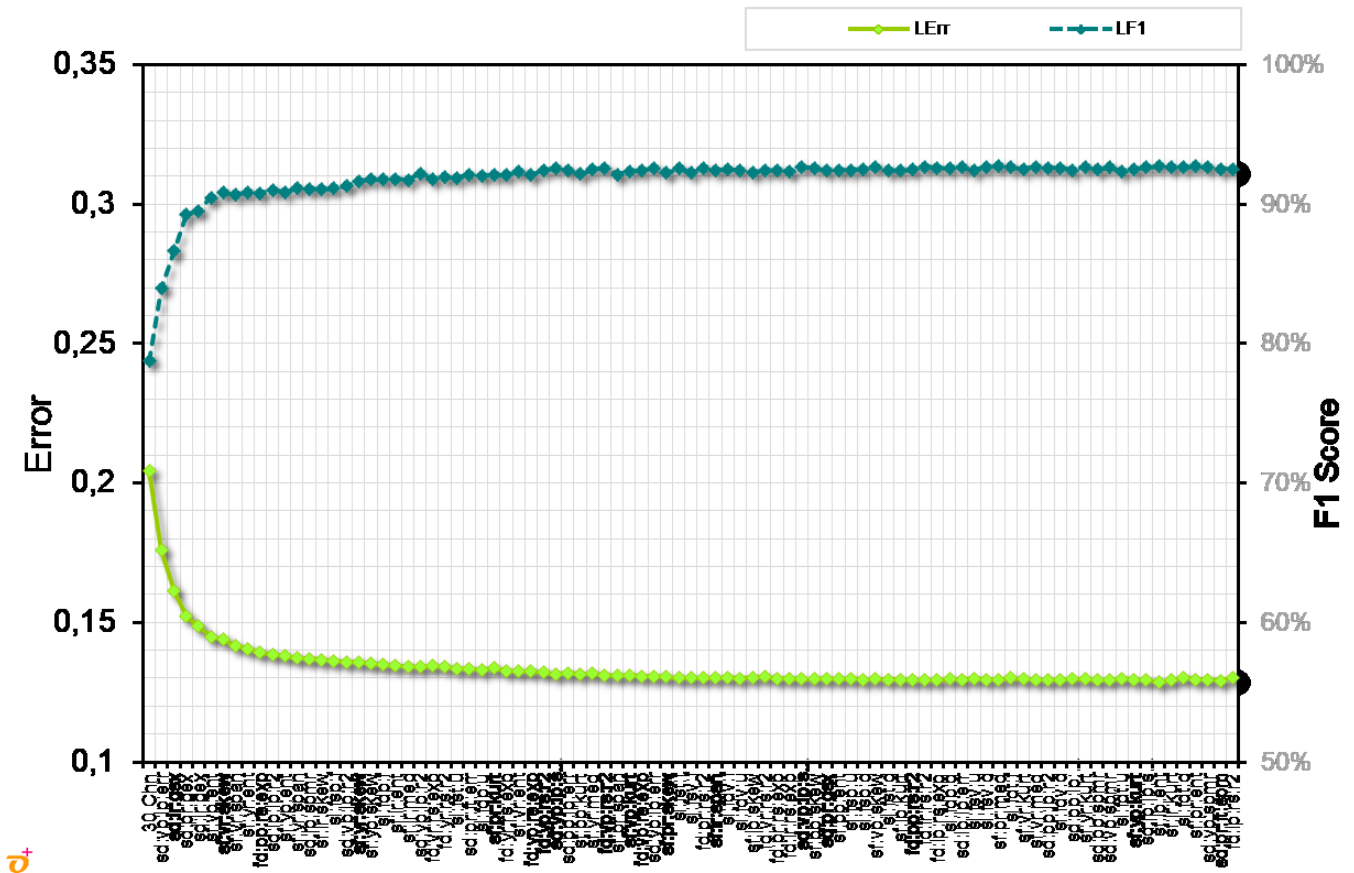


Figure 231: Visualization of F1- Score and learning error for the SFS analysis

### Sequential Backward Selection

Similar to SFS but reversed, SBS analysis started with 88 channels, sequentially removing feature by feature. Results are illustrated in Figure232.



#	LErr	Lent	Lacc	LF1	chans	chan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0,1315	36,8%	98,1%	92,2%	88	all																	
2	0,1307	36,8%	98,1%	92,6%	87	-fd.lr:rs:r2																	
3	0,1303	36,8%	98,1%	92,4%	86	-si:fsv:d																	
4	0,1305	36,8%	98,1%	92,4%	85	-sd:pp:ip:spm																	
5	0,1304	36,8%	98,1%	92,4%	84	-sf:pr:med					c												
6	0,1305	36,8%	98,1%	92,4%	83	-fd:vp:rs:r2					o												
7	0,131	36,8%	98,1%	92,5%	82	-si:fsv:u					o												
8	0,1304	36,8%	98,1%	92,5%	81	-sf:yr:med					o												
9	0,1306	36,8%	98,1%	92,5%	80	-sf:lp:ent					o												
10	0,1306	36,8%	98,1%	92,5%	79	-si:fsv:f					o												
11	0,1312	36,8%	98,1%	92,3%	78	-sd:yp:spmr					o												
12	0,131	36,8%	98,1%	92,3%	77	-sf:lr:skew					o							c					
13	0,1311	36,8%	98,1%	92,3%	76	-si:fdt:u					o								o				
14	0,1308	36,8%	98,1%	92,4%	75	-sf:pr:skew					o		c						o				
15	0,1306	36,8%	98,1%	92,3%	74	-sd:pp:ip:r2					o	o							o				
16	0,1312	36,8%	98,1%	92,2%	73	-sf:vr:kurt					o	o							o				
17	0,1306	36,8%	98,1%	92,3%	72	-fd:pp:rs:r2					o	o							o				
18	0,1307	36,8%	98,1%	92,6%	71	-sf:yp:ent					o	o							o				
19	0,1309	36,8%	98,1%	92,4%	70	-si:fdv:d					o	o							o				

Figure 232: SBS results

For example, in row 5, we have used 84 channels, which means 4 features have been deleted. The F1- score value has increased from 92,2% to 92,4%, keeping the 5<sup>th</sup> feature and leaving out the rest.

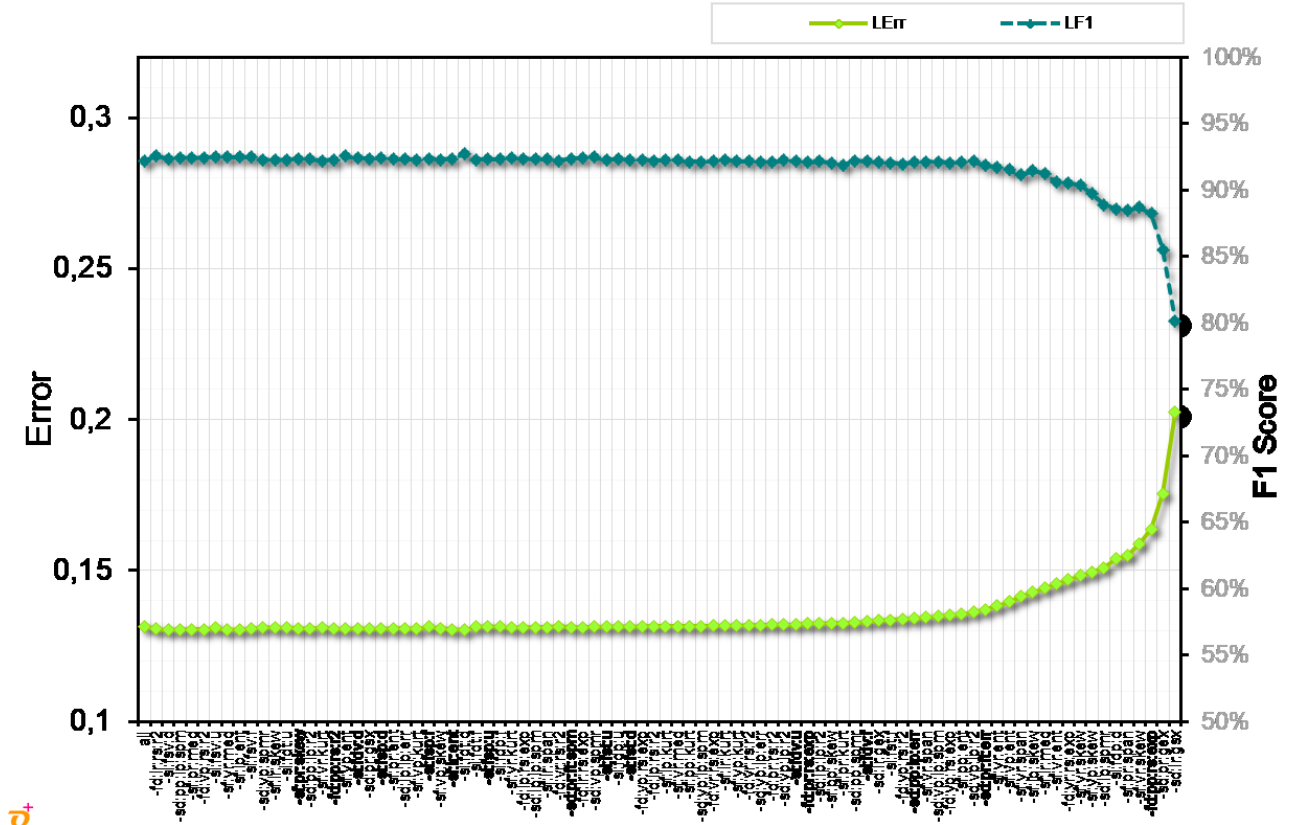


Figure 233: Visualization of F1-Score and learning error for the SBS analysis

The graph in Figure 233 should be read from right to left. The first points to the right include all features and we can see that reducing feature by feature, the value of both error and F1-Score are increasing.

**SFS vs SBS**

Comparing the results of these two analyses it can be identified that the learning error varies in both cases between 0,13 and 0,2 and F1-score between 80% and 93%.

Taking a closer look (Figure 234 A and B) it can be seen that the peak performance for SBS is reached at 92,7% corresponding to 61 channels.

In the case of SFS (Figure 235 A and B) peak performance is reached at the same value of 92,7% corresponding to 89 channels. Considering the fact that the analysis here started at 30 channels, therefore subtracting 30 out of 89, we reach a total number of 59 channels.

In conclusion it can be said that the highest value of the criterion function (F1-score) has been reached in both cases after the same average number of iterations, that is 60. This means that a subset of 28 features gives the best score.

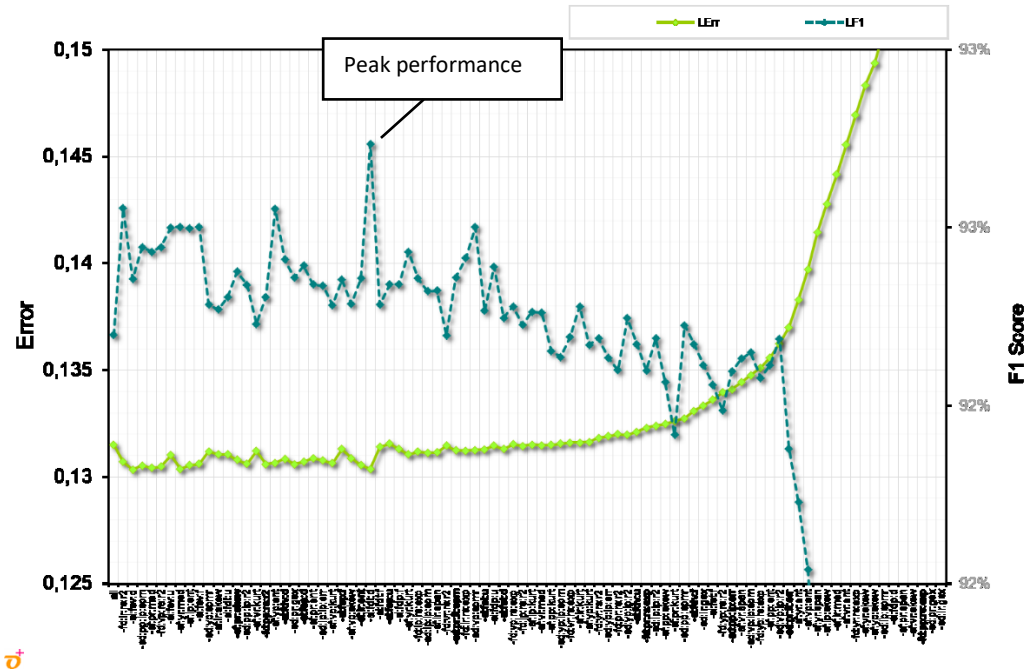


Figure 234 A

LErr	Lent	Lacc	LF1	chans	chan
0,13122	36,8%	98,1%	92,2%	73	-sf.vr:kurt
0,1306	36,8%	98,1%	92,3%	72	-fd.pp.rs:r2
0,13066	36,8%	98,1%	92,6%	71	-sf.yp.ent
0,13086	36,8%	98,1%	92,4%	70	-si.fdv:d
0,1306	36,8%	98,1%	92,4%	69	-sd.pr.gsx
0,13071	36,8%	98,1%	92,4%	68	-si.fsp:d
0,13088	36,8%	98,1%	92,3%	67	-sf.pr.ent
0,13077	36,8%	98,1%	92,3%	66	-sd.lp.ip:err
0,13064	36,8%	98,1%	92,3%	65	-sf.vp:kurt
0,13131	36,8%	98,1%	92,4%	64	-si.fsp:f
0,13088	36,8%	98,1%	92,3%	63	-sf.vp.skew
0,13055	36,8%	98,1%	92,4%	62	-sf.lr.ent
<b>0,13038</b>	<b>36,8%</b>	<b>98,2%</b>	<b>92,7%</b>	<b>61</b>	<b>-si.fdt:d</b>
0,13141	36,8%	98,1%	92,3%	60	-si.fdt:f
0,13156	36,8%	98,1%	92,3%	59	-si.fsp:u
0,1313	36,8%	98,1%	92,3%	58	-si.fdp:f

