Probabilistic modelling of oil rig drilling operations for business decision support: a real world application of Bayesian networks and computational intelligence.

FOURNIER, F.A.

2013

PROBABILISTIC MODELLING OF OIL RIG DRILLING OPERATIONS FOR BUSINESS DECISION SUPPORT: A REAL WORLD APPLICATION OF BAYESIAN NETWORKS AND COMPUTATIONAL INTELLIGENCE.

FRANÇOIS A. FOURNIER

PhD 2013

PROBABILISTIC MODELLING OF OIL RIG DRILLING OPERATIONS FOR BUSINESS DECISION SUPPORT: A REAL WORLD APPLICATION OF BAYESIAN NETWORKS AND COMPUTATIONAL INTELLIGENCE.

FRANÇOIS A. FOURNIER

A thesis submitted in partial fulfilment of the
requirements of the
Robert Gordon University
for the degree of Doctor of Philosophy.

This research programme was carried out in
collaboration with ODS-Petrodata Ltd. and IHS Inc.,
under the Knowledge Transfer Partnership (006922),
with funding from the Technology Strategy Board, UK.

March 2013

Abstract

This work investigates the use of evolved *Bayesian networks* learning algorithms based on *computational intelligence meta-heuristic* algorithms. These algorithms are applied to a new domain provided by the exclusive data, available to this project from an industry partnership with ODS-Petrodata, a business intelligence company in Aberdeen, Scotland. This research proposes statistical models that serve as a foundation for building a novel operational tool for forecasting the performance of rig drilling operations. A prototype for a tool able to forecast the future performance of a drilling operation is created using the obtained data, the statistical model and the experts' domain knowledge. This work made the following contributions: applied K2GA and *Bayesian networks* to a real-world industry problem; developed a well-performing and adaptive solution to forecast oil drilling rig performance; used the knowledge from industry experts to guide the creation of competitive models; created models able to forecast oil drilling rig performance consistently with nearly 80% forecast accuracy using either Logistic regression or *Bayesian network* learning using *genetic algorithms*; introduced the node juxtaposition analysis graph which allows the visualisation of the frequency of nodes links appearing in a set of orderings, providing new insights when analysing node ordering landscapes; explored the correlation factors between model score and model predictive accuracy and showed that the model score does not correlate with the predictive accuracy of the model; explored a method for feature selection using multiple algorithms and drastically reduced the modelling time by multiple factors; proposed new fixed structure *Bayesian network* learning algorithms for node ordering search-space exploration. Finally, this work proposed real-world applications for the models, based on current industry needs, such as *recommender systems*, an oil drilling rig selection tool, a user-ready rig performance forecasting software and rig scheduling tools.

Keywords

Bayesian Networks; Industry application; Oil and Gas; Metaheuristic; Business Intelligence; Computational Intelligence; Weka; Recommender Systems; Scheduling; Forecasting

Table of Contents

List of Figures, Tables and Equations

Acknowledgements

First of all, I would like to thank bullet points for the clarity they provide to any text every time they are used. Then, I would like to thank the producers of my favourite whiskies for the inspiration they gave me during the long hours of writing:

- Bruichladdich Rocks 14 years old
- Clynelish 14 years old
- Cragganmore 12 years old
- Lagavulin 16 years old
- Bunnahubhain Darach ùr
- Glenfiddich 15 distillery edition
- Jura (Prophecy & Elixir)

Then I would like to thank Star Trek in all of its iterations (TOS, TAS, TMP, TWOK, TSFS, TVH, TFF, TUC, TNG, GEN, FC, INS, NEM, DS9, VOY, ENT, ST), from 1965 to 2394, for the inspiration it gave me when I was not writing and for the excuses to procrastinate it also provided.

Human-wise, I would mostly like to thank my partner and friends:

- Alexandra Weber, for being here, for all her enduring support and for the proofreading of every draft but even more for the encouragement to actually complete the writing in the dark hours of slow progress.
- My entire family for supporting me at a distance even though I have been away from home for 9 years and counting.
- Graeme Nesham, for his kindness, support and all the useful details about oil rig specifications. From crane to water depth, he knows it all!
- John Hartley, for his friendship and for teaching me years ago nearly all I know about oil exploration and extraction.
- François Delobel, Serge Kruppa and Eric Werkhoven for showing me what computing was all about and inspiring me to innovate continuously, early in my computing career.

The important people behind this work are also:

- John McCall, Andrei Petrovski and Peter Barclay, my supervisors, for the advice and for all the draft reading they endured.
- Robert Steven and John Hartley for the expertise they provided to the project.

Finally, I would like to thank the following organisations:

- ODS-Petrodata Ltd. and IHS Inc. for employing me before, after and somehow during the length of my research efforts.
- The Technology Strategy Board for the funding of the research project through the Knowledge Transfer Partnership program (partnership 006922).

There are lots of other people I would like to thank for where I am today. Thank you all, all along the road.

Chapter 1: Introduction

Effective use of resources in today's industry is crucial to the competitiveness of any company. Business decision support is primarily data and model-driven. In this thesis, I am investigating methods and applications of data models to offshore oil drilling rigs data collected by market intelligence company ODS-Petrodata[1]. The operation of oil drilling rigs is highly expensive. Their effective selection is crucial to the running of offshore exploration and exploitation projects. The aim of the project is to use appropriate *computational intelligence* techniques to gain added value from the data ODS-Petrodata have collected by virtue of providing their services. Ultimately, the intention is to improve the decision making support made available by ODS-Petrodata with their data analytics.

The following aims are part of the overall research target:

- Investigate the use of statistical tools and *Bayesian networks* learning algorithms. Apply algorithms to the new domain provided by the exclusive data, available to this project.
- Research and develop a novel operational tool for forecasting success of rig operations. Using the data obtained, the statistical model and experts' domain knowledge, create a tool to predict the future success of a drilling operation.

This project provides an opportunity to consider a real-world industry problem. The combination of automatic structure learning and experts' knowledge provides the project with an invaluable opportunity to develop new data models in order to forecast oil drilling rig performance and create or improve various applications for corporate decision support. Moreover, the approach demonstrated in this project has a wider scope. It provides a stepping stone for the study of the application of *computational intelligence* in a real-world environment. The methodologies and findings are most likely applicable to many real-world problems and bring the potential to accelerate the rate at which new theoretical computing research is applied commercially.

This project was undertaken as a Knowledge Transfer Partnership (KTP). Knowledge Transfer Partnership is a UK-wide government-sponsored programme to encourage collaboration between businesses and universities. The aim is to enable businesses to improve their competitiveness, productivity and performance through the transfer of cutting edge technology. [1] ODS-Petrodata is a market intelligence company specialised in the upstream offshore oil and gas industry [2]. By

[1] Acquired by IHS Inc. in April 2011.

developing their offer using automated analysis tools, the company can demonstrate objectivity and scientific rationale in their analysis. The use of these tools, in addition to extending the scope of their services, may provide the statistical backing to the experts' analyses, embedded in their various market reports.

The problem at hand is challenging due to the number of variables and the number of states each of these variables can assume. That issue is tackled with various techniques discussed in this thesis, such as clusters, discretisation, machine learning, data analytics and expert manual selection. One important limitation of human analysis is that it is always restricted by what the analyst can remember at any given time. The cross-influence of multiple variables might often be replaced by a 'gut-feeling' the analyst develops over time with experience. This leads to a lack of explainability of the forecasts. Incidentally, explainability is one of the major strengths of the technique used in this project: *Bayesian networks*. It is deemed possible to interpret and explain the results more easily using the model probabilities, than with other *artificial intelligence* methods such as *neural networks* [3], [4], [5], [6]. This novel application of *Bayesian networks* helps to fill in the gap in the current industry practice and to contribute to quantifying some of the uncertainty in decision making.