Figure 234 B

Figure 234: Performance evaluation of the SBS analysis

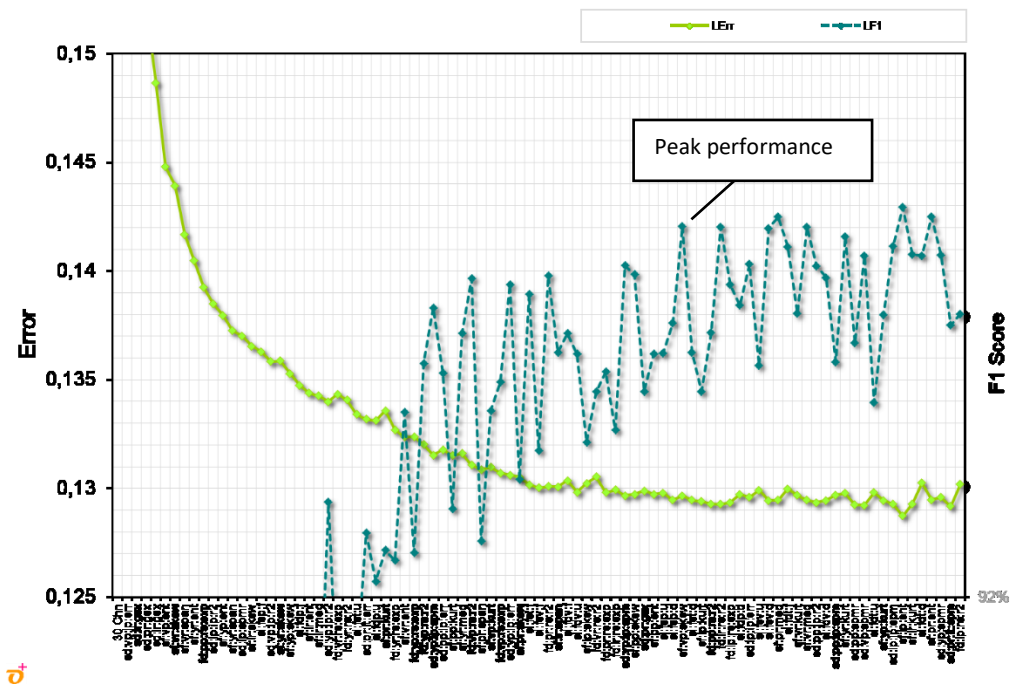


Figure 235 A

LErr	Lent	Lacc	LF1	chans	chan
0,1301	36,8%	98,2%	92,6%	75	fd.pr.rs:r2
0,13008	36,8%	98,1%	92,5%	76	sf.lr.span
0,13037	36,8%	98,1%	92,5%	77	si.fdv:f
0,12983	36,8%	98,1%	92,4%	78	si.fdv:u
0,13024	36,8%	98,1%	92,3%	79	sf.lp.skew
0,13053	36,8%	98,1%	92,4%	80	fd.vr.rs:r2
0,12981	36,8%	98,1%	92,4%	81	fd.pr.rs:exp
0,12993	36,8%	98,1%	92,3%	82	fd.lr.rs:exp
0,12968	36,8%	98,2%	92,6%	83	sd.vp.ip:spm
0,12974	36,8%	98,2%	92,6%	84	sf.pp.skew
0,12988	36,8%	98,1%	92,4%	85	sd.pr.gsx
0,12974	36,8%	98,1%	92,4%	86	sf.pp.ent
0,1298	36,8%	98,1%	92,4%	87	si.fsp:u
0,12947	36,8%	98,1%	92,5%	88	si.fsp:d
<b>0,12966</b>	<b>36,8%</b>	<b>98,2%</b>	<b>92,7%</b>	<b>89</b>	<b>sf.vp.skew</b>
0,12946	36,8%	98,1%	92,5%	90	si.fst:d

Figure 235 B

Figure 235: Performance evaluation of the SFS analysis

### Random Search Extension

For the RSX, the objective function will also be considered F1 Score. In total, 818 random iterations took place for a single channel and for two channels. The best scoring features have been selected as illustrated in the Figure 236.

LErr	VErr	TErr	Vent	Tent	LF1	chans	tag	sd:pr:gsx	sd:lr:gsx	sd:pr:gex	sd:pp:ip:err	sd:lp:ip:err	sd:vp:ip:err
0,2074	0,2074	0,2074	36,5%	36,2%	79,5%	1	1 chn	c		3	47	48	49
0,1541	0,1542	0,1544	36,2%	37,1%	88,4%	2	2 chn		c		c		

Figure 236 RSX analysis output

For the single channel, highest F1 score has a value of 79,5% and for 2 channels 88,4%.

*Lerr*, *Verr*, *Terr* represent model errors for learning, validation, and testing subsets. *Vent* and *Tent* are entropies for validation and testing sets.

Analyzing a graphical representation of these results (Figure 237) it is evident that the overall performance of two features is higher than a single channel.

Winning features in this case are all derived from stroke detector. For a single channel, the winning feature is the raw sucker rod position with normalized absolute differences in the whole stroke. For two channels, the winners are the raw sucker rod load with normalized absolute differences in whole stroke and processed sucker rod position with instantaneous phase and mean square error of linear fit.

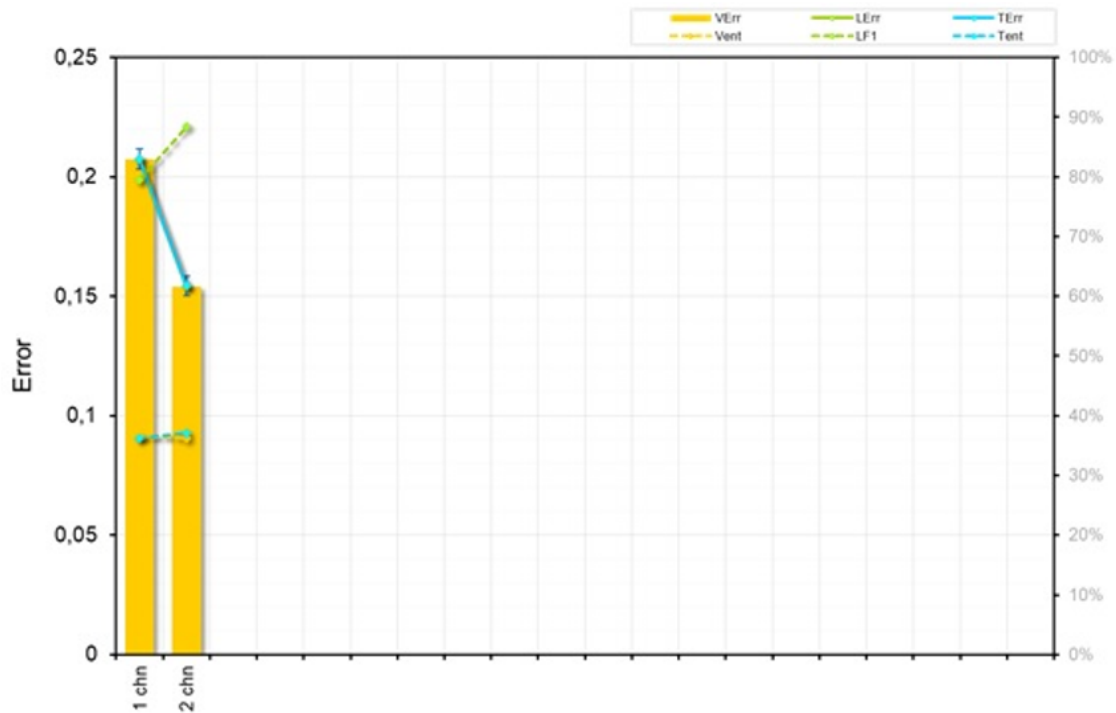


Figure 237: Results visualization for the RSX analysis

### Exhaustive Search Extension

In this case all features have been paired between them and as a result 3916 iterations took place. Similar to RSX, best candidate for the single channel and best candidates for the pair have been selected, as illustrated in Figure 238.

							sd:pr:gsx	sd:lr:gsx	sd:pr:gex	sd:lr:gex	sd:vp:ip:err	sd:yp:ip:err	fd:lp:rs:r2	fd:vp:rs:exp
LErr	Lent	Lacc	LF1	chans	tag		1	2	3	4	49	50	66	67
0,2024	36,8%	95,5%	80,2%	1	1 chn						c			
0,1751	36,8%	96,8%	86,5%	2	2 chn		c							c

Figure 238: ESX analysis output

*Lent* is the entropy for the learning subset and *Lacc* is the classification accuracy for the learning subset. Highest F1 score for a single channel is 80,2% and for two channels it is 86,5%.

The winning feature for the single channel is also derived from the stroke detector, that is processed sucker rod velocity obtained using the instantaneous phase and root mean square error of linear fit. For two channels the winners are the raw sucker rod load with normalized absolute differences in the whole stroke, obtained from the stroke detector together with the

processed sucker rod velocity with rescaled range statistical analysis and Hurst exponent calculated using fractal dimensions.

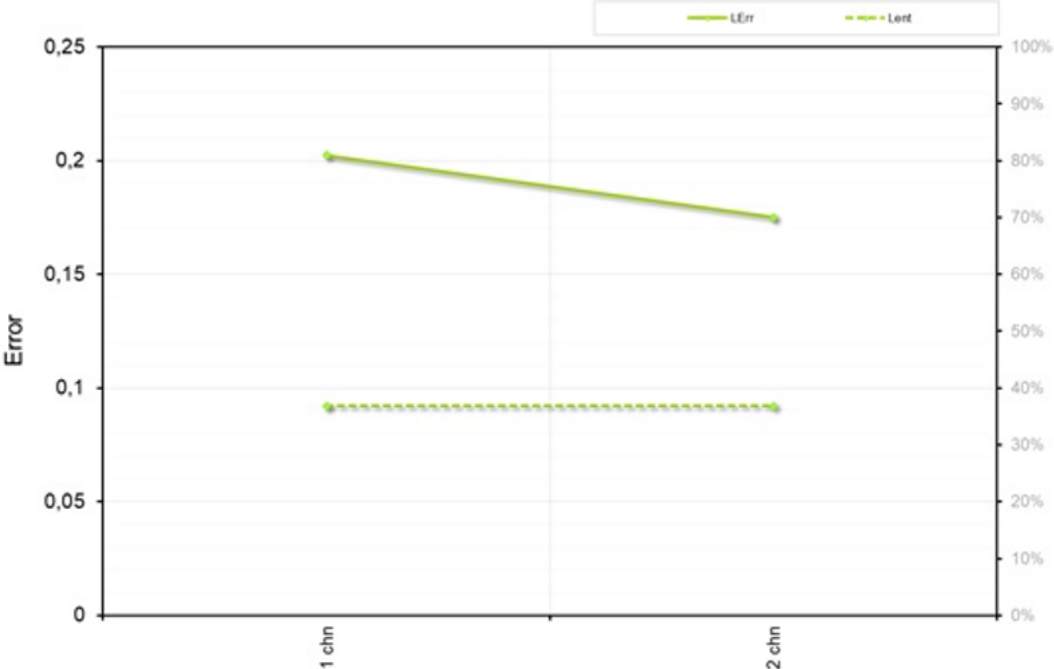


Figure 239: Results visualization for the ESX analysis

Figure 239 illustrates the fact that the error is lower for two channels instead of a single channel.

**RSX vs ESX**

Comparing the outputs of these two analyses it can be concluded that a pair of two features gives better results as compared to a single feature. The F1 score value is higher in the ESX analysis as all features have been considered. Nevertheless, RSX can also provide an optimal result in case we want a good, reasonably high value that it's not the highest possible.

## 7 Heuristic QC model

In this chapter the heuristic algorithm based on artificial neural networks for determining data quality will be described.

### 7.1 ANN Training

Due to a very large number of dynamograms available it was necessary to come up with an automatized method for identifying erroneous measurements. The most efficient and secure way to do it was by training an algorithm based on artificial neural networks which is able to classify the data in two categories, “good” and “bad”.

The features described in chapter 6 calculated for a subset of manually labelled cards have been used as input. The subset contains examples with “good” and “bad” labels and was set as the desired output for the neural network.

The data to be used for training and testing were organized in an Excel spreadsheet. The next step was to split the input data into three parts. 60% were allocated for learning, 20% for validation, and the remaining 20% for testing.

A repeated random sub-setting validation, also known as Monte Carlo cross-validation was performed for the error evaluation. The data gets divided into several multiple splits of training and validation data. For each split the model is fit for the training data and the accuracy is evaluated using validation data. The results are averaged over the splits. The main advantage of Monte Carlo cross-validation is that the amount of splits is independent from the number of iterations. (Wikipedia)

Multilayer perceptron (MLP), illustrated in Figure 240, was selected as the type of artificial neural network to be used, the number of experts in the cluster is 10. Three different configurations have been trained and tested. As general parameters for all three models, standard backpropagation was selected as training method, and local adaptive learning with weight and error backtracking iRProp+ (Improved Resilient Propagation Plus) as a learning rule. The number of hidden layers varies from 0 to 10 in two models and from 0 to 20 in one model. After manually labelling some part of the data and classifying it in two categories as “good” and “bad, the models were trained.

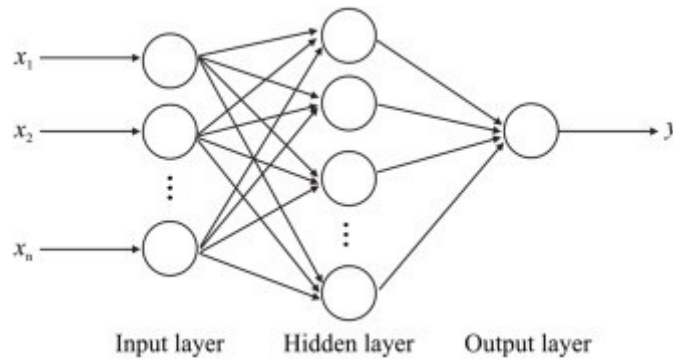


Figure 240: MLP architecture (Fath et al. 2020)

### 7.1.1 Training Results

After training all three models, the one with the best performance based on the smallest validation error and highest F1-score was selected. The network configuration is (MLP+0c i244b-l3s-o1bL2/R+)AMx10.8e5.4(k5.4), this means that the MLP network has 244 inputs with bias, 3 hidden units using sigmoid activation function without bias, 1 linear output unit with bias, 8<sup>th</sup> expert out of 10 was selected and trained with the 4<sup>th</sup> subset out of 5 splits.

Table 1 illustrates the confusion matrix for the validation set of the best performing network and Table 2 shows the additional performance evaluation of the model based on entropy, accuracy, precision, recall and F1-score.

In Table 1, blue cells represent the number of correctly identified classes as a result of the trained neural network. This means that 292 dynamometer cards with label “bad” have been identified correctly. The same way, 4423 cards have been correctly classified as “good”. Meanwhile, 24 cards from class “bad” have been identified incorrectly as “good”. Similarly, 9 cards from class “good” have been identified as “bad”.

The total number of dynamometer cards used for the validation set is 4748, out of it, 316 have been identified as bad and 4432 as good. As a result, only 33 measurements out of 4748 have been incorrectly classified. Resulted accuracies for each class are 92.4% for “bad”, 99.8% for class “good”; 99.5% is the precision for class “bad” and 97.0% is the precision for class “good”. The confusion matrices for learning, testing, and validation for all three models that have been trained can be found in Appendix B.



Table 1: Confusion matrix for the validation set of the best performing network

Class	bad	good	
	92.4%	99.8%	
<b>bad</b>	<b>292</b>	9	97.0%
<b>good</b>	24	<b>4423</b>	99.5%
	24	9	33
	316	4432	4748

In order to further evaluate the performance of the model, additional parameters have been calculated and results are shown in Table 2.

Accuracy is the percentage of correct predictions, precision is the ratio of correctly identified observations to the total number of observations, recall is the ratio of correctly identified observations to the all observations in the actual class, and F1-score is the harmonic mean of precision and recall.

Table 2: Model performance evaluation

Sub-set	Accuracy	Precision	Recall	F1-score
<b>Validation</b>	99.3%	96.1%	98.1%	97.2%

### 7.1.2 Challenges Encountered in ANN Training

The training dataset is used for fitting the neural network in order to update the model weights and create a good mapping of inputs and outputs. An optimization algorithm is used for the training process to look through all possible values for ANN's weights that will produce a high performance of the training dataset. (Brownlee 2019)

Training problem is hard because the algorithm depends on labelled training data. Sometimes the amount of labelled data is not enough and as a result, it becomes difficult to fit the parameters in a complex model without overfitting, in a supervised training.

Training neural networks requires solving a non-convex optimization problem, therefore gradient – based training is able to find the local minima only. Having a deeper architecture, achieving a good generalization becomes more difficult with increasing number of local minimums.

Diffusion of gradients can also result into a deficient tuning of the lower layers and as a result lead to unsatisfactory outcome. In other words, the weights in the lower layers change more slowly due to the fact that increasing network depth leads to a rapid decrease of gradients magnitude. (Audhkhasi, Sethy, Rambhardan and Shrinkanth 2013)

During the training process, the derivatives of slopes can become very small or very big, as a result it leads to vanishing or exploding gradients.

As a rule, stochastic gradient is used to tackle challenges related to training neural networks. Nevertheless, it can give no guarantees and lead to following situations:

- Questionable solution quality as the optimization process can or cannot find an appropriate solution as solutions can only be compared relatively.
- Long training time due to the iterative nature of search.
- Possible failure as due to regions with zero gradient, the optimization process may get stuck or isn't able to identify a viable solution. (Brownlee 2019)

Main challenges encountered in the training process during current work, were mainly related to the data quality. In order to have a proper subset to be used for training, a part of dynamometer cards had to be labelled by hand. As a result, over 12,300 cards have been manually labelled as "good" and "bad".

#### **7.1.2.1 Manual Labelling of Dynamometer Measurements**

It was mentioned several times that in supervised learning a labelled subset of training data is necessary for the neural network input. As in the current work, the amount of available data is very large, respectively the manually labelled subset was required to have a significant amount of samples. Therefore, 12,300 digital surface dynamometer cards were labelled manually, and further used for quality checking the entire dataset.

During the labelling process a few challenges have been encountered as:

Occasional inability to distinguish between bad measurements due to sensor error or pump function. Figure 241: An example of bad measurement is illustrated in Figure below.

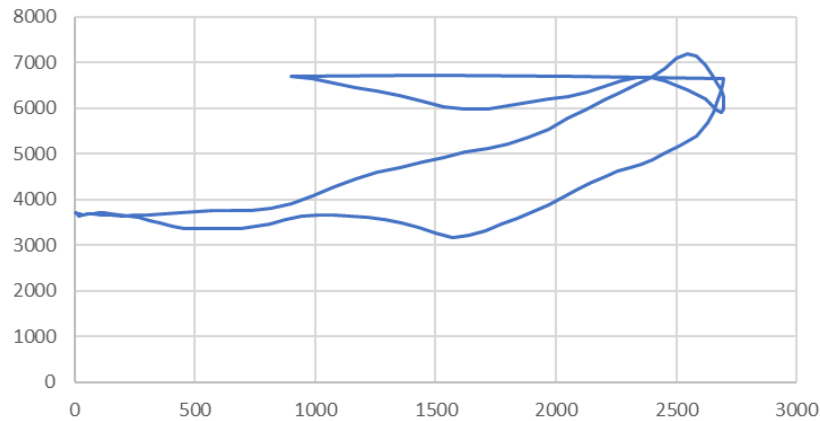


Figure 241: An example of bad measurement

In such cases, the situation was discussed with OMV experts and based on their experience and knowledge about particular wells, the distinguishment was made. The importance behind making this difference is that if the measurement was bad due to a sensor error, then it was labelled as “bad” and later removed from the dataset. Whereas, if the measurement was bad due to a pump malfunction, then the issue was addressed to production engineers and the well was checked.

- Missing data gaps due to periods when measurements were stopped. During the course of historical data provided, there were some periods of time for which no data is available. This phenomenon is explained by the fact that sensors got changed over time, therefore no recordings have been made for that duration. The challenge in this situation is that for some of the wells the data quality is different, compared to previous time periods and due to missing data, it is unknown what happened in between. This is not affecting the quality check as much as pump states identification, problem that will be discussed more into details in the next chapter.
- Long training time due to a high volume of data and multiple number of iterations was another challenge. Multiple network parameters have been modified over the course of training in order to reduce the computational time. Covariance based weight adaptive initialisation was selected as the most efficient weight initialisation method.

After feeding the data in the best performing model and checking the results, it was found that 88% of the total dataset was good data and the remaining 12% was removed. This is a very good indicator which means that most of received measurements were good and could be further used for trend identification. It is also good news for OMV as it means that most of the data is of a high quality and can be analysed and used for production optimization.

# 8 Heuristic Pump State Model

In this chapter, the method developed to detect sucker rod pump states is explained.

## 8.1 Sucker Rod Pump Diagnosis

SRP is the most widely used artificial lift system, being present in more than 750,000 wells around the world. Maintenance and optimization activities for this type of system can be rather expensive and time consuming, fact which leads to a constant necessity of installing a monitoring system, to assure a proper operation of the pump.

A diagnosis system used to detect any pump malfunctions and maintain the pumping system intact is achieved by the means of dynamometer measurements. Resulted dynamometer cards are used to analyse the downhole working conditions of the pump and identify its efficiency. By analysing a dynamometer card, it can be decided upon modifying some operating conditions such as pump speed, stroke length, pump size, in order to increase production and efficiency.

Dynamometer cards are divided into two categories, surface and downhole. Surface dynamometer cards records all forces acting on the pump and rod string during a full pump cycle. These forces are the dynamic and static rod forces, dynamic and static fluid forces, dumping effects that appear as a result of friction between the pump, fluid column, rods, and tubing. The combination of all these forces as a function of the polished rod position represents the surface dynamometer card.

Downhole dynamometer cards can be calculated from surface cards, using the wave equation and recreate rod string elastic nature assuming a downhole friction factor. A downhole card illustrates the fluid load of the pump plunger over a pump cycle. See an example in Figure 242.

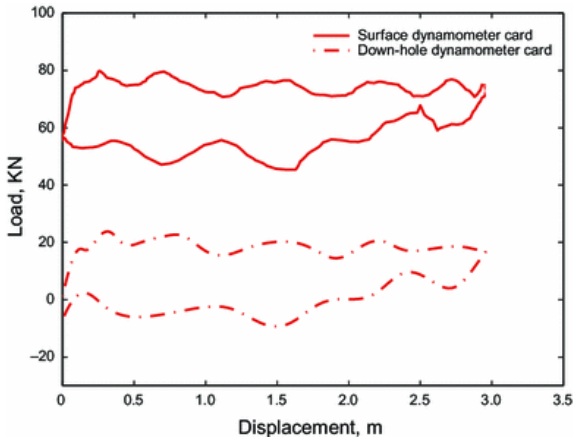


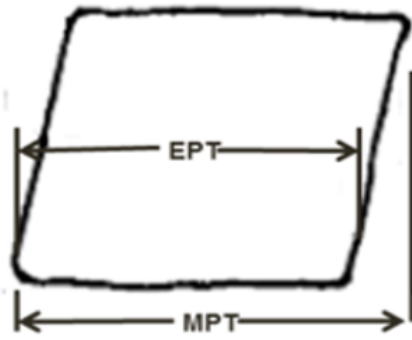
Figure 242: Surface and downhole dynamometer card example (Li et al. 2015)

## 8.2 Sucker Rod Pump States

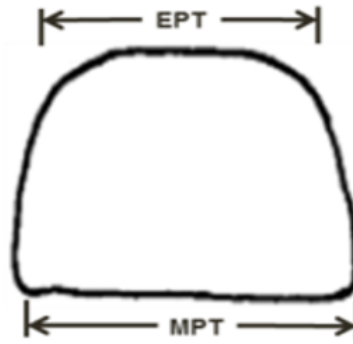
Depending on the operating conditions, several pump states of SRP can be identified in dynamometer cards.

Figure 243 illustrates surface dynamometer cards with various pumping conditions for unanchored tubing. Figure A shows the normal pumping conditions without gas. In this case, the pump is working properly. Figure B depicts the leaking travelling valve situation and Figure C the leaking standing valve case. Figure D depicts one of the most common conditions, fluid pound. It takes place when tubing is hit due to rod buckling, the resulted high loads are damaging the travelling valve ball and seat, and as a result the pump valve rod undergoes buckling and creates plunger side – loading into the barrel. These stresses significantly decrease rod life, and the impact is transmitted all the way up to the gearbox. Figure E portrays the gas interference situation. Gas Interference is similar to fluid pound and although the impact is less, the inefficiency level stays the same. During this situation, extra strokes are required which will produce the same damage to the equipment. Entering gas travels up to the tubing string and may cause leaking stuffing box and brings in an erratic pump performance. Figure F shows a condition named gas lock. In this case the pump operates at very low volumetric efficiency and due to gas compression and expansion, no liquid is being pumped. As a result, there is no valve action so that the card has a very small area.

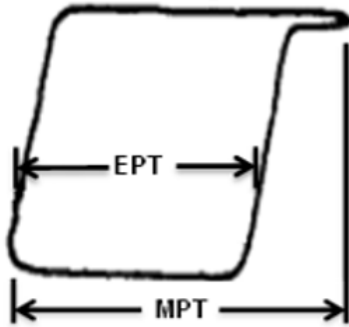
Additionally, to afore described pump states, many other pump malfunctions can be derived based on the interpretation of dynamometer measurements.



A: Normal



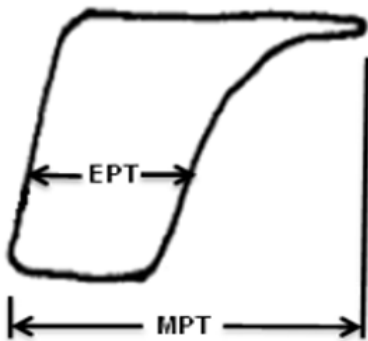
B: Leaking travelling valve



C: Leaking standing valve



D: Fluid pound



E: Gas interference



F: Gas lock

Figure 243: Dynamometer cards pump states

(EPT = Effective Plunger Travel, MPT = Maximum Plunger Travel)

### **8.3 Sucker Rod Pump States Identification – ANN Training**

In order to be able to perform a trend analysis, it is important to first identify pump behaviours in existing data.

Data was split the same way and the networks were of the same type (multilayer perceptron) and have the same parameters and architecture. The difference was the input data.

The subset used for input consisted of 17,330 manually labelled dynamometer cards from 52 wells. 10 major pump states have been identified and used as labels. These are:

1. Normal
2. Fluid pound
3. Gas interference
4. Fluid pound + tagging
5. Gas interference + tagging
6. Leaking travelling valve
7. Overtravel
8. Pumped off
9. Tagging
10. Unset anchor

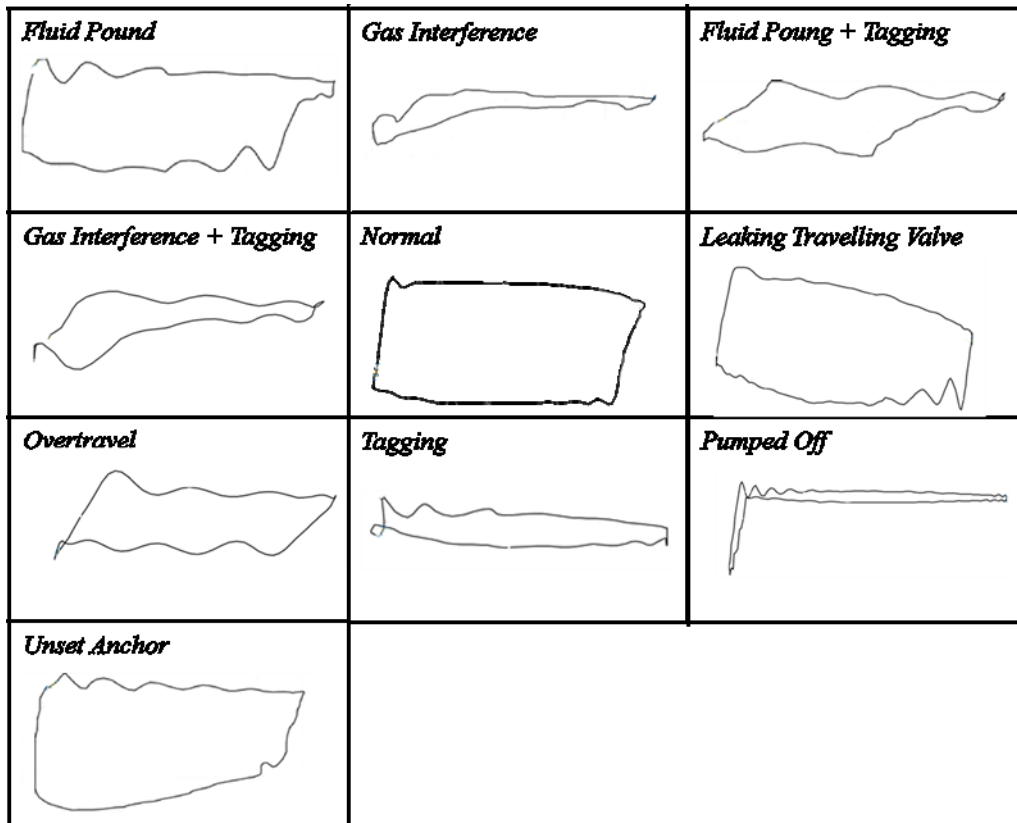


Figure 244: Identified pump states

Figure 244 illustrates examples for each pump state identified in the data during visualization and labelling processes.

The approach was to train 4 different models and select the best performing network based on lowest validation error and highest F1-score percentage. F1- score was selected as the main model metric because it combines both precision and recall and works better for imbalanced class distribution as compared to accuracy which works better for a symmetric data set. Considering this criteria, network with the following parameters had the best performance: (iMLP+0c i163b-l8s-o10b L2/R+)AMx10.10e5.4(k5.4), this means that the MLP network has 163 inputs, 8 hidden units and 10 outputs, 10<sup>th</sup> expert out of 10 was selected and trained with the 4<sup>th</sup> subset out of 5 splits.

Table 3 illustrates the confusion matrix for the validation set of the best performing network and Table 4 shows the additional performance evaluation of the model based on parameters like entropy, accuracy, precision, recall and F1-score.