Figure 1: An oil drilling rig (semi-submersible): J.W.Mclean, stacked (parked) in Invergordon, Scotland[2]

[2] © Copyright 2010 Graeme Nesham, reproduced with permission

In this research, I am investigating the suitability of *Bayesian networks* learning algorithms using *evolutionary algorithms* to model oil drilling rig data as well as some possible ways to involve human expertise in the model building of a real-world problem. My main contributions are to the application of model learning techniques and approaches to the real-world data model building exercise. Within the context of my research, I am also facilitating the development of model-based decision support for *Rigs* and *Wells* exploitation in the oil and gas industry. Finally, in the light of the results provided by this research I am suggesting novel applications of *Bayesian* models in other research fields (*forecasting*, *recommender systems*, *scheduling*, etc.) and in the real world (oil and gas industry).

1.1 Objective and Motivation

This research is realised in partnership with market intelligence company ODS-Petrodata Ltd. and the Robert Gordon University, Aberdeen. This setting provides the project with an access to experts in the domain of the data used. In addition to creating a learned data model, a second potential gain from this project consists in taking advantage of the expert guidance in learning the data structure.

The overall project was initially drafted to answer a specific problem identified in today's market intelligence industry. The initial aim of the project was defined over the year preceding the start of the project and sought to investigate the possibility of automatically and efficiently scheduling oil drilling rigs using modern modelling techniques. That objective was then updated to match the real-world needs from the industry more closely. In the current industry practice, a company needing an oil drilling rig for a drilling operation will publish a demand into a broking system or send '*invitations to tender*' to various rig owners or contractors. After identifying a suitable rig in the responses, the price is set using rig valuation and day rate indications as a negotiation tool. During the negotiation a date suitable both for the rig and within the client's requirements is also agreed on. One of the sub-optimal operations is the selection of a suitable rig. A company will often manage its global operations on a regional level, ignoring neighbouring regions in most of the cases. Regional schedulers cooperate only to a limited degree and often rely on a local pool of oil drilling rigs. Exchanging rigs between regions is a rare practice. This research aims to help improve the use of rig fleets. With the help of the company experts, I identified that the common part to all of the rig decision making is the rig performance. The entire decision making, from tendering to scheduling, is based on that information.

The literature review and the domain of application encouraged me to investigate in greater detail how some part of the process could be performed more efficiently. By conducting research on that project, I intend to investigate the following questions:

- What are suitable tools to determine the correlated factors of large datasets (such as the *Wells*, *Rigs* and *Demands* data, issued from the oil and gas market intelligence industry)?
- Which factors are necessary or desirable to be able to forecast with an acceptable accuracy the oil drilling rig performance?

1.2 Ethical Considerations of Technological Developments

The three year joint research project was partly funded by the company in which the results of the research will be applied. I have, however, made every effort possible for the project's results to remain completely impartial and re-producible. This research has a potential impact on real world activities and, as such, its impact should be considered carefully. For that, I reviewed some of the literature in order to understand better the ethical dilemma the project's development might be exposed to. In [7], Cummings explores the bias in decision making, induced by task automation. I am expecting limited issues during the research phase but ethical issues may arise in the future, depending on how the final product issued from this research is used and its commercialisation. Considering one example of application – the use of the models developed as the base for a *recommender system* – Bergemann helps understand the impact of *recommender systems* on the sales environment in [8]. This supports the thinking process of figuring out what is ahead, when developing applications for the real-world inspired from research. More generally, technology has the potential to impact lives. A relevant source of information, highlighting the potential issues that can be encountered by any technology development, is found in the talk by Horowitz [9]. In this same presentation, he highlights the power that is derived from data. He exposes the following questions: "if we can do something, should we?" and "What's the right thing to do?" He "reviews the [...] new powers that technology gives [...]: to know more -and more about each other- than ever before. [...]" In that same talk, Horowitz says "There's not a formula. There's not a simple answer." He mentions Hannah Arendt, who voiced the idea that: "most evil done in this world is not done by people who choose to be evil. It arises from not thinking" [9]. Against this background, Horowitz proposes the following steps to review the ethical considerations when making decisions: take responsibility, explain how you made that decision, get a point of view from someone in a different field, think about problems differently than technologists, list human considerations, make ethical decisions, care about what happens with the technologies developed. In my work, I have clearly highlighted the basis of each of my decisions to ensure a reproducibility of the results. I

have consulted business and technological experts who believe the technology developed can deliver value to the industry without creating abnormal risks. Further review will then be done at the time of productising the result of this industry-oriented research.

1.3 Publications and Presentations

Some parts of my research have been published and publicly presented in the following publications, reports and public presentations of the project:

- Fournier, F. A., McCall, Y., Wu, J., Petrovski, A., Barclay, P. J., Application of evolutionary algorithms to learning evolved Bayesian network models of rig operations in the Gulf of Mexico, IEEE UKCI 2010.
- Fournier, F. A., McCall, J., Petrovski, A., Barclay, P. J., evolved Bayesian network models of rig operations in the Gulf of Mexico: preliminary experiments, poster for SICSA/SEABIS workshop.
- Fournier, F. A., McCall, J., Petrovski, A., Barclay, P. J., Evolved Bayesian network models of rig operations in the Gulf of Mexico, IEEE CEC 2010 / WCCI 2010.
- Fournier, François. *Recommender Systems*: Technical report and literature review [Internet]. Version 13. Knol. 2010 Feb 18. Available from: http://knol.google.com/k/françois-fournier/recommender-systems/ (retrieved November 2011).
- Fournier, F. A., Rig operations data modelling for decision support, KTP associate seminar, presentation at Culloden visitor centre, Inverness, introduction by Barclay, P. J., 18 March 2010.
- Fournier, F. A., On the building of a model: The work of the KTP project on building a probabilistic model based on Rigs and Wells data, ODS-Petrodata corporate talk, 17 November 2010.

1.4 Thesis Organisation

This thesis is organised into the following parts:

- **Chapter 1** introduces the background and motivation to this research and exposes the ethical framework used.
- **Chapter 2** investigates the literature of the field of application, including an exploration of the real-world context to this research. The second part of that chapter is focused around

developing a detailed overview of the *Wells* and *Rigs* data used in this project, the methodologies and assumptions for their selection, manipulations, combinations and exploitation.

- **Chapter 3** reviews the literature for the algorithms and techniques used in this research, their origin, design and implementation.
- **Chapter 4** reviews the benchmarks used to compare the results as well as the performances of the various algorithms.
- **Chapter 5** provides an analysis of the evolved *Bayesian network* models in the context of rig operations in the Gulf of Mexico.
- **Chapter 6** explores the performance of the algorithms and benchmarks the results. It also presents additional work done on the analysis of the data.
- **Chapter 7** reviews a real world technology application and exhibits additional possible novel and tangible applications that could be derived from this research in future work such as oil drilling rig selection and drilling duration forecast.
- Finally, **Chapter 8** summarises proposed future work and overall conclusions.

Chapter Summary: The first chapter introduces the overall research target for this work: investigating the use of statistical tools and *Bayesian networks* learning algorithms and developing a novel operational tool for forecasting success of rig operations. The chapter exposes the objective and motivation for this work, exposes the research questions and approaches ethical considerations of technological developments. Finally, the chapter lists earlier publications produced as part of this research and provides an overview of the thesis organisation.

Chapter 2: Offshore Oil Drilling Rig: Commercial Background

Oil drilling rigs are operated by contractors who hire out their services to oil companies for both exploration and exploitation. Typically, a rig operating offshore in the Gulf of Mexico can cost from $400K to $600K per day [10]. With rig operations lasting weeks or even months at a time, variations in the efficiency with which rigs are operating can affect profitability by millions of dollars. It is, therefore, important to be able to identify and analyse factors affecting efficiency.

There are many ways of defining efficiency. Oil drilling rig efficiency is usually assessed by industry experts on the basis of practical experience [11] but there is currently no industry-wide standardised approach for the objective measurement and prediction of efficiency. Efficiency on its own cannot be directly compared between rigs without considering many external and influencing factors such as weather, the specific nature of the geological layers being drilled through, and other environmental or managerial factors. Determining which factors are relevant and how they are related is largely left to the judgment of managers and other experts in the field. Their approach is based mainly on empirical observations and experience. In some cases, the rig selected for a job will be over-specified or under-specified, leading either to unnecessary expenses or poor outcomes, such as significant delay. It is this uncertainty surrounding the rig selection process that identifies rig operations management as an application area for data modelling.

This chapter explores the background to this research (section 2.1 and section 2.2). Then I elaborate on the problem at hand (section 2.3). Finally, I explore the data used in this research (section 2.4).

2.1 Background - Offshore Drilling

The offshore drilling process is split into two main steps: exploration and exploitation (or production). Various offshore drilling platform types exist within those two categories. Table 1, drawn from Nergaard [12], summarises the main different types of offshore drilling platforms available. More detailed information on those principal rig types is presented in Table 2.

Table 1: Offshore drilling platform types in the Gulf of Mexico

Exploration	Floaters	Semis-Submersible
		Ships
	Bottom Support	Jack-ups
Production / Exploitation	Surface platforms	Permanent
		Tenders
	Subsea	Semi-Submersible
		Ships
		Jack-ups

Rig owners contract rigs to drilling companies for specific pre-established needs in both exploration and production. The offshore drilling market is dynamic, highly competitive, and regionally specific [13]. Key differences across regions are legislative and geological variations as well as sea conditions; however, cultural differences and practices across regions and across companies also often impact results. To better understand the subject matter at hand, it will prove useful to consider Freudenrich's presentation of a simplified path to oil and gas production in [14]. More details on how oil is extracted offshore are provided in the documents [15], [16], delivered to the US congress following the BP Deepwater Horizon oil spill. Oil is located using various survey methods and tools, including geological analysis, gravity meters, magnetometers and seismology technologies. Once a site is selected, it is surveyed to find its boundaries. Then an oil drilling rig is brought on site and starts drilling. As drilling progresses, a specialist fluid called '*mud*' circulates through the pipe and out of the drill bit to float the rock cuttings out of the hole. When a pre-set depth is reached, the drill bits are removed from the hole and a *steel-and-cement casing* is installed. When reaching the final depth, various logs and tests are performed and samples are taken for analysis. The well is then secured and installed in order to let the oil flow in a controlled manner. Once the oil is flowing, the oil drilling rig is removed from the site and production equipment is set up to extract the oil from the well.

Table 2: Main offshore oil drilling rigs type definition[3]

Type	Description
Semi-Submersible	Semi-submersible rigs are floating platforms that obtain their buoyancy from ballasted watertight pontoons located below the ocean surface and, thus, below the wave movements. The operating decks are kept high above the surface. They need assistance to move between locations and are used for water depth greater than 120 meters.
Ships (Drillships)	Drillships are ships fitted with drilling apparatus and equipped with dynamic positioning systems to maintain relative positions. They can drill in deep water and can independently move between locations. They can be used in depth of more than 2500 meters.
Jack-ups	Jack-ups are self-elevating mobile platforms with 3 or 4 legs, capable of raising themselves over the surface of the sea. They are used in shallow waters generally up to 120 meters deep and require assistance to move between locations.
Permanent (Fixed)	Fixed platforms are anchored directly into the sea bed. They are used to extract long-term oil deposits and cannot be moved without being fully dismantled. They can be used in depth of up to about 520 meters.

[3] More information on oil drilling rigs types can be found in "Types of offshore oil rigs" by McLendon [214]. Information in this table has also been provided by the partner company experts. Other platform types exist but are not described here.

Regarding performance, Harris [11] explains that no two rigs perform the same but that "consistently good results are a good indication of a rig's capability" [11]. He highlights three main criteria, used to select rigs: technical suitability, price, and availability. Osmundsen et al. [17] highlight more evaluation criteria for selection. In no particular order, they state that typical evaluation criteria can be: *expertise*, *financial strength, day rates*, *ability to complete on time*, *compliance with regulations*, *operational efficiency and achievements*, *Health and Safety Executive* (*HSE*) *system and culture*, *High Pressure High Temperature* (*HPHT*) *capability*, *crew expertise and experience* [17]. The data detailed by Osmundsen are the starting point to select and prepare the data described in section 2.4.

2.2 Background - The Rig Tendering Process

Rig tendering is the process by which a company contracts a rig for a given operation. According to Harris [11], a successful operation depends on many factors which are difficult to measure. The tendering process for selecting a rig has remained largely unchanged since his publication in 1989. The variability in the drilling process and the fact that the tendering process takes place in advance generates uncertainty. This creates the need to find ways to quantify and reduce uncertainty in predicting the performance of potentially available rigs so that an informed selection can be made.

When selecting a rig for a drilling programme, an operator typically has three main criteria: technical suitability, price, and availability. Some technical parameters are absolute and determine the type of rig and equipment. Examples are water depth, pressure and temperature ratings, etc. However, alternatives can sometimes be suitable; for example, semi-submersibles have been known to operate in jack-up water depth [11]. Many of the other technical requirements included in an invitation to tender are often preferences rather than necessities. It is commonly recognised that, if the well is drilled efficiently, a higher priced bid can lead to a lower overall cost. Likewise, a low priced bid can become expensive if accidents extend the drilling time [11]. Considering availability, requirements will tend to be stricter in a low-demand rig market compared to the situation when rigs are in short supply. However, the market maintains a system of '*extension options*' which is one of the main sources of uncertainty on rig availability [11]. These extension options are pre-negotiated exclusive rights in a contract to extend the contract of the rig to perform additional work. This work is usually dependent on the outcome of the main contract. Another potential measure, according to Harris, is the rig's safety ratings as "there is a correlation between a good operation and a good safety record" [11].

The usual process starts with a company in search of a contractor sending out an invitation to tender. The contractor will then respond to the invitation, presenting various options available, depending on the nature of the potential non-compliance (the rig responding to the tender does not match all of the specifications). With all the responses considered, there will appear some variation in potential and decisional tradeoffs [11]. In recent years, a move toward the search for quality has been made and bidders in Europe are often asked to provide percentage downtime and indicators of drilling efficiency for the past six wells including water depth, mooring time, loss of time, repair time [17]. However this information is not often available in most regions across the globe.

2.3 The Problems at Hand

In order to properly focus this work, I obtained a range of utilisation scenarios from one of ODS-Petrodata Rigs expert, Robert Steven. I worked with Robert Steven and John Hartley, ODS-Petrodata's experts, within the Rigs and the Wells departments respectively. Each of them has over 20 years experience working and analysing the data I am using in this work. They are also responsible for the collection and aggregation of most of the data available to this research.

Performance being defined as the drilling speed for the purpose of this work, performance and duration are intrinsically linked: the performance of a rig on a given well is the drilled distance divided by its duration (in days). All the scenarios provided to the project by Robert Steven (VP Rigs, ODS-Petrodata, 2010) rely on rig performance or well duration:

- **Comparing well duration outcomes with different oil drilling rigs:** Oil companies can enter known location, well depth, water depth and other technical parameters for a planned well and then compare forecast well duration outcome for different oil drilling rigs. This can be used to contribute to the rig selection process in a tender exercise.
- **Preliminary well cost budgeting:** User could enter location, well depth, water depth and other technical parameters but probably leave rig null. This will show likely well duration that can then be used to provide rough cost estimates prior to the completion of detailed well planning.
- **Choosing rig specification/category prior to tender:** Prior to going to the market with a tender, an oil company can review impact of different rig specifications on the well duration outcome to enable the optimum rig specification to be identified.
- **Benchmarking contractor performance:** Operator could view the outcome of wells for various rig managers/contractors to identify the company most likely to achieve the best

drilling performance. Rig manager/contractor could do the same to benchmark their performance against their competitors.

- **Benchmarking operator performance:** Operator could view the outcome for wells according to different operators as a way of benchmarking their own performance versus other operators. Oil companies in license partnerships could also use the same approach to establish if the operator of their license was achieving the drilling performance standard that other operators would achieve.
- **Regulatory authorities:** There may be opportunities for regulatory authorities to check submitted operator well plans versus the outcome suggested most likely by the forecast. This, for example, could identify plans that were out of the norm for one reason or another.