Table 3: Confusion matrix for the validation set of the best performing network

Class	fpond	fpond+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset anchor	
	97.3%	0%	97.4%	60%	100%	99.8%	25%	0%	100%	100%	
fpond	401	3	2			3		4			97.1%
fpond+tag		0									0%
gasif	5		304	8		2	4				94.1%
gasif+tag			1	12			2				80%
ltv					97						100%
normal	6		5			2056					99.5%
overtravel							2				100%
poff								0			0%
tagging									48		100%
unset anchor										84	100%
	11	3	8	8	0	5	6	4	0	0	45
	412	3	312	20	97	2061	8	4	48	84	3049

Table 4: Model performance evaluation

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	98.5%	68%	77.1%	72.2%

Confusion matrices for learning, testing, and validation for all trained models are found in Appendix C.

In Table 3, blue cells represent the number of correctly identified classes as a result of the trained neural network. This means that 401 dynamometer cards with label “fpond” have been identified correctly. The same way, 2056 cards have been correctly classified as “normal”. Meanwhile, 11 cards from class “fpond” have been identified incorrectly, 5 as “gasif” and 6 as

“normal”. Similarly, 3 cards from class “normal” have been identified as “fpound” and 2 as “gasif”, and so on for the rest of the classes.

The total number of dynamometer cards used for the validation set is 3049, out of it, 45 have been identified with a wrong class. Resulted accuracies for each class are shown in percent in the second row under classes names and precisions in the last column of the table.

After the classification was performed on labelled data, using supervised learning, the rest of the data (unseen) was fed into the selected network to be classified in those 10 classes mentioned earlier and check how accurate are the results provided by the network.

The network identifies a class for every input based on probabilities. In order to make sure that these classes were assigned correctly, some measurements classified by the network have been visualized in Excel for assessing the accuracy of the classification. As a result, it was concluded that all measurements have been correctly classified, which means that there is a good match between input and output.

It is worth mentioning that most measurements have been classified as „normal”, indicating that the majority of the wells are operating at good conditions and assure maximum production and efficiency. Nevertheless, the next two most common pump states are fluid pound and gas interference. These are most damaging malfunctions and as a result, it is important to have a model which offers the possibility to identify and predict trends in pumps behaviour, in order to avoid future failures and improve production and pumps efficiency.

## **8.4 Challenges Encountered in ANN Training**

Training time and manually labelling the dataset to be used for training were the main challenges encountered.

It was expected to have a longer training time as the amount of data was very large, in order to achieve a shorter computing time, covariance base adaptive weight initialisation has been applied.

### **8.4.1 Distinguishing Between Similar Dynamometer Cards Shapes**

Most common pump states were normal, therefore easy to identify. However, distinguishing between fluid pound and gas interference can be challenging sometimes as shapes look similar and only minor differences can make a separation between the two. Whenever this difficulty was encountered during the manually labelling process, other parameters such as gas-liquid ratio and historical data related to previous pump malfunctions for the particular well were checked. By accessing additional data about well’s production and pump behaviour, it was clear whether the condition was fluid pound or gas interference. As for the network it would

have been too complex to include additional inputs, some well distinguishable examples have been selected as input for training, accompanied by features described in Chapter 6. The final check between networks output and real measurements showed that the model was capable of differentiating between fluid pound and gas interference with 98% precision, as well as other pump states.

## 9 Trend Analysis

The main goal in this project is the identification of trends present in SRP states behaviour and forecasting future pump states based on the identified trend. In this chapter, the methodology of trend identification and forecasting will be described, altogether with challenges encountered and results discussion.

### 9.1 ARIMA

Autoregressive integrated moving average model represents a generalisation of the autoregressive moving average model. It is usually applied to time series data for forecasting. The “AR” part represents the variable that is regressed based on its previous variables. The “MA” part represents the regression error as a linear combination between the errors that occurred concomitantly and at diverse times in the past. The “I” part denotes the fact that the values of data points have been replaced with the difference between these values and the previous values with the purpose to fit the data as precise as possible. (Wikipedia)

ARIMA is described by three parameters:  $(p, d, q)$ .

$(p)$  stands for the auto regressive part (AR) and it relates to the past values in the regression equation for a series  $X$ , as well as defining the number of lags used in the model. If for example ARIMA (2,0,0) then the series looks as following:

$$X_t = c + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + e_t \quad 30$$

$\varphi_1$  and  $\varphi_2$  are parameters for the model.

$(d)$  stands for the differencing in the integrated part (I). A differenced series is a series where current and previous values have been subtracted  $d$  times. It is applied for stabilizing the series.

$(q)$  stands for the error in the moving average part (MA) and is described as the total of previous error terms  $e_t$ . It determines the number of terms included in the model.

$$X_t = c + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} + e_t \quad 31$$

Differencing, autoregressive, and moving average components of an ARIMA model can be written as a linear equation:

$$X_t = c + \varphi_1 x_{d \ t-1} + \varphi_p x_{d \ t-p} + \dots + \theta_1 e_{t-1} + \theta_q e_{t-q} + e_t$$

where  $x_d$  is  $X$  differenced  $d$  times and  $c$  is a constant. (Dalinina)

### 9.1.1 The Main Challenge

The main requirement for applying ARIMA to time series data is for the data points to be sampled regularly. As mentioned earlier, in chapter 5, dynamometer cards' sampling rate is rather irregular, therefore before proceeding with ARIMA, a separate process for interpolating and resampling data had to be initiated. In order to identify the best resampling time interval, the variation of pump states in every well was analysed. As a result, it was found that in most cases, pump states vary between 2 minutes to 7 minutes. See an example in Figure

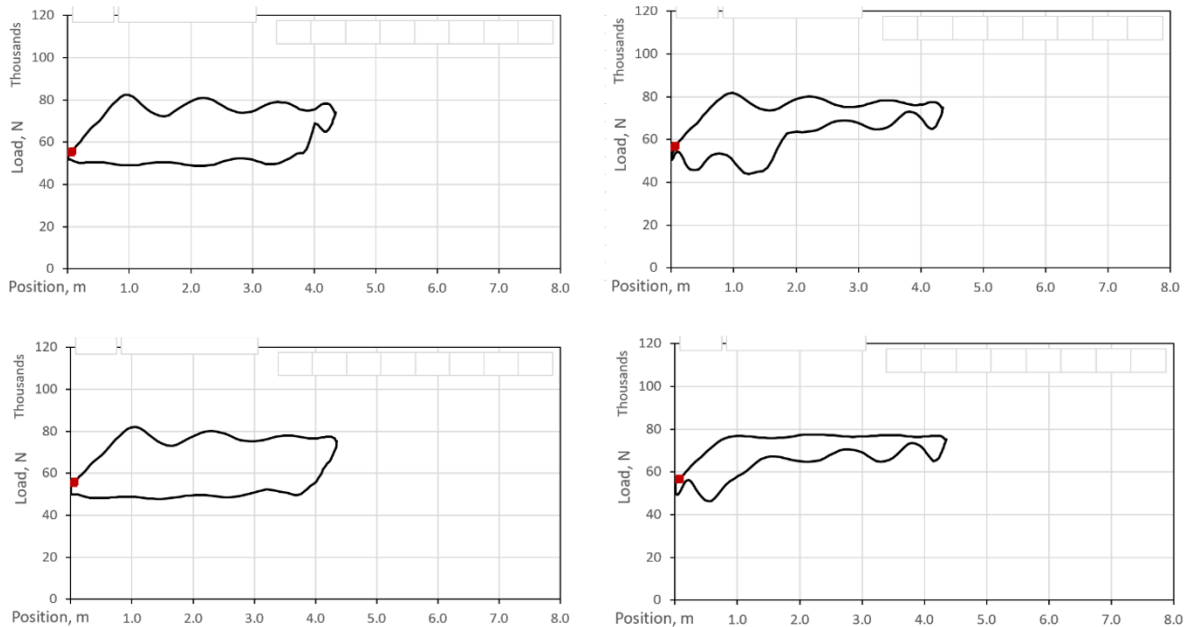


Figure 245: Every 2 minutes pump states variation in a single well

Three resampling intervals have been tested as following, one hour, one minute and every five minutes. In the one-hour interval, a lot of data points were not taken into consideration therefore the risk of running into aliasing was too high. In one minute-interval the data points were too frequent and ARIMA model took too long to be run. Therefore, the five-minute interval was selected as the most optimal resampling rate.

The software used to perform the data modifications and apply ARIMA is R-Studio. In order to achieve the regular data set, an additional timeseries was created containing a timestamp at every fifth minute. This timeseries was bound with existing data points and then cubic interpolation was applied for replacing missing values.

Individual probabilities for every pump states have been used as the values for ARIMA estimation, trend identification and forecasting.

In order to perform the quality check of resampled data points, a visual comparison between the original data and resampled data has been plotted as illustrated in the figures below.

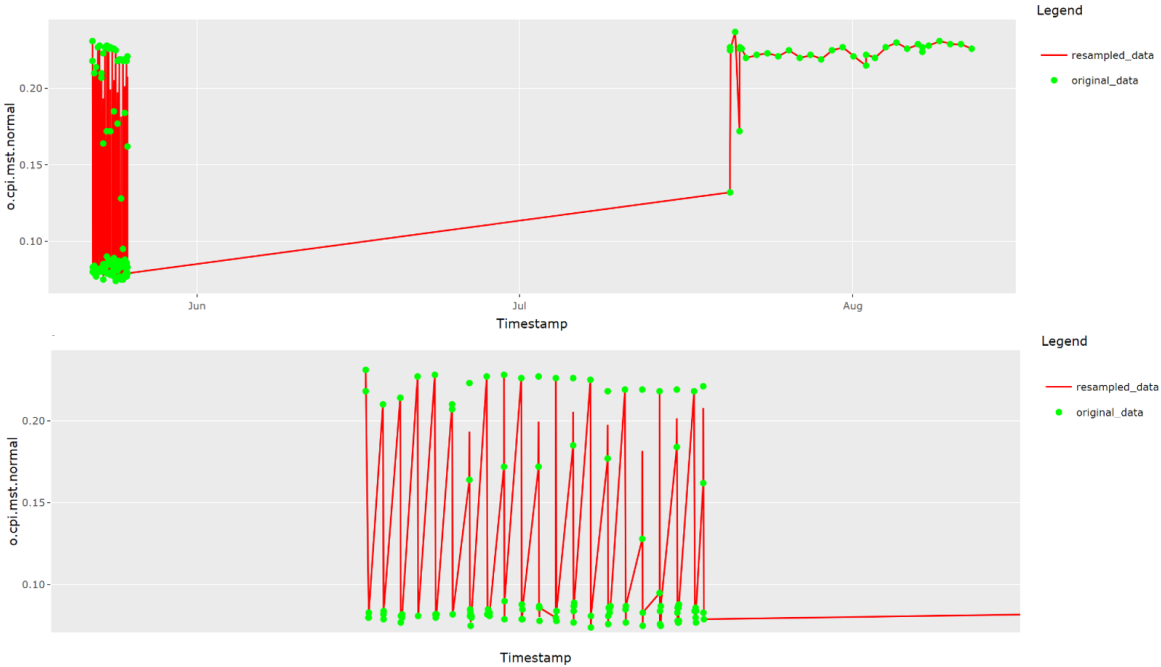


Figure 246: Original data vs resampled data, normal pump state

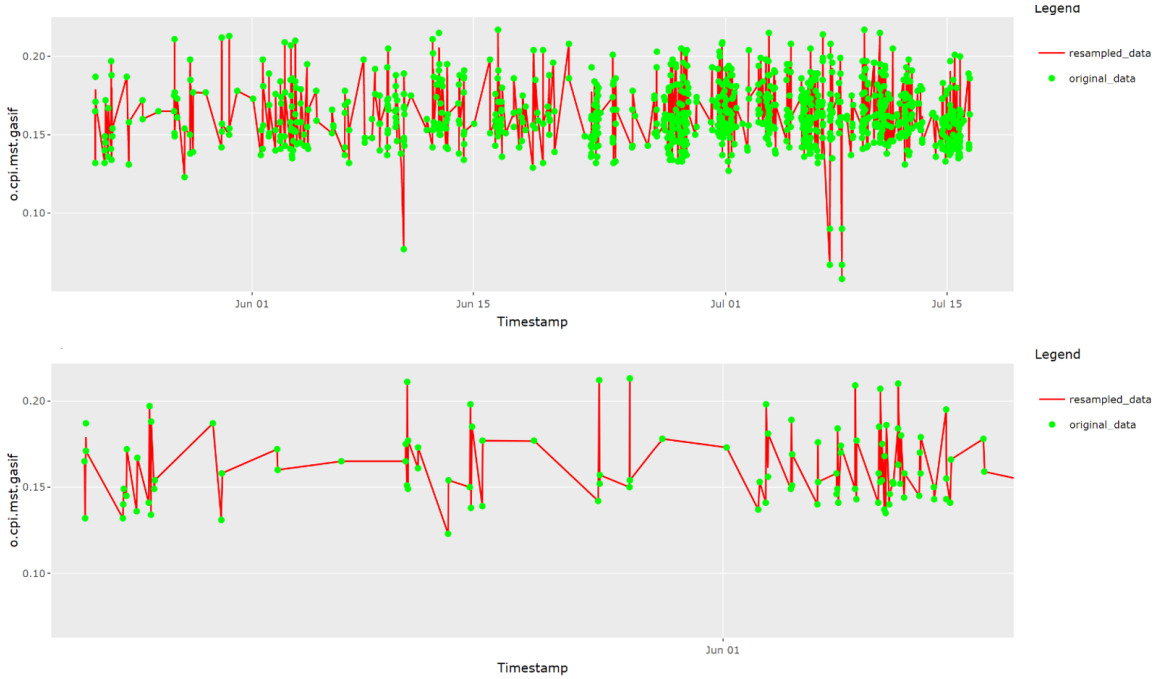


Figure 247: Original data vs resampled data, gas interference pump state

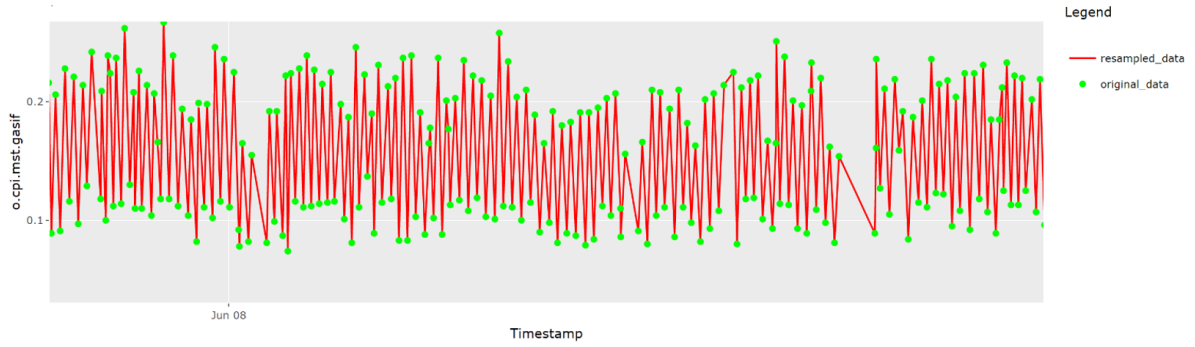
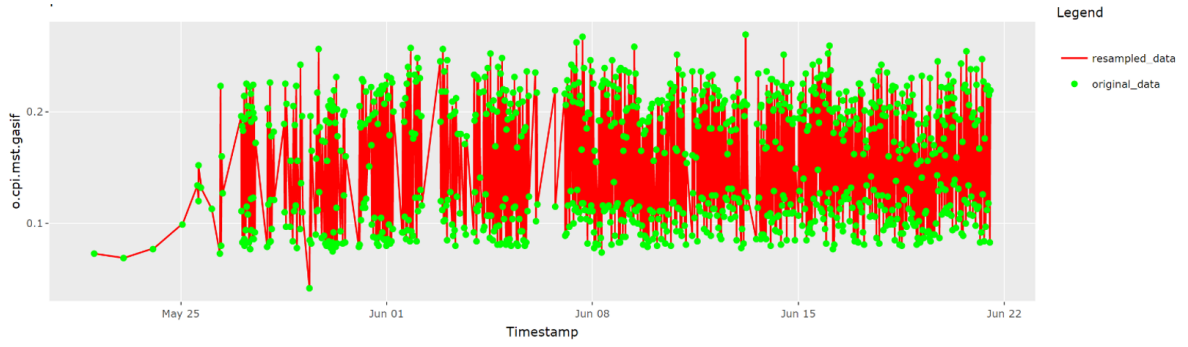