These scenarios have been used as the basis for this investigation in order to create a suitable tool to assist business decision making for the oil and gas industry in regards to oil drilling rig performance.

2.4 Gulf of Mexico Dataset

The datasets used in this thesis are based on Rigs and Wells data sourced by ODS-Petrodata Ltd. within its market intelligence commercial databases. ODS-Petrodata's RigPoint [10] database covers worldwide offshore oil drilling rig contracts and activities. Currently, it covers over 25 years of historical rig activity. Since 2007, they added to their databases the coverage of Wells data. This extension covers both historical and current drilling activities within the offshore industry for the Gulf of Mexico. Historical and current data are collected in several tables. More generally, there are multiple levels of data included in Rigs and Wells databases, including operational data (*water depth*, *footage drilled*, *operation dates* and *durations*, etc.), technical data (*cantilever capacity*[4], *water depth rating*, *age of the rig*, etc.) and time data (*start date*, *spud date*, *total depth date*, *termination date*, *time on site*). Using the techniques detailed in this part, these complex real-world data were extracted into useable datasets.

The data selection procedure comprises the following steps:

- The first step was to list the data available in ODS-Petrodata's databases (section 2.4.1).
- From this list, the variables were selected with a limited amount of incomplete data (20-25% maximum missing data, based on the expert's recommendations) and with a potential relevance to the performance prediction problem as suggested by the scenarios drafted by

[4] More details on the technical terms can be found in the Glossary (page 161).

the domain expert in section 2.3. These fields have been discussed with the domain experts and selected, based on their relevance to the domain of rig performance (section 2.4.2).

- Finally, a method was used to increase the data available by devising an algorithm to automatically link data from separate databases (section 2.4.3).

2.4.1 Review of Available Data

The data available from *ODS-Petrodata* is split into two databases. The *Rigs21s* database and the *Well extension* databases contain 348 data tables (including types and lookup tables) and over ten thousand fields. The main tables considered are *Rig*, *Deployment* and *Contracts* from the *Rig21s* database as well as the *Well* table from the *Well extension* database. Those are accompanied by additional tables providing more details on each object (for example: *dates, tonnage, on-board tools specifications, depths, well casing sizes and installation depths*, etc.).

The databases are heavily tied to the company history. Initially, there were 2 separate databases within two separate companies with different markets. *Offshore Data Services (ODS)* was collecting the Gulf of Mexico data and *Petrodata* was collecting North West Europe data. To develop their markets, the companies merged and the databases were integrated. This explains some inconsistencies and the large amount of data missing in some fields. For example, the field named '*ShoreBase*' collects the name of the city a rig is attached to. It was maintained from 2002 to 2004 for the Gulf of Mexico data only but was then retired, owing to a lack of commercial value. The data remain in the database and little documentation on the motivation for the data collection is available other than the knowledge of some of the senior company employees. There are similar occurrences of data fields through the tables of the database, adding a level of complexity to the data selection process.

The current organisation of the data collection teams still reflects this legacy. The main changes and improvements in the company procedures and organisation are that the different teams share sources, procedures and exchange data related to each other's specialities. At the time of writing, the data teams concerned by the collection of relevant data are split in two collaborating departments: *Rigs* and *Wells*, both of which maintain their own separate data-based products. The teams are also geographically separated between Houston (Texas, USA), Aberdeen (Scotland, UK) and Singapore. Each local team tends to prioritize the coverage of its own geographical location. The creation of the *Wells Extension* database, along with the *Wells* department is a recent addition to the company assets (2007). The collaboration between the teams being recent, the historical data is found to be naturally segregated in the database and links are often missing between *Rig*,

Deployments and *Well Activities*[5]. Recent data is profiting from some new communication procedures between departments and is becoming more comprehensive. When inputting a new *Well* or a new *Deployment*, the data entry teams now systematically consult each other to match the information. This has led to 221 links in the database between *Wells* and *Deployments* being established. Those links are used as a basis to the work on automated linking later in this chapter.

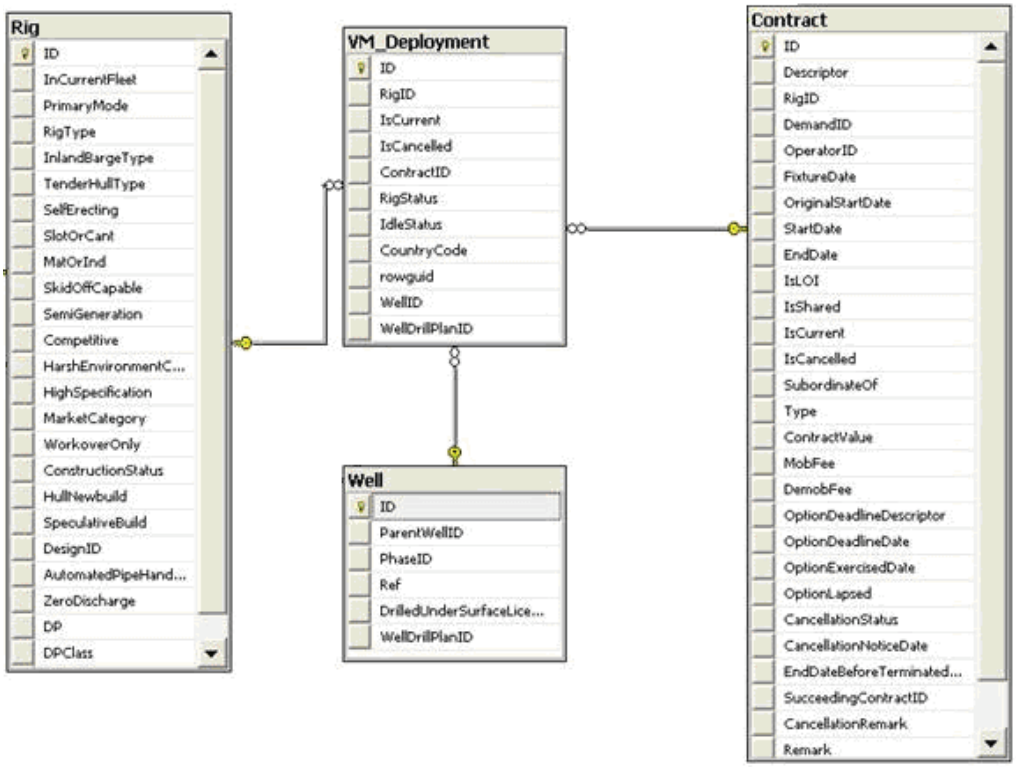


Figure 2: Rig21sWE database extract with the four main tables used for data selection and the link between Well and Deployment

The data quality is mostly maintained by manual cross-checking across teams and by reviewing the current and historical records every time new data is added. The teams are mostly checking for inconsistencies in the data and improbable sequences of actions. Since 2009, there is also an automated *Data Quality Query* (*DQQ*) tool which allows querying the database, using simple rules, in order to detect illogical data. For example, the tool will return an alert when a rig is recorded as contracted at different locations in the world at the same time or when drilling beyond the possible physical depths. This data is then handled by a human operator to correct data coherence.

[5] Deployment and Well activities are defined as such by ODS-Petrodata: A Contract can contain one or more Deployments. A Deployment is an action performed by the oil drilling rig such as "moving to/from location", "undertaking maintenance" or "drilling on location". A Well activity is a detailed action taken on a specific well when in a "drilling" deployment. This includes drilling a main well or a sidetrack, re-entering a well to drill further, etc. There can be none, one or many well activities per deployment.