Figure 248: Original data vs resampled data, gas interference pump state

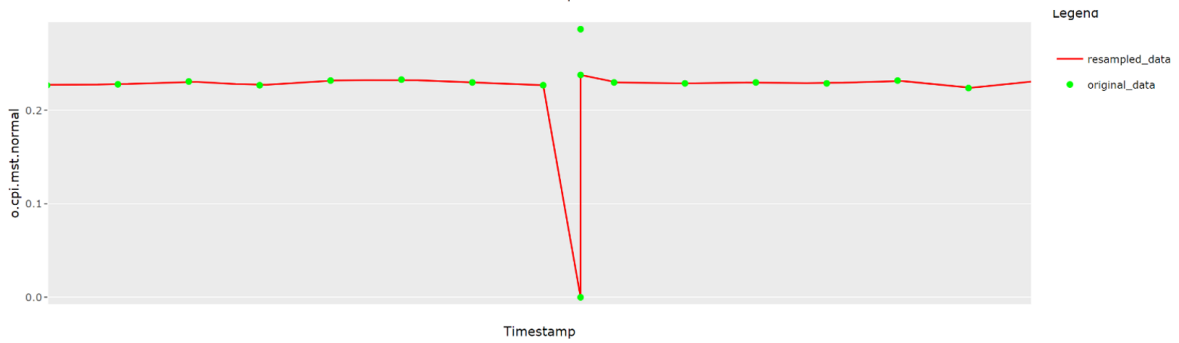
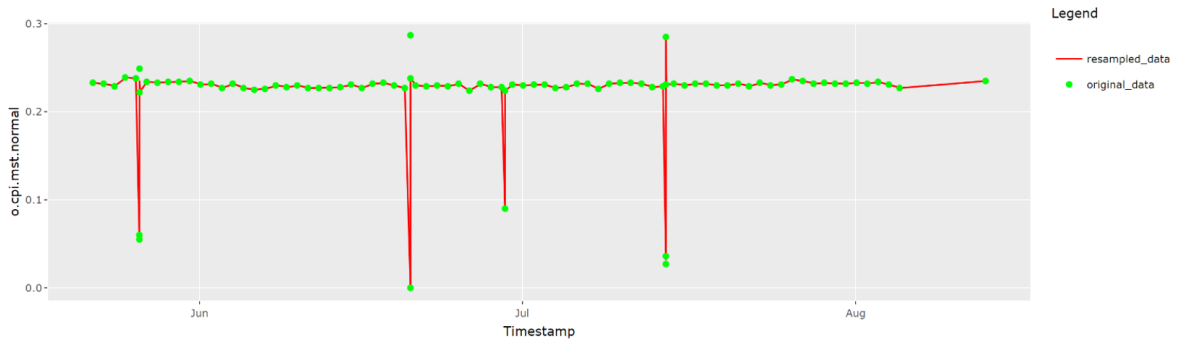


Figure 249: Original data vs resampled data, normal pump state

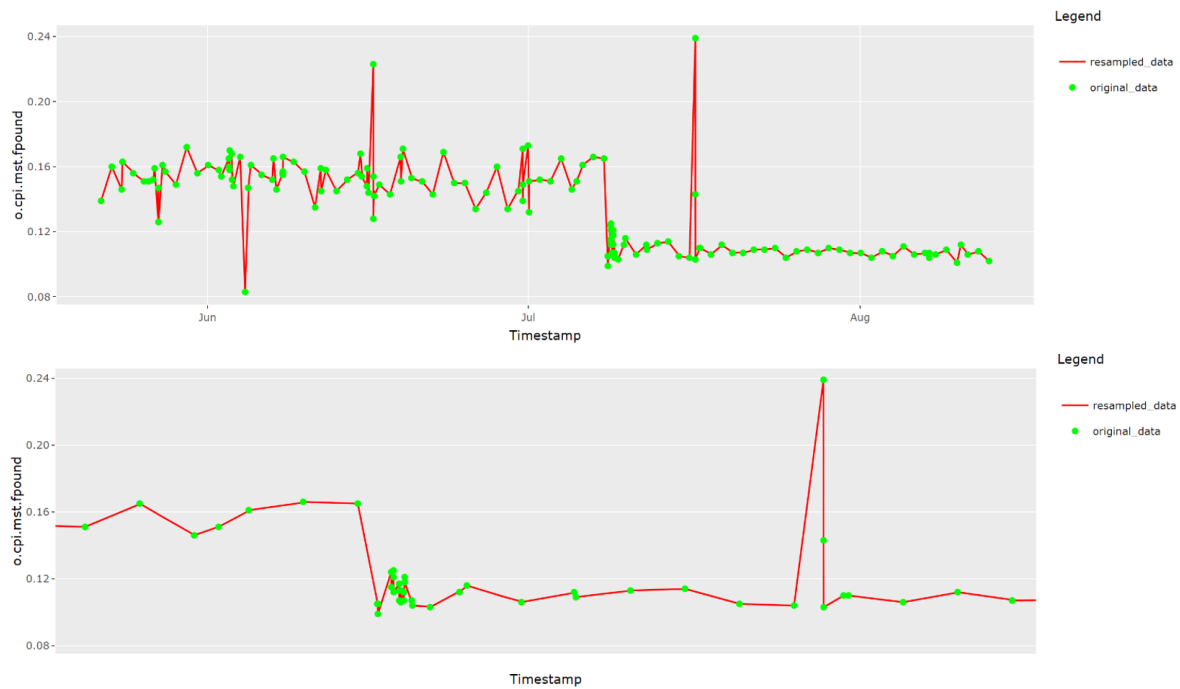


Figure 250: Original data vs resampled data, fluid pound state

Each figure illustrates a test set of a single well in the first picture and a zoomed caption of the same test set in the second picture. It can be noticed that resampled data fits the original data. Of a great importance is to be noticed that already at this step one can identify patterns in pump behaviour, prior to trend identification and forecasting.

In Figure 246 in May there is a change in pump states occurring at exactly every four hours, switching between normal behaviour and fluid pound. This can be an indicator of accumulation of low gas pressure between the valves. Such condition can lead to extremely high stresses and equipment failure. Therefore, some protective measures have to be taken such as installing a smaller downhole pump, reducing the speed of the pumping unit, or shortening the stroke length.

Figure 247 depicts a well with a continuous pump state and a rather irregular sampling rate varying between one sample a day up to multiple samples a day, with some outliers that form negative peaks in the resampled data.

Figure 248 is an example of a good sampling rate with one measurement stored every hour.

Figure 249 illustrates a well where the pump behaves normally most of the time and every once a month there is a fluid pound happening. This is another example of a good sampling rate, one measurement a day has been stored. The peaks created due to fluid pound may not affect negatively the equipment, but it is worth to monitor this well in time in order to identify the moment when the occurrence of abnormal behaviour becomes more often.



Figure 250 is a typical example of a transition from one pump state to another. This can be noticed in the beginning of July where the switch from fluid pound to normal state changes abruptly. Also, the sampling rate after this change has become more regular and often, this is an indicator that some workover or intervention took place in the wellbore altogether with adjusting sensors activity.

## 9.2 Forecasting

ARIMA model was built based on multiple channel estimation and the so called *marima* package was used (multivariate ARIMA). The selection was made in favour of *marima* as the time series in the data set is multivariate and the effect of forecasting is significantly improved. A multivariate time series is a data set that contains more than one time-dependent variable. This means that each of the variables does not depend only on its past values but also on the other variables.

In order to define *marima* models in R -Studio, 0/1 indicator arrays corresponding to the AR part are created. Value 1 indicates the fact that a parameter is to be assumed at that position. Value 0 shows that the parameter corresponding to that position is 0.

A pseudo-regression method is used for estimating the model. In R - Studio, this will allow *marima* to look for the best model in a step-by-step procedure. The model is then iterated until it reaches convergence. (Spliid)

## 9.3 Trend Analysis

According to OECD Glossary of Statistical Terms: “The trend is the component of a time series that represents variations of low frequency in a time series, the high and medium frequency fluctuations having been filtered out.”

There are multiple functions available in R – Studio which can be used for trend calculation. STL is one of the most robust and functional options. It stands for Seasonal and Trend Decomposition using Loess. This method was developed in 1990 by R.B. Cleveland, Cleveland, McRae, and Terpenning. The benefits of this function compared to other available options is that it can handle any seasonality, the user is able to control the smoothness of the trend-cycle, and unusual observations present in the time series do not affect the estimated trend. (Hyndman and Athanasopoulos)

In this case, as the data is multivariate, there is a high chance of multiple seasonality to be present in the time series. Therefore, the MSTL (multiple seasonal decomposition) function was selected. Seasonal components are determined using STL, allowing multiple seasonal

periods. Trend components are extracted from the last iteration of STL. In the situation of non-seasonal time series, data is decomposed only into two components, trend and remainder.

A major advantage of the MSTL function is that it is completely automatic, which results in a better equilibrium between overfitting seasonality and permitting its steady changes over time. (Hyndman and Athanasopoulos)

In the current project, trend analysis was performed for every pump state's individual probability of occurrence.

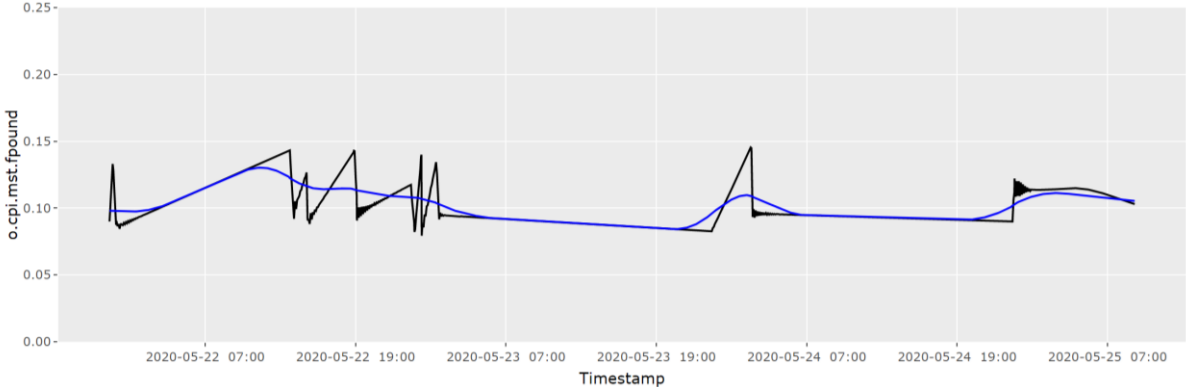


Figure 251: Identified trend for the fluid pound state

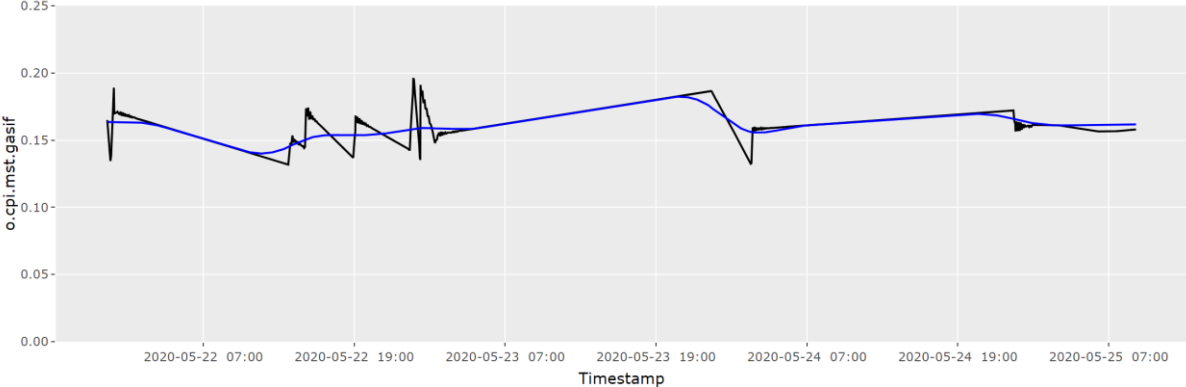


Figure 252: Identified trend for the gas interference state

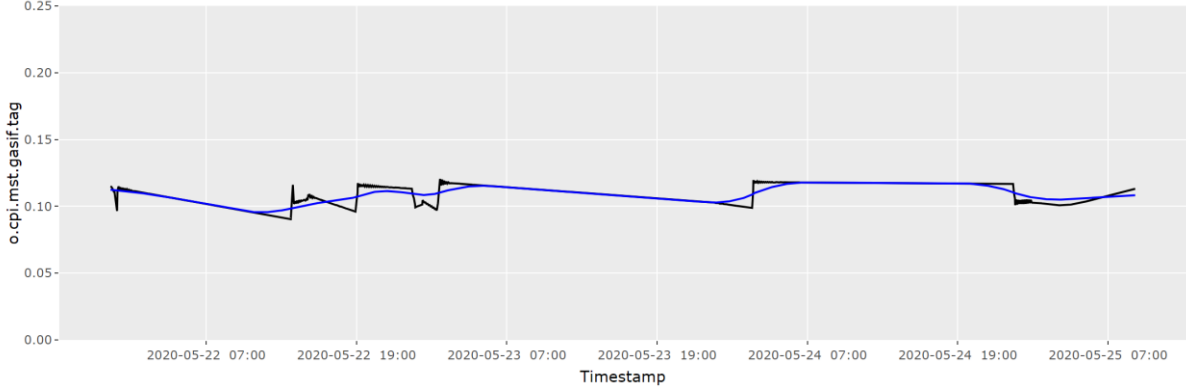


Figure 253: Identified trend for the gas interference + tagging state

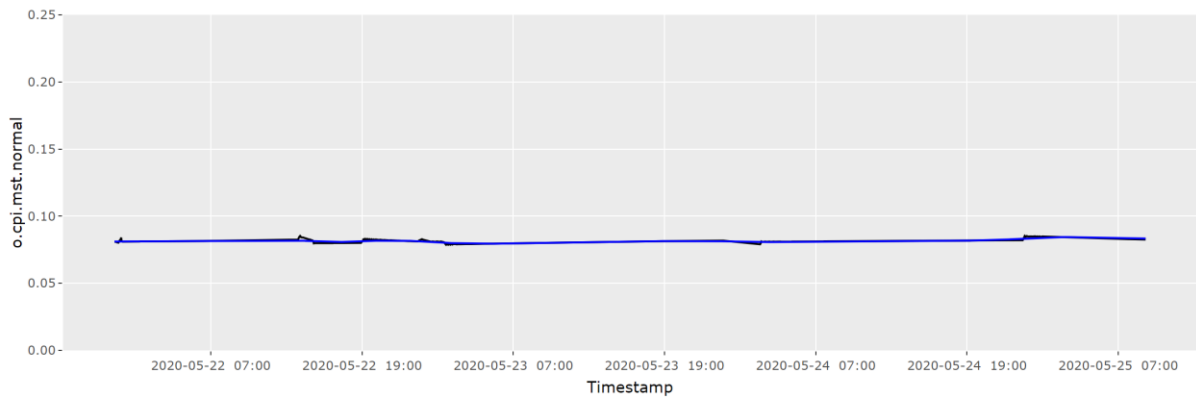


Figure 254: Identified trend for the normal state

Figure 251 - Figure 254 represent some examples for the trend identified in the data. The black line illustrates the original data points plotted over time and the blue line is the trend identified with the MSTL function. The zig-zag shape of the black line is given by the outliers in the original data which does not influence the trend identification process. As it can be seen the identified trend is non-linear and follows the variations of the original data. Up to this point it was important to establish a precise way for trend identification. In the next steps the same procedure will be applied to the forecasted data and compared to the original data trend in order to make sure that forecasted values are trustworthy.