2.4.2 Expert Guided Selection

An overview of the data distribution with 127 columns selected from the most important[6] fields within the main tables in the database was created and their relationship to a *rig* and a *well* was represented. This has been verified by the project's experts, together with details on the data distributions. The discussion was focused on the relevance of the fields as an indicator or influencer of rig performance. The expert defined that one of the best measures of rig performance is the average performance footage drilled by a rig per day over the length of a deployment (*AveragePerformanceFootagePerDay)*. This field has 25% of data missing in the dataset. Although those additional rows with missing values can be ignored when testing the validity of the model, the additional data provides more information to the learning process of the model. There is no row in the dataset with all of the values present. Each row has at least one of its values missing.

Following discussions with the experts, the fields deemed unrelated to performance or containing insufficient data were removed from the list. This is discussed in section 5.1. Some fields were related to the performance measure but had insufficient data or required complex manipulations to extract (i.e. spread across multiple data tables and databases) and to use (large numbers of values without possible categorisations, for example).

Overall, the data selection was designed to have data coverage of the following categories in the datasets: *financial data, rig availability measure, compliance with regulations, operational efficiency, rig expertise, rig specifications, well information, and environment*.

2.4.3 Data Linking

The dataset for the project contained rare occurrences of matching data from each database as the ODS-Petrodata just started manually linking the data from the two domains (*Rigs* and *Wells*). Originally, the Rigs database contains details of rigs drilling operations whereas the *Wells* database contains details of drilled wells. The linking process matches a historic set of operations data from the *Rigs* database with the data relating to a specific well from the Wells database. There were 221 links for over 20000 potential data points extracted from the databases. In order to have a sufficient amount of data from this dataset, it was necessary to link the *Wells* data and the *Rigs contracts* data. The first step in drawing the linking procedure detailed in Figure 3 was to consult with data experts to obtain a sample of already linked data. I obtained 312 records and systematically mapped the

[6] The measure of importance here was defined to be the relative use of the data field within the commercial products of ODS-Petrodata.

data which matched for each record according to a time sequence based on start and end date of contract and drilling operations. This provided the insights and understanding of the data necessary to devise the algorithm in Figure 3. An example of data used to devise the algorithm is presented in Table 3. The algorithm in Figure 3 has been created to perform an automatic linkage of the data and was implemented in *SQL*. It assigns a rate to each *Rig – Well* potential match sequentially and then selects the *Well* matching the *Rig* with the highest score based on the rate of the matching. When the algorithm cannot differentiate within multiple potential matches, the data is left unmatched.

- For each Rig, when rig is known
- Sort the deployments by date; Sort the wells by date
- Sequentially rate the deployment matching level to the well, by creating a rate:

Deployment start date = Well Spud Date & Deployment end date = Well Completion Date at +/- n Days	SE(30)
Deployment start date = Well Spud Date at +/- n Days	S(20)
Deployment end date = Well Completion Date at +/- n Days	E(10)
Well Spud Date >= Deployment Start Date & Well Completion Date <= Deployment End Date	I(10)
Well Spud Date <= Deployment Start Date & Well Completion Date >= Deployment End Date	W(10)
Well Spud Date >= Deployment Start Date & Well Spud Date <= Deployment End Date OR Well Completion Date >= Deployment Start Date AND Well Completion Date <= Deployment End Date	O(10)
Well Operator = Deployment Operator OR Well Operator = Deployment "DrillFor" Organisation	+1
Deployment Block = Well Block	+1
Deployment SubNationalRegion Code = Well SubNationalRegion Code	

(S= Start, U = End, I = Well within deployment Dates, W = Deployment within Well Dates, O = There is some Overlap in the dates) The scores next to the letters define the priorities for matching (SE>S>E=I=W=O)

- Rank the match in order and pick the best score for each match (when there is a clear best). This algorithm leaves some matching with multiple candidates but no systematic automatic differentiation could be identified which could guarantee the quality of the data.

Figure 3: Data Linking Algorithm

The result from applying this linking algorithm to the entire database is an additional 8358 links created out of 11348 unmatched rig deployment records and 35815 wells records for 482 rigs operating or having operated in the Gulf of Mexico. Added with the existing matched records, this allowed the model learning to obtain 12998 useable data points.

This linking method has been reviewed by the data experts and a significant sample of the links has been manually checked by them. The final matching covered the entirety of the few known examples accurately. This method has been validated as accurate enough on unknown data as well to use the data links generated directly in ODS-Petrodata production environment. The cases found where more than one match is possible are discarded. In a future iteration and for production purpose, a manual check of the data might be possible by a trained data expert as the number of

data rows remaining to match is limited (634 data rows with matching conflict remaining at the time of writing). From these new links, it might be possible over time to learn additional rules and to refine the algorithm.

Table 3: Well-Deployment ↔ Rig record matching example

Well

	SpudDate	TotalDepthDate	CompletionDate	SNR	Block	WaterDepth	LocType	WLType	Phase	WellID	ParentWellID
w1	09/01/2001	31/01/2001	26/02/2001	EC	263	167	Offshore	Surface	New Well	100104987	
w2	04/03/2001	05/07/2001	15/07/2001	WC	296	52	Offshore	Surface	New Well	100147284	
w3	18/07/2001	04/08/2001	23/08/2001	BA	417	91	Offshore	Surface	New Well	100102438	
w4	24/08/2001	28/08/2001	11/09/2001	BA	417		Offshore	Kick-off	Sidetrack	100102439	100102438
w5	13/09/2001	22/09/2001	04/10/2001	BA	417	97	Offshore	Surface	New Well	100102440	
w6	08/10/2001	16/10/2001		EI	300	204	Offshore	Surface	New Well	100110031	
w7	18/12/2001	17/01/2002	03/02/2002	SS	191	70	Offshore	Surface	New Well	100135702	
w8	05/02/2002	26/02/2002	11/03/2002	SS	184	60	Offshore	Surface	New Well	100135643	
w9	30/03/2002	21/04/2002	30/04/2002	ST	187	153	Offshore	Surface	New Well	100140452	
w10	08/05/2002	14/05/2002	04/06/2002	ST	248	178	Offshore	Surface	New Well	100140805	

Rig

	DStartDate	DEndDate	RigStatus	SNR	Block	ContractDescriptor
d1	06/01/2001	26/02/2001	Drilling	EC	263	1 well
d2	26/02/2001	15/07/2001	Drilling	WC	296	6 months
d3	15/07/2001	23/08/2001	Drilling	BA	417	1 well + options
d4	23/08/2001	11/09/2001	Drilling	BA	417	1 well + options
d5	11/09/2001	04/10/2001	Drilling	BA	417	1 well + options
d6	04/10/2001	18/11/2001	Drilling	EI	300	1 well + options
d7	16/12/2001	03/02/2002	Drilling	SS	191	2 wells
d8	03/02/2002	01/04/2002	Drilling	SS	184	2 wells
d9	01/04/2002	30/04/2002	Drilling	SS	187	2 wells
d10	30/04/2002	04/06/2002	Drilling	ST	248	1 well

Chapter Summary: This second chapter introduced the commercial background of offshore oil drilling rigs. It reviewed the basics of the offshore drilling background and the rig tendering process used by the industry to select oil drilling rigs. The problems at hand are exposed by reviewing a list of scenarios provided by industry experts in order to guide the progress of this research. The chapter then provided a review of the *Gulf of Mexico* dataset including the available data, the data selection and the work done to prepare the data.

Chapter 3: Data Modelling: Technology Review

Many approaches exist to data modelling such as stochastic modelling [18], *knowledge discovery* [19] or *Bayesian networks* [20,21]. This research focuses on *Bayesian network* modelling as a starting point to this investigation. This choice has been made because of the capacity of *Bayesian networks* to model knowledge under uncertainty. This is due to the fact that the probability theory on which *Bayesian networks* are based provides the framework for reasoning under uncertainty [22], [23]. As Kjærulff and Madsen mention in [23], "*Bayesian networks [...] are ideally suited knowledge representations for use in many situations involving reasoning and decision making under uncertainty. These models are often characterized as normative expert systems as they provide model-based domain descriptions, where the model is reflecting properties of the problem domain (rather than the domain expert) and probability calculus is used as the calculus for uncertainty*". This suggests that these tools might be ideally suited as the incomplete nature of the data used creates uncertainty. I used various tested and proven modelling algorithms [24–26], mostly based on evolutionary computation. In [27], Kordon mentions that *evolutionary computation* is a key method of *computational intelligence* to use for the problem of forecasting. In recent years, various approaches have been tried to induce *Bayesian networks* from data [21,28–32]. There are examples for the use of *expert knowledge* in order to improve models [33–35]. One example uses *evolutionary algorithms* in the process [36].