### 9.3.1 Forecasted Values

As it was previously mentioned, the forecasting of future values was performed using MARIMA type model. The function within MARIMA that is responsible for the forecasting is *arima.forecast*.

So as to test the veracity of the forecasted values, the last part of the original dataset was deleted and then forecasted using *arima.forecast*. Afterwards, the forecasted values were compared with the original values as a measure for quality assurance.

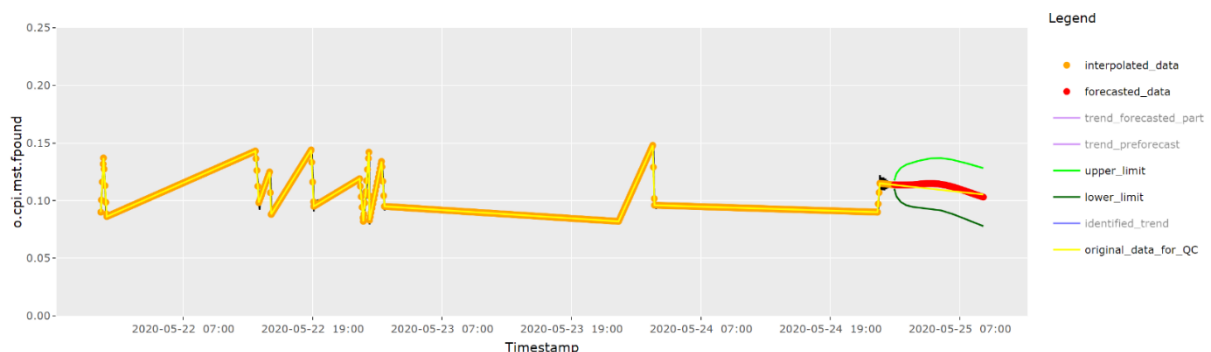


Figure 255: An example of forecasted data points

In Figure the yellow line represents the original data plotted over the interpolated data (orange line) and forecasted data (red line). Green lines represent the confidence interval or prediction interval. It has two limits, upper and lower. They represent the range within which the real

future observations should lie. It is very important to have this interval identified so as to minimize the uncertainty in future values. There is no general method for calculating the prediction interval as it is difficult to evaluate. In the current work a 95% interval was assumed and calculated using standard deviation of the forecast distribution.

It can be observed that interpolated data fits exactly the original data, the forecasted data lies within the confidence interval and follows the original data path. Therefore, it can be concluded that predicted values are trustworthy and almost identical to the real data points. A closer look of the forecasted interval is shown in Figure 256.

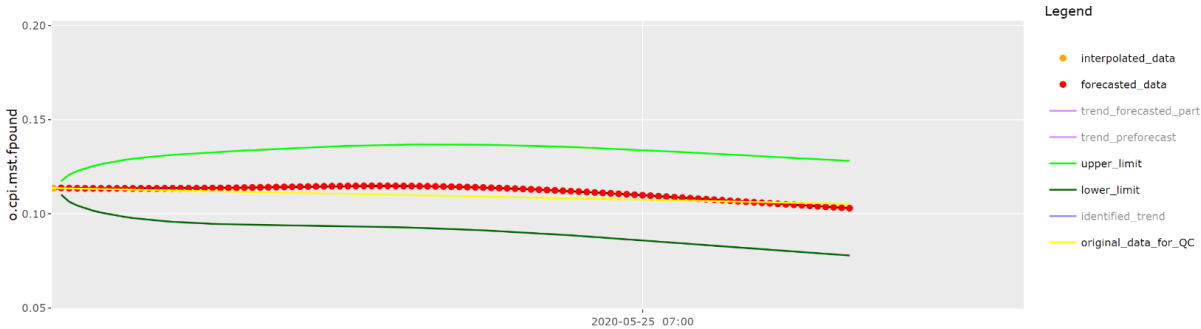


Figure 256: Zoomed view of the forecasted interval

### 9.3.2 Trend of the Forecasted Values

The same method described in 10.3.1 has been applied to identify the trend in forecasted values.

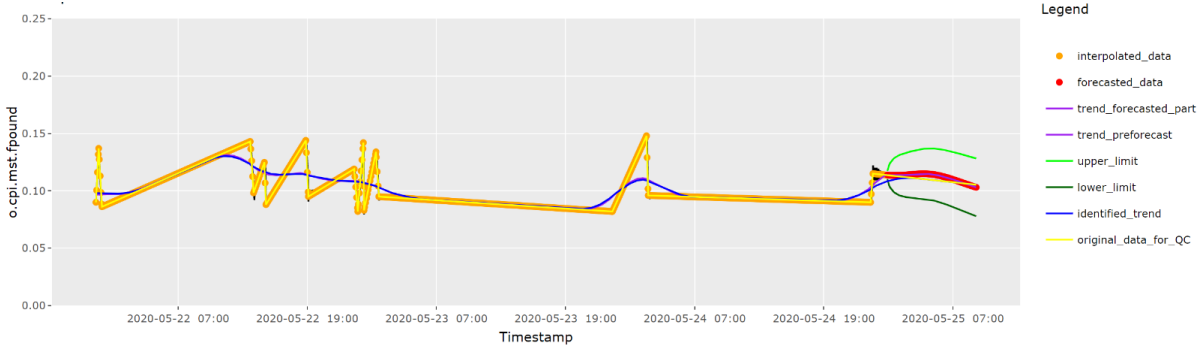


Figure 257: Identified trend for the original data and the forecasted data

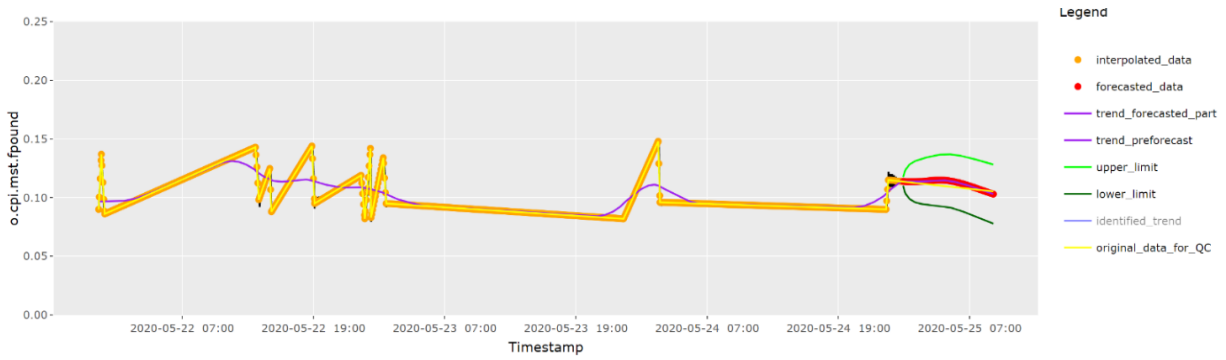


Figure 258: Identified trend for resampled and forecasted data

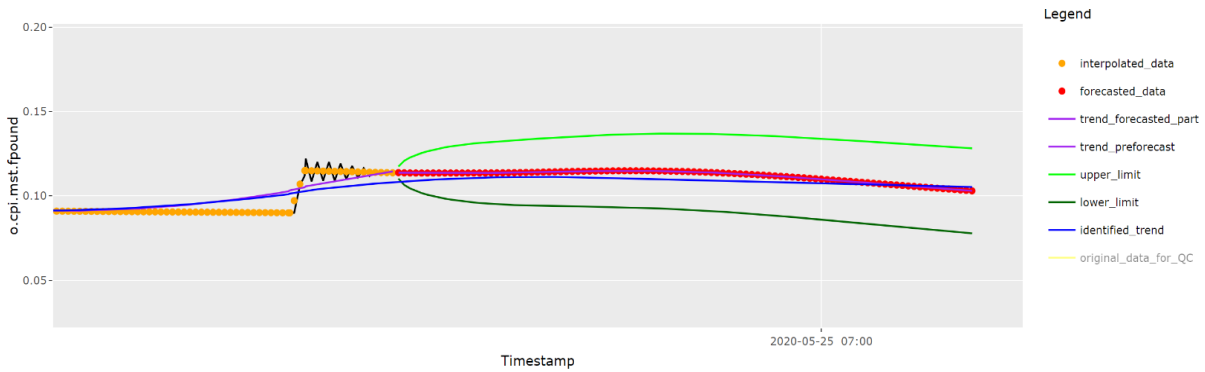


Figure 259: Zoomed view of predicted trend vs forecasted values

In

Figure Figure 257 and Figure 258 it can be seen that the trends of the original data and forecasted data are indistinguishable. When we take a closer look in Figure at the trend of the forecasted data, it is observed that it follows the forecasted values line. This is a good indicator for the trendline veracity.

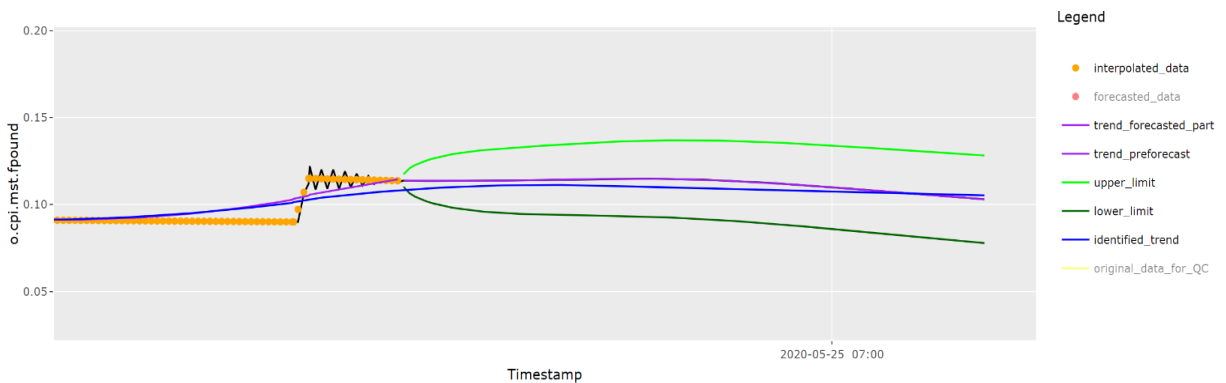


Figure 260: Zoomed view of predicted trendline vs original data trend

An additional quality check measure for the predicted trendline (purple line) is the comparison with the original data trend (blue line), represented in Figure 260. As it is impossible to reach 100% match with the original trend, the error between the predicted data and original data was calculated and resulted in 0.21%.

In conclusion, it should be mentioned that this is an excellent result and provides well acceptable predicted values, very close to the original data.

## 10 Conclusions and Recommendations

The main goal in this work was to investigate and find ways of applying artificial intelligence techniques, machine learning algorithms, data visualization methods and analysis in order to identify trends in sucker rod pumps behaviour and predict future operating states. As a result, the outcome should serve as a tool for production optimization and improving wells operation.

### 10.1 Contributions

Multiple methods and types of artificial neural networks have been studied for identifying sucker rod pump states. Main challenges were understanding provided data, preparing it, building and evaluating models' performance. ARIMA was used for identifying trends in SRP behaviour and predict future pump states. The most important contributions of this work are as following:

- A consistent literature review on machine learning algorithms, different types or artificial neural networks available, application and use in different industries and potential application in oil and gas industry, use of moving average techniques for trends identification in time series data
- Understanding available data, organizing and integrating it in a common format catalogue
- Identifying most common sucker rod pump malfunctions
- Creating testing subsets out of manually labelled data
- Extracting features and using them as input to the neural networks in order to identify surface dynamometer cards quality and pump states
- Training several models based on feed-forward neural networks and selecting the best performing one. Applying the model on the remaining data and evaluating the results related to data quality and pump malfunctions.
- Application of Autoregressive Integrated Moving Average for identifying trends in pump states and forecasting future behaviours.

### 10.2 Findings

Modern technologies provide an endless source of inspiration and possibilities for applying collected data to increase businesses success rate. Nevertheless, most companies do not use available data to its full potential. It is important to understand the magnitude of potential carried by artificial intelligence and machine learning for improvement in decision making processes, risk reduction and cost optimization.

One of the greatest and most demanding challenges in this work was the data acquisition process, data cleaning, filtering and organizing. Due to a very large amount of data available, it was necessary to automatize the aforementioned processes by creating a standardised way of visualizing data, applied for data processing and further investigations.

By extracting features from dynamometer cards and using them as inputs for neural networks, it was possible to develop models for detecting good quality measurements and next for identifying pump states. All results were very accurate and solid. After verifying networks outputs using comparison with real data, it can be stated that developed models are reliable and ready to use for real field operations. Furthermore, they carry a high degree of flexibility which allows the application and extension to new data without retraining or reprogramming.

The application of ARIMA model to identify pump trends behaviour has shown reliable results. By comparing past pump behaviours and ARIMA outputs, it can be proved that predicted trends are trustworthy and can be used in daily operations for avoiding failures in sucker rod pumping systems.

The combination of artificial neural networks models output and ARIMA forecast provides an intelligent diagnostic system for sucker rod pumps.

### **10.3 Recommendations**

After performing a detailed search on what has been done in oil and gas industry related to machine learning, being aware of most common operating issues and the work that has been carried out over the course of this research, following recommendations are suggested:

1. It is of major importance for a company to comprehend the power of data which is stored and perspectives of its application altogether with intelligent algorithms. Surface dynamometer cards alone represent an extensive source for operational improvement of an oil and gas company. Combined with ML algorithms, a series of issues can be addressed and improved.
2. Assuring constant data recording without gaps over the timeline. This will allow an improved trend analysis as information about pumps behaviour is not lost over time.
3. If constant data recording is not possible, then calibrating the recordings storage time intervals to constant ones is recommended. In case of modifications, creating a notifying system containing the reason behind is of important benefit. This will ease the work of the data analyst or production engineer who operates with this data.
4. Coming up with more complex models capable of evaluating additional datasets (e.g. complete failure and workover reports) in order to have a more precise and faster



prediction of pump malfunctions and an understanding of potential reasons behind. For example, by using more diverse data which would contain multiple types of pump malfunctions, later used for the network input. With a different/new input, neural networks can be retrained in order to be able to detect a larger pool of potential pump states and failures. Workover reports, well schematics and other historical data available can be utilized to improve the output of the trend analysis and failure forecasting model and adding automatic recommendations for procedures to be followed in order to avoid the particular unwanted events.

## 11 References

1. Adeyemi, B. J.; Sulaimon, A. A. (2012): Predicting Wax Formation Using Artificial Neural Network. PetroWiki. Available online at <https://onepetro.org/SPENAIC/proceedings-abstract/12NAICE/All-12NAICE/SPE-163026-MS/159347>.
2. Analytics Vidhya: Entropy – A Key Concept for All Data Science Beginners.
3. Anish, Athul (2020): Time Series Analysis.
4. Bouw, E. K. (2017): Analysis of End of Field Life Techniques and predicting Liquid Loading using Artificial Neural Network. Master's dissertaiton. TU Delft, Delft. Geoscience & Engineering.
5. Brownlee, Jason (2019): Why Training a Neural Network Is Hard.
6. Dalinina, Ruslana: Introduction to Forecasting with ARIMA in R.
7. Dugar, Diva (2018): Skew and Kurtosis: 2 Important Statistics terms you need to know in Data Science.
8. Ganti, Akhilesh (2021): Median. Investopedia. Available online at <https://www.investopedia.com/terms/m/median.asp>
9. Lin, David C.; McGowan, Craig; Blum, Kyle P.; Ting, Lena H. (2019): Yank: The Derivative of Force is an Important Biomechanical Variable in Sensorimotor Systems. Journal of Experiemntal Biology. Availbale online at <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6765171/#:~:text=be%20termed%20yank%2C%20and%20defined,not%20just%20a%20scalar%20value>.
10. EAG 1Source (2020): A Guide to Machine Learning in the Oil & Gas Industry.
11. Esmael, Bilal; Arnaout, Arghad; Fruhwirth, Rudolf K.; Thonhause, Gerhard (2013): A Statistical Feature-Based Approach for Operations Recognition in Drilling Time Series. In *International Journal of Computer Information Systems and Industrial Management Applications* 5 (2150-7988), pp. 454–461.
12. Fath, Aref Hasemi; Madanifar, Farshid; Abbasi MAsood (2020): Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems.
13. Fortune Business Insights (2019): Artificial Lift System Market Size.
14. Guresen, Erkam; Kayakutlu Gulgun (2011): Definition of artificial neural networks with comparison to other networks. In : *Procedia Computer Science* 3, pp. 426–433.

15. Hyndman, Rob J.; Athanasopoulos, George: Forecasting: Principles and Practice. A comprehensive introduction to the latest forecasting methods using R. 2nd. Available online at <https://otexts.com/fpp2/>.
16. IBM Cloud Education: Machine Learning.
17. Jones, Tim M. (2017): Unsupervised learning for data classification.
18. Jones, Tim M. (2018): Supervised learning models.
19. Krishnamurthy, Dhruva (2021): The Components of a Neural Network.
20. Kuhl, Frank P.; Giardina, Charles R. (1982): Elliptic Fourier Features of a Closed Contour. In *Computer Graphics and Image Processing* (18), pp. 236–258.
21. Li, Kun; Gao, Xian-Wen; Zhou, Hai-Bo; Han Ying (2015): Fault diagnosis for down-hole conditions of sucker rod pumping systems based on the FBH–SC method. In *Petroleum Science* (12), pp. 135–147.
22. McCullum, Nick (2020): The Ultimate Guide to Recurrent Neural Networks in Python.
23. McLean, Dan; Clay, Roger; Gould, Wayne (1998): Production Management of Electric Submersible Pumps Using Expert System Technology. OnePetro. Available online at <https://onepetro.org/SPEATCE/proceedings-abstract/98SPE/All-98SPE/SPE-49189-MS/190784>.
24. Mitchell, Cory (2020): Rescaled Range Analysis.
25. Yugesh, Verma (2021): A Complete Guide to Sequential Feature Selection. Developers Corner. Available online at <https://analyticsindiamag.com/a-complete-guide-to-sequential-feature-selection/>.
26. Siddheshwar, Vilas Patil (2018): A Review of Dimensionality Reduction in High-dimensional Data Using Multi-core and Many-core architecture. Second Workshop on Software Challenges to Exascale Computing. Available online at [https://scec18.github.io/slides/paper\\_presentation\\_SCEC\\_2018\\_siddheshwar.pdf](https://scec18.github.io/slides/paper_presentation_SCEC_2018_siddheshwar.pdf).
27. Audhkhasi, Kartik; Abhinav, Sethy; Rambhardan, Bhuvana; Shrikanth, S. Narayanan (2013): Generalized Ambiguity Decomposition for Understanding Ensemble Diversity. CiteSeerX. Available online at <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.749.2153&rep=rep1&type=pdf>.
28. Nazi, G. M.; Ashenayi, Kaveh; Lea J.F.; Kemp, Frank (1994): Application of Artificial Neural Network to Pump Card Diagnosis. PetroWiki. Available online at

<https://onepetro.org/CA/article-abstract/6/06/9/70033/Application-of-Artificial-Neural-Network-to-Pump?redirectedFrom=fulltext>.

29. Pascual, Christian (2018): Understanding Regression Error Metrics.
30. PetroWiki: Sucker-Rod Lift.
31. Shukla, Pratik; Iriondo, Roberto (2020): Main Types of Neural Networks and its Applications.
32. Spliid, Henrik (Ed.): Multivariate Time Series Estimation using marima. Symposium i anvendt statistik 2016. Available online at <http://www.dst.dk/da/Statistik/Publikationer/VisPub?cid=025874>.
33. Wikipedia: Autoregressive integrated moving average.
34. Wikipedia: Cross-validation (statistics). Edited by Wikipedia.
35. Wikipedia: History of artificial neural networks.

# 12 Appendices

## Appendix A: Snapshots of Data Matrix

OMV Data Set -2, Inventory				SURFACE DATA																
#	Well Name	Well Name Abbreviation	Well Tag	Time Series				Current			Leakage Current ESP			Active Power			Dyn. Spiegel MURAG			
				From	To	Number of Samples	Sampling Interval (h)	Zeros	Min	Max	Number of Samples	Min	Max	Number of Samples	Zeros	Min	Max	Number of Samples	Min	Max
				Data Inventory 2v01.xlsx																
1	Bernhardtstahl Süd 5	BE S 5	bes5	25/10/2017	02/09/2018	2568	1	1285	44.36	49.42	4	-100	-100	2568	0.08	11.33	2568	1492	1937.89	
2	Bernhardtstahl Süd 6	BE S 6	bes6	25/10/2017	25/01/2018	2208	1	2208	39.11	43.81	4	-100	-100	2208	8.66	14.22	2208	500.88	499.69	
3	Erdpress 24	ERD 24	erd24	25/10/2017	02/09/2018	2567	1	2567	42	44.34	4	-100	-100	2567	5.01	10.76	2567			
4	Hochletten 20	HL 20	h20	25/10/2017	02/09/2018	2567	1	2567	26.85	28.21	4	-100	-100	2567	5.26	7.23	2567			
5	Hochletten 66	HL 66	h66	27/12/2017	21/02/2018	1343	1	1343	0	0	4	-100	-100	1343	0	0	1343			
6	Matzen 214	MA 214	ma214	25/10/2017	21/02/2018	2856	1	2856	2856	0	0	-100	-100	2856	2856	0	0	2856		
7	Matzen 568	MA 568	ma568	25/10/2017	21/02/2018	2856	1	2843	21.44	43.3	4	-100	-100	2856	0	0	2856			
8	Matzen 569	MA 569	ma569	25/10/2017	30/12/2017	72	1	72	67.68	68.72	4	-100	-100	72	4.85	8.87	72			
9	Prottes 217	P 217	p217	25/10/2017	21/02/2018	2855	1	2855	2855	0	0	-100	-99.99	2855	2855	0	0	16	-99.98	-99.95
10	Prottes 218	P 218	p218	25/10/2017	12/02/2017	912	1	912	34.79	36.74	4	-100	-100	912	11.96	13.39	912			
11	Prottes 221	P 221	p221	22/11/2017	20/02/2018	2160	1	2160	58.88	67.02	23	-100	-99.96	2160	0.11	21.58	2160	15	-99.99	-99.98
12	Prottes T Süd 3b	P T S 3b	pts3b	25/10/2017	19/02/2018	2808	1	2808	2808	0	0	-100	-100	2808	2808	0	0	2808		
13	Prottes T West 1	P T W 1	ptw1	25/10/2017	18/02/2018	2783	1	2704	2704	0	0	-100	-100	2783	2783	0	0	2783		
14	Schönkirchen 231	S 231	s231	25/10/2017	01/03/2018	1680	1	1680	1680	0	0	-100	-100	1680	1680	0	0	1680		
15	Schönkirchen 439	S 439	s439	31/10/2017	20/02/2018	2677	1	2677	1	0	0	-100	-100	2677	1	0	2677	1	0	0
16	Schönkirchen 442	S 442	s442	25/10/2017	19/02/2018	2808	1	2808	0	26.91	34.15	-100	-100	2808	1.42	15.82	2808	73.41	-375.06	
17	Schönkirchen 443	S 443	s443	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
18	Schönkirchen 446	S 446	s446	25/10/2017	16/02/2018	2736	1	2736	2736	0	0	-100	-100	2736	2736	0	0	2736		
19	Spannberg 23	SPA 23	spa23	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
20	Spannberg Süd 3a	SPA S 3a	spa3a	01/04/2018	21/02/2018	1152	1	1152	1152	0	0	-100	-100	1152	1152	0	0	1152		
21	Spannberg Süd 4a	SPA S 4a	spa4a	01/04/2018	21/02/2018	1152	1	1152	1152	0	0	-100	-100	1152	1152	0	0	1152		
22	Spannberg Süd 5	SPA S 5	spa5	12/05/2017	21/02/2018	1872	1	1872	1872	0	0	-100	-100	1872	1872	0	0	1872		
23	Spannberg Süd 6	SPA S 6	spa6	01/04/2018	21/02/2018	1152	1	1152	1152	0	0	-100	-100	1152	1152	0	0	1152		
24	Spannberg Süd 7	SPA S 7	spa7	23/11/2017	21/02/2018	2160	1	2150	2150	0	0	-100	-100	2150	2150	0	0	2150		
25	Spannberg Süd 8	SPA S 8	spa8	23/11/2017	21/02/2018	2160	1	2160	2160	0	0	-100	-100	2160	2160	0	0	2160		
26	Spannberg Süd 11	SPA S 11	spa11	25/10/2017	19/12/2017	1320	1	1320	1320	0	0	-100	-100	1320	1320	0	0	1320		
27	Bernhardtstahl Süd 4	BE S 4	bes4																	
28	Bernhardtstahl Süd 8	BE S 8	bes8																	
29	Bockfless 201	BO 201	bo201																	