This chapter exposes the *Bayesian networks* data modelling technology I am using in my research (section 3.1). Next part (section 3.2) surveys one of the techniques for *Bayesian network* learning using a *search and score* approach, *evolutionary computation* (section 3.3). This is followed by a review of the nature-inspired meta-heuristic, *genetic algorithm* (section 3.4). Then, one of the base methods for modelling, *logistic regression* is examined (section 3.5). *Logistic regression* is used here as a comparison to benchmark the results obtained with *Bayesian networks*. Finally, I review the metrics of quality that are used in this modelling exercise (section 3.6).

3.1 Bayesian Networks

Bayesian networks are probabilistic models based on *Bayesian inference* [37] which is a method to apply the *Bayesian theorem* to update probability estimates. They are useful for representing knowledge under uncertainty. They can be represented using a *directed acyclic graph* associated with a joint probability distribution [24]. Jensen explains that "a *Bayesian network* is a compact representation of the joint probability table over its universe" [38]. *Bayesian networks* assume that

"each variable is conditionally independent of all its non-descendants in the graph, given the values of all its parents" [39]. Each node of the graph represents a random variable X_i related to a problem domain. Each variable has a finite set of mutually exclusive states. Conditional dependencies between variables are represented by edges in the graph and the joint probability distribution can be factorised according to these conditional dependencies. Formally, the joint probability distribution $P(X)$ over the set of random variables $X_1, \dots, X_n$, given $Pa(X_i)$ as the set of parent nodes for node X_i, is represented by:

Equation 1: Application of the Bayesian theorem to Bayesian networks

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i))$$

To make use of the power of *Bayesian networks* in knowledge representation and inference, the network has to be constructed for the given problem. The underlying *directed acyclic graph* structure representing the network has to be learned and then the conditional probabilities calculated. Learning the underlying structure is a hard problem [25] because the number of possible structures grows super-exponentially with the number of variables [40]. One widely used approach to this problem is *search and score*. A metaheuristic is used to search a space representing possible networks. Each solution is scored according to how well it represents the observed distribution of the data. Various authors have presented metaheuristic approaches to this task, including *genetic programming* [41] and *genetic algorithm* [42], [43], [44], [45]. Other approaches in the literature include *hill-climbing* methods [46] and *simulated annealing* [47], [48].

Figure 4 is an example of a simple *Bayesian network* illustrating the likelihood of the state 'Grass wet' (G) occurring, given the events '*Sprinkler*' (S) and '*Rain*' (R). Both event S and R can cause the grass to be wet ($G = true$) and the rain usually has an impact on the use of the sprinklers. The *directed acyclic graph* details those relationships. One way to use this network, using the *Bayes theorem*, would be to forecast the probability (P) that the grass is wet knowing the status of the events S and R. Such probability would then be $P(G, S, R) = P(G|S, R)P(S|R)P(R)$. Other probability propagations can be done to forecast the state of parent nodes given the state of a child node. There are other version of this problem involving other variables, such as the one in [49]. In this work, I am using the same modelling technique (a network of probabilities) to model the probability of a specific rig performance when provided with information on the various events surrounding the rig. This allows the model to provide a reasonable expectation of the likelihood of

a performance even when some of the variables are not informed. The variable's values will still be partially informed to a certain degree of certainty using the network of probabilities.

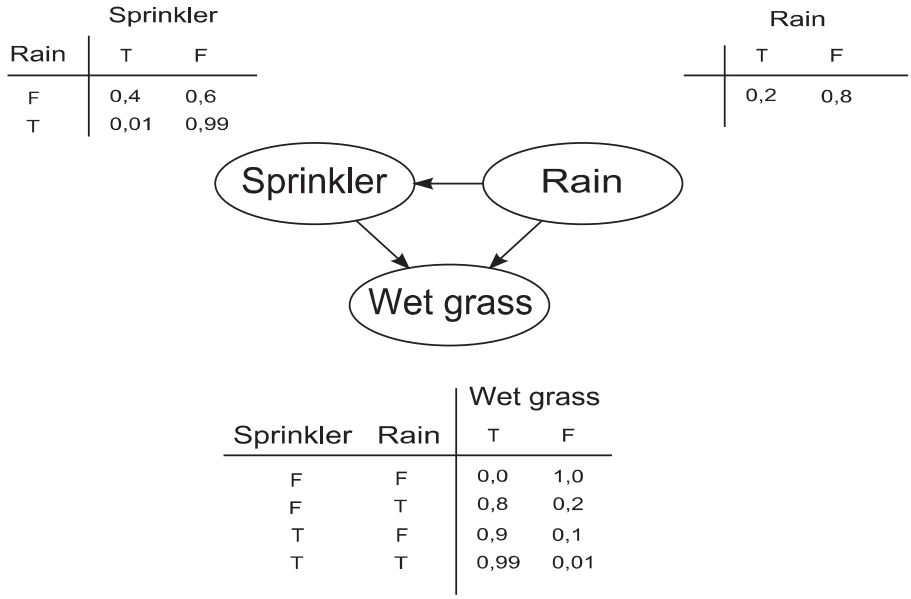


Figure 4: An example of Bayesian Network [50]

3.1.1 Other Applications of Bayesian Networks

Various standard data sets are available for empirical testing of *Bayesian network* structure discovery algorithms. For example, those are issued from domains such as *medical diagnosis*, *car diagnosis* [24], *intensive care patient alarm monitoring* [24], *interplanetary probe raw data interpretation* [37], *search heuristic for problem solving* [37], *virtual office assistant* [37], *automatic context detection* [37], *waste water treatment diagnosis* [51], *knowledge assessment* [52], *smart phone mobile application usage* [53], *microbial risk assessment* [54] and *assessment of debris flow hazards* [55]. Within the oil and gas industry, various attempts to use *Bayesian networks* have been made over the years, in other domains. Some examples are *petro-physical decision support* [56], *safety instrumentation and risk reduction* [57], *prospect analysis in the North Sea* [58], *reservoir uncertainty quantification* [59] and *oil wells productive zones determination* [60]. Those applications of *Bayesian networks* have common features with the problem at hand such as the interconnections of variables, the number of variables or the size of the dataset.

The use of *Bayesian networks* is made through a process called inference. This is the calculation of $P(X|Y)$ for some variables or sets of variables X and Y [39]. The exact probabilistic inference process in *Bayesian networks* is *NP-hard* [61] but in the case of a limited number of parents (*trees*, *forest* or *polytree* graphs, for example), this is still tractable [39].

3.1.2 Limitations of Bayesian Networks

Bayesian networks have a remarkable power to address inferential processes but suffer some limitations. Their operation is based on prior distribution of knowledge. A new request from an unforeseen situation will provide no good results to the end user. However, techniques exist to update *models* to incorporate new data.

Another issue, as highlighted earlier, is the computational difficulty of discovering a new network. This process is an *NP-hard* task which is extremely costly and often impossible to perform, considering the number and combination of variables.

Finally, [37] defines another limitation from the prior beliefs used in the Bayesian inference processing. "A *Bayesian network* is only as useful as this prior knowledge is reliable" [37]. The quality of the network is heavily based on the quality of the data. "An excessively optimistic or pessimistic expectation of the quality of these prior beliefs will distort the entire network and invalidate the results." In the case of my problem, I have separated the dataset into two parts. This allowed me to test the model against over-fitting the data used for creating the model on the one hand and to test the reliability of my model on the other. Those issues are also explored by Duespohl et al. [62].

3.2 Search and Score, Learning Bayesian Networks Using K2

To be used in knowledge representation and inference, *Bayesian networks* have to be constructed for the specific problem. Indeed, “To fully specify a *Bayesian network*, one has to first define the underlying *directed acyclic graph* structure representing the network and then the *Bayesian network*'s probability distribution" [24]. Finding the underlying graph is a *non-deterministic polynomial-time hard* (*NP-hard)* problem [63] because the number of structures grows super-exponentially, given the number of variables in that problem [40]. Evaluating all possible structures is infeasible in most domains. Finding cheaper approaches to learning the structure of *Bayesian networks* from data is an area of research that has gained in momentum in recent years and is now widely practiced and investigated [26]. The algorithms typically used to perform that task can be grouped into two main approaches:

- conditional independence tests methods,
- search and score metrics methods.

The conditional independence approach estimates from the data if there is any conditional dependence between the variables. This is usually measured with the standard statistical or information theoretical tests such as *Pearson's product moment correlation coefficient*, mutual information or the *Chi-Squared* test [50,64,65].

The search and score approach searches for the best structure in the space of possible structures [26]. This approach uses a scoring function measuring the fitness of each structure for the data. The structure found to have the best score is then returned [28]. Algorithms of this type may also require node ordering, in which a parent node precedes a child node so as to narrow the search space [28].

An approach to this problem is to approximate networks using heuristics, such as *K2* [66], [64]. The *K2* algorithm was proposed by Cooper and Herskovitz [21], [28]. *K2* assumes that a priori all structures are equally likely and that cases in the data occur independently and are complete. Moreover, it assumes the presence of a node ordering and imposes a maximum number of parents (inbound edges) for each node. When these conditions are satisfied, *K2* starts with an empty ancestor set for each node and incrementally adds links that maximize a score (*CH-score*, for example) of the resulting structure. The algorithm stops when no more ancestor node additions improve the score. In [11], I observe that although widely used, K2 is prone to local optima and may not find the globally best structure. Also, it relies on prior knowledge of the node ordering and so may return non-equivalent structures given different orderings.

One strategy to find optimal orderings is to use meta-heuristics to search the space of orderings. A range of heuristic search techniques have been used to solve this problem such as: *hill climbing* [46], [67], [68], *genetic algorithms* [42–45,69], [70], [71,72], *genetic programming* [41], *simulated annealing* [47], [48], [73], *Tabu search* [74], *ant colony optimization* (ACO) [75], [76] and *particle swarm optimisation* (PSO) algorithms [77], [78], [79], [80].

A variation of the *genetic algorithm* approach is proposed in [81] and explored in more depth in [24] and [25]. In [24], Kabli proposes to "search the space of node orderings rather than the full space of structures". Using the node ordering as an input, the work described in [81] uses the existing well-performing *greedy search* algorithm K2. In [24] and [25], Kabli investigates the replacement of a full *K2* search by fixed chain structures to evaluate orderings.

3.3 Evolutionary Computation for Bayesian Network Learning

As illustrated above and in [21], [24,25,30,31], learning *Bayesian networks* is an *NP-hard* [63] task and the widely used *K2* algorithm can be very expensive to run. Alterations to the *K2* algorithm as well as various different algorithms have been tried over time. Alternative algorithms often heavily rely on *evolutionary computation* [82].

A first approach, illustrated in *ChainGA* [24] is to use a different scoring function on the search space of nodes ordering to find a solution and then reconstruct the *Bayesian network* using *K2* (deterministic greedy search). Other approaches are making use of *ant colony optimisation* [83] or *particle swarm optimisation* [77] in order to learn the *Bayesian networks* structure [84], [85].

In this work, I applied approaches of *genetic algorithms* and *ant colony optimisation*, associated with a *K2* learning algorithm to the problem at hand. Other *evolutionary algorithms* can be used with K2 and are referenced in the literature. Some of them are: *evolutionary programming* [41], e*volutionary algorithms* [86], [87] (*steady state*, *hybrid steady state*, *elitist*, *hybrid elitist* [43,72]) and *estimation of distribution algorithm* [88,89].

3.4 Nature-inspired Metaheuristic Algorithms

Metaheuristic is a computational method whose aim is to optimise a problem using iterations in order to improve solutions towards a near-optimal solution, when provided by a measure of quality [90]. Modern metaheuristic algorithms often take their inspiration from nature; *genetic algorithm* and *ant colony optimisation* are such algorithms that I tried on my problems. This section exposes the basics and origins of these algorithms. One of the early traces of this concept can be found in Robbins and Monro's work on stochastic optimisation methods [91] as well as in Fermi and Metropolis's work [92] on *pattern search* [93] in 1952. The first evolution process was carried out by Barricelli in 1954 [94]. It is in 1986 that Glover first mentioned the term *metaheuristic* [74]. Those algorithms are also part of the family of algorithms called *stochastic optimization algorithms* [90], [95].

3.4.1 Genetic Algorithms

In [96], Holland proposed the concept of *genetic algorithm*, which was then further developed in Goldberg's book [97]. This search metaheuristic is an *evolutionary algorithm* that mimics the process of natural evolution.

A *genetic algorithm* assumes a *population* of *individuals* (also called *candidate solutions*) which are encoded by a *genome* (also called *chromosomes*). These *individuals* are the solutions to an optimisation problem. Solutions can be represented in binary as strings of 0s and 1s, integers or orderings [98]. A *genetic algorithm* usually consists of 4 parts:

- **Initialisation:**
  Usually, the initial population is generated at random. The number of individuals in the population depends on the problem and the cost of the evaluation. Then, the steps from *selection*, *reproduction*, *crossover* and *mutation* are repeated a number of times in order to create generations that evolve until the *termination* condition is satisfied.
- **Selection:**
  For each generation, a portion of the population is selected and bred in order to create a new generation. The selection is fitness-based and measured by a fitness function, as dictated by the problem. The fitter solutions are more likely to be selected but are not selected by default in order to maintain better population diversity. It is possible here to rate the fitness of the entire population or to only rate a sample of the population.
- **Reproduction:**
  The purpose of this step is to create the next generation of individuals. This is usually done by using two operators on the genome of selected individuals.

  Crossover: In this step, two or more *parents* are selected to create a *child* individual (also called *offspring*). This is done by merging the genetic material from the parents using various means selected, depending on the problem and the previously chosen representation. Larrañaga et al. provide more information on *crossover* operators in [43].

  Mutation: This operator intends to maintain a genetic diversity in between generations. The mutation potentially alters one or more gene values in one individual's chromosome. The probability of this change happening is set as a parameter to the algorithm. It is usually set low in order to maintain the effects of the evolution. A high probability of mutation would equate to a random search. Larrañaga and Kuijpers [98], as well as Eberhart and Shi [99], provide information on *mutation* operators.

  Alternative operators, such as *regrouping*, *colonization-extinction*, or *migration* operators, are exposed by Akbari and Ziarati [100].

- **Termination:**

  When a set criterion is attained, the evolution stops and the best candidate solution is returned as a result. The set criterion can be either, or a combination of:

  - a solution is found that reaches a minimum set fitness,
  - a fixed number of generations is reached,
  - a set time has expired,
  - the set of best solutions is not improving anymore,
  - in some rare cases, a manual inspection (especially in the case of human-in-the-loop algorithms [101], [102]).

3.4.2 Ant Colony Optimisation

In 1992, Dorigo proposed the *ant colony optimisation* algorithm [103], from his 1991 collaboration with Colorni and Maniezzo [104]. The algorithm is based on the real life *ant* behaviours. Initially, *ants* wander randomly and when they find food, they lay down a *trail* of *pheromones* marking the return path. When other ants find that *trail*, they will probably follow the path instead of travelling at random as there is likely to be *food* at the destination. They will then repeat the behaviour. As time goes, the *trail* evaporates and the *ants* are less likely to follow it. The longer it takes for an ant to travel a given *trail* or part of the *trail*, the more the *pheromones* evaporate. Over time, the *pheromone density* becomes more important on shorter *trails*. The evaporation system also avoids that all *ants* follow the same path and do not explore, which would lead to a premature convergence. The communication mechanism used between the ants is called *stigmergy* [105], [106]. It is a mechanism of indirect coordination between agents or actions as a form of self-organization. It allows for simpler agents and decreases the need for direct communication. In the case of *ants*, communication between the agents (*ants*) is done using the map, by depositing the *pheromones* trails. In summary, the algorithm is based on the colony progressing through different states of the problem with each *ant* incrementally contributing to the solution.