OMV Data Set -2, Inventory				PRODUCTION DATA								DYNAMOMETER CARDS (digital)							
#	Well Name	Well Name Abbreviation	Well Tag	Date Interval		Timestamps	Brutto	WC	Density	GOR	Oil	Gas	_Dynamogramm		Date Interval		_aprol_	n_	
				From	To	#	(m3/d)	(%)	(g/m3)	(t/d)	(m3/d)	Number of Measurements	From	To	Number of Measurements	Number of Measurements			
				Data Inventory 2v01.xlsx															
1	Bernhardtstahl Süd 5	BE S 5	bes5	16/11/2009	09/06/2018	173	173	104	102	102	173	173	268,042	01/09/2017	08/02/2018	NA	46,416	129	
2	Bernhardtstahl Süd 6	BE S 6	bes6	22/07/2014	24/04/2018	100	100	57	56	56	100	100	193,098	01/09/2017	24/01/2018	NA	45,680	3	
3	Erdpress 24	ERD 24	erd24	22/07/2014	12/05/2018	118	118	88	88	88	118	118	437,748	04/08/2017	11/11/2018	NA	45,469	137	
4	Hochletten 20	HL 20	h20	11/11/1992	27/05/2018	567	567	515	513	513	567	567	417,376	04/08/2017	19/09/2018	NA	42,142	232	
5	Hochletten 66	HL 66	h66	28/01/1999	15/05/2018	418	418	369	369	369	418	418	181,925	27/12/2017	11/11/2018	NA	NA	NA	
6	Matzen 214	MA 214	ma214	19/10/1992	31/05/2018	4179	4179	4093	4095	4078	4179	4179	388,528	04/09/2017	20/02/2018	NA	NA	NA	
7	Matzen 568	MA 568	ma568	08/05/2017	24/05/2018	49	49	36	35	35	49	49	368,365	10/10/2017	11/11/2018	NA	NA	NA	
8	Matzen 569	MA 569	ma569	17/10/2017	30/04/2018	50	50	36	30	30	50	50	14	27/12/2017	13/08/2018	NA	NA	NA	
9	Prottes 217	P 217	p217	16/03/2017	16/05/2018	38	38	31	31	31	38	38	283,150	23/08/2017	11/11/2018	NA	39,942	220	
10	Prottes 218	P 218	p218	30/03/2017	17/05/2018	50	50	33	33	33	50	50	23,046	10/10/2017	17/10/2018	NA	NA	NA	
11	Prottes 221	P 221	p221	11/08/2017	13/05/2018	27	27	20	20	20	27	27	1,180	22/11/2017	11/11/2018	NA	NA	NA	
12	Prottes T Süd 3b	P T S 3b	pts3b	27/03/2014	18/05/2018	115	115	88	88	88	115	115	202,997	22/08/2017	11/11/2018	NA	NA	NA	
13	Prottes T West 1	P T W 1	ptw1	18/11/2009	15/05/2018	258	234	167	161	161	258	258	80,611	01/09/2017	11/11/2018	NA	NA	NA	
14	Schönkirchen 231	S 231	s231	30/06/2008	12/07/2008	278	276	192	184	184	276	278	110,153	25/10/2017	25/10/2018	NA	NA	NA	
15	Schönkirchen 439	S 439	s439	10/04/2013	30/05/2018	165	165	151	151	151	165	165	50,621	31/10/2017	11/11/2018	NA	NA	NA	
16	Schönkirchen 442	S 442	s442	27/06/2013	17/04/2018	171	171	143	140	140	171	171	392,139	21/09/2017	11/11/2018	NA	46,416	138	
17	Schönkirchen 443	S 443	s443	11/06/2013	24/07/2017	84	84	56	51	51	84	84	2	04/10/2017	04/10/2017	NA	NA	NA	
18	Schönkirchen 446	S 446	s446	20/01/2015	16/05/2018	121	121	109	108	108	121	121	156,202	31/08/2017	11/11/2018	NA	45,378	133	
19	Spannberg 23	SPA 23	spa23	18/07/2008	14/05/2018	273	273	141	132	132	273	273	5,363	05/09/2017	11/11/2018	NA	NA	NA	
20	Spannberg Süd 3a	SPA S 3a	spa3a	02/09/1999	26/04/2018	369	369	306	303	303	369	369	68,713	04/01/2018	20/02/2018	NA	NA	NA	
21	Spannberg Süd 4a	SPA S 4a	spa4a	18/10/2000	19/04/2018	309	309	253	251	251	309	309	178,252	04/01/2018	11/11/2018	NA	NA	NA	
22	Spannberg Süd 5	SPA S 5	spa5	10/10/2000	20/04/2018	347	347	297	295	295	347	347	204,126	05/12/2017	11/11/2018	NA	NA	NA	
23	Spannberg Süd 6	SPA S 6	spa6	01/09/2000	27/04/2018	352	352	278	276	276	352	352	93,977	04/01/2018	04/09/2018	NA	NA	NA	
24	Spannberg Süd 7	SPA S 7	spa7	01/06/2002	28/05/2018	239	239	210	209	209	239	239	262,351	23/11/2017	11/11/2018	NA	NA	NA	
25	Spannberg Süd 8	SPA S 8	spa8	01/08/2002	29/04/2018	345	345	201	197	197	345	345	163,897	23/11/2017	07/11/2018	NA	NA	NA	
26	Spannberg Süd 11	SPA S 11	spa11	10/02/2015	25/04/2018	57	57	46	46	46	57	57	136,555	23/10/2017	11/11/2018	NA	NA	NA	
27	Bernhardtstahl Süd 4	BE S 4	bes4										4,668,431			NA	NA	NA	
28	Bernhardtstahl Süd 8	BE S 8	bes8										10,097	13/08/2018	16/10/2018	NA	NA	NA	
29	Bockfless 201	BO 201	bo201										129	26/08/2018	11/11/2018	NA	NA	NA	

OMV Data Set -2, Inventory				DYNAMOMETER CARDS (Images)			SRP ANALYSIS			WO		MEDIUM	PUMP	WELL SCHEMAT
----------------------------	--	--	--	----------------------------	--	--	--------------	--	--	----	--	--------	------	--------------

## Appendix B: QC Models Performance Evaluation

### Model C (best performance)

Validation

Class	bad	good	
	92.4%	99.8%	
<b>bad</b>	<b>292</b>	9	97.0%
<b>good</b>	24	<b>4423</b>	99.5%
	24	9	33
	316	4432	4748

Learning

Class	bad	good	
	94.4%	99.9%	
<b>bad</b>	<b>922</b>	12	98.7%
<b>good</b>	55	<b>13524</b>	99.6%
	55	12	67
	977	13266	14243

Testing

Class	bad	good	
	92.7%	99.8%	
<b>bad</b>	<b>357</b>	9	97.5%
<b>good</b>	28	<b>4354</b>	99.4%
	28	9	37
	385	4363	4748

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	99.3%	96.1%	98.1%	97.2%
Learning	99.5%	97.1%	99.2%	98.1%
Testing	99.2%	96.3%	98.5%	97.3%

**Model A**

Validation

<b>Class</b>	<b>bad</b>	<b>good</b>	
	91.1%	99.8%	
<b>bad</b>	<b>288</b>	7	97.6%
<b>good</b>	28	<b>4425</b>	99.4%
	28	7	35
	316	4432	4748

Learning

<b>Class</b>	<b>bad</b>	<b>good</b>	
	93.4%	99.9%	
<b>bad</b>	<b>913</b>	13	98.6%
<b>good</b>	64	<b>13253</b>	99.5%
	64	13	77
	977	13266	14243

Testing

<b>Class</b>	<b>bad</b>	<b>good</b>	
	91.2%	99.8%	
<b>bad</b>	<b>351</b>	7	98.0%
<b>good</b>	34	<b>4356</b>	99.2%
	34	7	41
	385	4363	4748

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	99.3%	95.5%	98.5%	97.0%
Learning	99.5%	6.7%	99.1%	97.9%
Testing	99.1%	95.5%	98.6%	97.0%

**Model B**

Validation

<b>Class</b>	<b>bad</b>	<b>good</b>	
	91.1%	99.9%	
<b>bad</b>	<b>288</b>	5	98.3%
<b>good</b>	28	<b>4427</b>	99.4%
	28	5	33
	316	4432	4748

Learning

<b>Class</b>	<b>bad</b>	<b>good</b>	
	93.8%	99.9%	
<b>bad</b>	<b>916</b>	11	98.8%
<b>good</b>	61	<b>13255</b>	99.5%
	61	11	72
	977	13266	14243

Testing

<b>Class</b>	<b>bad</b>	<b>good</b>	
	91.2%	99.7%	
<b>bad</b>	<b>351</b>	12	96.7%
<b>good</b>	34	<b>4351</b>	99.2%
	34	12	46
	385	4363	4748

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	99.3%	95.5%	98.8%	97.1%
Learning	99.5%	96.8%	99.2%	98.0%
Testing	99.0%	95.4%	98.0%	96.7%



## Appendix C: Pump States Models Performance Evaluation

Model B (best performance): (iMLP+0c i163b-l8s-o10b L2/R+)AMx10.10e5.4(k5.4)

Validation	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset anchor	
		97.3%	0%	97.4%	60%	100%	99.8%	25%	0%	100%	100%	
		401	3	304	8	97	2056	2	0	48	84	45
	fpound	401	3	2			3		4			97.1%
	fpound+tag		0									0%
	gasif	5		304	8		2	4				94.1%
	gasif+tag			1	12			2				80%
	ltv					97						100%
	normal	6		5			2056					99.5%
	overtravel							2				100%
	poff								0			0%
	tagging									48		100%
	unset anchor										84	100%
		11	3	8	8	0	5	6	4	0	0	45
		412	3	312	20	97	2061	8	4	48	84	3049

Class	Learning										
	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset anchor	
	98.8%	0%	99%	38.8%	100%	100%	23.8%	15%	98%	99.7%	
fpound	1162	6	1			1		17			97.9%
fpound+tag		0									0%
gasif	11	3	925	50			6				92.7%
gasif+tag			4	33			10				70.2%
ltv					267						100%
normal	3		2	1		6161				1	99.9%
overtravel			2	1			5				62.5%
poff								3			100%
tagging									147		100%
unset anchor										322	100%
	14	9	9	52	0	1	16	17	3	1	122
	1176	9	934	85	267	6162	21	20	150	323	9147

Testing	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		97.2%	0%	96.9%	50%	100%	99.8%	16.7%	0%	100%	100%		
		384	3	2			3		5				96.7%
	fpound	384	3	2			3		5				96.7%
	fpound+tag		0										0%
	gasif	4		349	17		2	5					92.6%
	gasif+tag	1		6	17								70.8%
	ltv					104							100%
	normal	5		3			2025						99.6%
	overtravel							1					100%
	poff	1							0				0%
	tagging									39			100%
	unset anchor										73		100%
		11	3	11	17	0	5	5	5	0	0		57
		395	3	360	34	104	2030	6	5	39	73		3049

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	98.5%	68%	77.1%	72.2%
Learning	98.7%	67.3%	82.3%	74.1%
Testing	98.1%	66.1%	76%	70.7%

**Model A: (iMLP+0c i163b-f7s-o10b L2/R+)AMx10.5e5.5(k5.5)**

Validation	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		96.9%	0%	97.9%	60%	100%	99.7%	0%	0%	98.2%	100%		
		381	2	327	12	111	2013	0	0	55	100		
	fpound	381	2	1			6		5				96.5%
	fpound+tag		0										0%
	gasif	9		327	8		1	2		1			94%
	gasif+tag			3	12			6					57.1%
	ltv					111							100%
	normal	3		1			2013						99.8%
	overtravel			2				0					0%
	poff								0				0%
	tagging									55			100%
	unset anchor										100		100%
		12	2	7	8	0	7	8	5	1	0		50
		393	2	334	20	111	2020	8	5	56	100		3049

Learning	Class											
	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
	99.2%	0%	99%	42.5%	100%	100%	30%	0%	100%	100%		
fpound	1179	7	2					16				97.9%
fpound+tag		0										0%
gasif	7	1	935	49			6					93.7%
gasif+tag			7	37			8					71.2%
ltv					269							100%
normal	3					6206						100%
overtravel							6					85.7%
poff								0				0%
tagging									134			100%
unset anchor										274		100%
	10	8	9	50	0	0	14	16	0	0		107
	1189	8	944	87	269	6206	20	16	134	274		9147

Testing	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		96.8%	0%	98.8%	37.5%	100%	99.7%	0%	0%	95.7%	100%		
		<b>388</b>	<b>0</b>	<b>324</b>	<b>12</b>	<b>88</b>	<b>2021</b>	<b>0</b>	<b>0</b>	<b>45</b>	<b>106</b>		
	fpound	11	2	19		3	4		1				96%
	fpound+tag												0%
	gasif			2			3						89%
	gasif+tag							3					70.6%
	ltv												100%
	normal	2		1	1								99.8%
	overtravel												0%
	poff												0%
	tagging												100%
	unset anchor												100%
		13	5	4	20	0	6	7	8	2	0		65
		401	5	328	32	88	2027	7	8	47	106		3049

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	98.4%	65.3%	64.7%	65%
Learning	98.8%	67.1%	74.8%	70.7%
Testing	97.9%	62.8%	65.5%	64.2%

**Model C: (MLP+0c i163b-l8s-o10b L2/R+)Amx10.3e5.4(k5.4)**

Validation	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		98.4%	0%	98.4%	7.4%	100%	99.8%	0%	0%	198%	98.9%		
	fpound	376	3	2			3		9	1			95.4%
	fpound+tag		0										0%
	gasif	3		315	24		1	6					90.3%
	gasif+tag				2								100%
	ltv					88							100%
	normal	3		3	1		2067				1		99.6%
	overtravel							0					0%
	poff								0				0%
	tagging									48			100%
	unset anchor										93		100%
		6	3	5	25	0	4	6	9	1	1		60
		382	3	320	27	88	2071	6	9	49	94		3049

Learning	Class											
	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
	98.3%	0%	99.2%	7.3%	100%	99.9%	0%	0%	98.6%	100%		
fpound	1184	8	6			4		13				97.4%
fpound+tag		0										0%
gasif	17	2	967	76		3	17		2			89.2%
gasif+tag				6			3					66.7%
ltv					290							100%
normal	3		2			6116						99.9%
overtravel							0					0%
poff								0				0%
tagging									137			100%
unset anchor										291		100%
	20	10	8	76	0	7	20	13	2	0		156
	1024	10	975	82	290	6123	20	13	139	291		9147



Testing	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		97.5%	0%	97.7%	6.7%	100%	100%	0%	0%	100%	100%		
		387	0	4	28	90	2058	0	7	49	95		
fpound		387		4					7				97.2%
fpound+tag			0										0%
gasif		4	2	304	28		1	9					87,4%
gasif+tag					2								100%
ltv						90							100%
normal		6		3			2058						99.6%
overtravel								0					0%
poff									0				0%
tagging										49			100%
unset anchor											95		100%
		10	2	7	28	0	1	9	7	0	0		64
		392	2	311	30	90	2059	9	7	49	95		3049

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	98%	60.1%	68.5%	64%
Learning	98.3%	60.3%	65.3%	62.7%
Testing	97.9%	60.2%	68.4%	64%

**Model D: (MLP+0c i163b-l8s-l8s-o10b L2/R+)Amx10.10e5.1(k5.1)**

Validation	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		97.8%	0%	99.7%	60%	100%	99.9%	0%	0%	100%	100%		
		<b>403</b>	3				1		4				98.1%
	fpound+tag		<b>0</b>										0%
	gasif	5		<b>311</b>	8		1	3					94.8%
	gasif+tag			1	<b>12</b>			5					66.7%
	ltv					<b>97</b>							100%
	normal	4					<b>2059</b>						99.8%
	overtravel							<b>0</b>					0%
	poff								<b>0</b>				0%
	tagging									<b>48</b>			100%
	unset anchor										<b>84</b>		100%
		9	3	1	8	0	2	8	4	0	0		35
		412	3	312	20	97	2061	8	4	48	84		3049

Learning	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		99%	0%	98.9%	44.7%	100%	100%	0%	0%	98.0%	100%		
		1164	6	1			2		20	1			97.5%
	fpound	1164	6	1			2		20	1			97.5%
	fpound+tag		0										0%
	gasif	7	3	924	46		1	4		2			93.6%
	gasif+tag			7	38			17					61.4%
	ltv					267							100%
	normal	5		2	1		6159						99.9%
	overtravel							0					0%
	poff								0				0%
	tagging									147			100%
	unset anchor										323		100%
		12	9	10	47	0	3	21	20	3	0		125
		1176	9	934	85	267	6162	21	20	150	323		9147

Testing	Class	fpound	fpound+tag	gasif	gasif+tag	ltv	normal	overtravel	poff	tagging	unset	anchor	
		98.5%	0%	98.1%	55.9%	100%	99.9%	0%	0%	100%	100%		
		389	3				1		5				97.7%
	fpound		0										0%
	fpound+tag	4		353	15		1	4					93.6%
	gasif			6	19			2					70.4%
	gasif+tag					104							100%
	ltv	6		1			2028						99.9%
	normal							0					0%
	overtravel								0				0%
	poff									0			0%
	tagging										39		100%
	unset anchor											73	100%
		6	3	7	15	0	2	6	5	0	0		44
		395	3	360	34	104	2030	6	5	39	73		3049

Sub-set	Accuracy	Precision	Recall	F1-score
Validation	98.9%	65.7%	65.9%	65.8%
Learning	98.6%	64.1%	65.2%	64.6%
Testing	98.6%	65.2%	66.2%	65.7%