The *ant colony optimisation* algorithm typically repeats those 3 steps until the termination conditions are satisfied (plus one step for the initialisation prior to start the loop) [107]:

1. **Ant solutions construction:**

   Each ant incrementally builds a solution by applying a state transition rule. The ant k moves from the state i to state j, where τ_{ij} is the amount of pheromone deposited on the arc between the states i and j, η_{ij} is the weighting function that represents the heuristic information. The transitional probability P is as in Equation 2.

Equation 2: An example of Ant Colony Optimisation state transition rule [107]

$$P_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{\ell} \tau_{i\ell}^{\alpha} \cdot \eta_{i\ell}^{\beta}}$$

$\alpha \in [0,1]$ and $\beta \in [0,1]$ are parameters to control respectively the influences of the *pheromone trails* and from the heuristics. Both j and ℓ are positions not yet visited by ant k. $\sum_{\ell}$ being the sum of the expression with all ℓ not yet visited by ant k. The same algorithm parameters as those used on benchmark problems by Wu et al. [26] have been used in this work.

2. **Local search:**
   This optional step allows for activities which are not possible with a single ant. For example, some usage performs a local search whose result can be used in the next step [107].
3. **Pheromones update:**
   This step is designed to increase the *pheromone* values for good or promising solutions as well as to decrease the *pheromone* values for bad solutions. All of the pheromones in the map are usually decreased by a *phenomenon* of virtual *pheromone evaporation*. This system avoids a premature convergence of the ants. Then the *pheromone* levels from the chosen set of good solutions are updated as in Equation 3 where $\rho \in [0,1]$ is the evaporation rate and Δ_{ij} is the fitness function.

Equation 3: An example of Ant Colony Optimisation pheromone updating rule [107]

$$\boldsymbol{\tau_{ij} \leftarrow (1-\rho)\,\tau_{ij} + \rho\,\Delta_{ij}}$$

This is a simplified depiction of the inner workings of *ant colony optimisation meta-heuristic*. As mentioned in [107], multiple variations exist depending on the problem at hand.

3.5 Logistic Regression

Logistic regression is a "widely used and accepted" method [108] that I am using here to benchmark the models obtained from this research. "The *logistic function* was invented in the 19th century for the description of the growth of populations and the course of autocatalytic chemical reactions" [109]. It is supposed that the origin of *logistic regression* comes from the suggestion of Verhulst between 1838 and 1847 [110], [111], [112]. One was published in *Correspondance Mathématique et Physique*, edited by Quetelet in 1838 [110], [113].

Logistic regression is used for prediction of the probability of occurrence of a value by fitting data to a *logistic* curve. It uses predictor variables that may be numerical or categorical. For my experiments, I am using *multinomial logistic regression* with a ridge estimator [114], based on Cessie and Houwelingen's work [115], and the Weka's implementation by Hall et al. [116]. Prior to using the algorithm, the missing values are replaced using a *ReplaceMissingValuesFilter* [117] which replaces all missing values with the modes and means from the training data. Moreover, the algorithm uses a *NominalToBinaryFilter* in order to transform the nominal attributes (variables with categories) into numeric attributes, as required by the *logistic regression*.

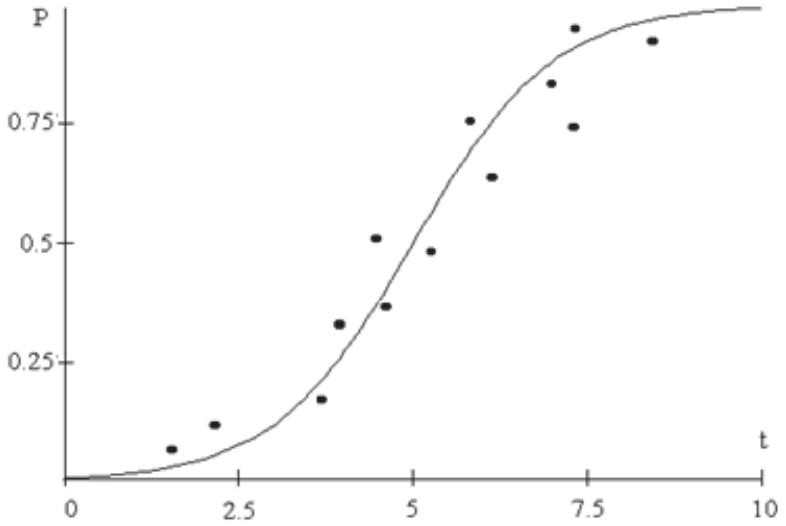


Figure 5: An example of data points with a standard Logistic function fitted to them

The *logistic* curve is a *sigmoid* curve as illustrated in Figure 5.

In the course of this research, various *data mining* algorithms were tried as proposed in the *data mining* platform Weka. *Logistic regression* is used as a benchmark measure of what is possible with regard to the forecasting accuracy because of its good success rate on the problem at hand, which allowed a comparison with the novel application of *Bayesian networks* presented in this work.

3.6 Metrics of Quality

Across this research multiple metrics of quality were used. In the first phase of the research, I wanted to evaluate the complexity associated with learning the model. (section 3.6.1). In the second phase of this research, I focused on evaluating the predicting power of the models built (section 3.6.2).

3.6.1 Building the Models

During the initial phase of this research, the aim was to verify the ability of the available computational resources to compute the models. Two measures were used to estimate the cost of the calculations performed: *model learning time* and *number of evaluations made by the scoring function*. The two are inter-dependent but the model learning time is easier to conceptualise for a human. In addition, the model learning time measure is accurate only when the entire test system is a controlled environment[7].

3.6.2 Models Evaluation

I am using model evaluations for two purposes throughout this research: scoring the *Bayesian network* in order to determine its suitability and measuring the performance of the final models on new data as an estimation of the models' ability to forecast accurately.

a) Bayesian Network Model Scoring

There are many metrics available [118] such as *Cooper-Herskovits* (*CH)* [28], *Bayesian Dirichlet* (BD), *BDe*, *BDeu* [119], *Minimum Description Length* (MDL) [31], *entropy*, *Log-Likelihood* (LL), *Akaike Information Criterion* (AIC), *Normalized Minimum Likelihood* (NML) and *Mutual Information Tests* (MIT). As is recommended in [118], I am using the *CH-score* in this research. This scoring mechanism is defined as part of *K2* but can potentially be replaced within the algorithm by other metrics of quality. One of the advantages provided by the *CH-score* is that it allows the scoring of both learned structures such as the ones learned by *K2* and fixed structures such and *Chain* or *Pyramid* which are explored later in this work.

The *CH-score* (Equation 4) captures the probability of a candidate network structure Bs given a set of data D. Formally, the discrete probability $P(Bs, D)$ is given by:

Equation 4: CH-Score [28]

$$P(B_s, D) = P(B_s) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

[7] For example, it requires the same computer configuration and the same background processor load.

Here q_i denotes the number of possible different instances the parent of variable X_i can take. r_i is the number of values X_i has, N_{ijk} denotes the number of cases in the dataset D in which X_i takes value k of its x_i instance, when its parent Pa_i has its jth value. N_{ij} is the sum of all N_{ijk} for all values x_i can take.

b) Forecast Accuracy

There are many methods to measure the forecast accuracy of a classifier model. For example, Weka [116] proposes: *percentage of correctly classified instances*, *Kappa statistic*, *Mean absolute error*, *Root mean squared error*, *Relative absolute error* and *Root relative squared error*. For this work, I chose to use the percentage of correctly classified instances as the measure of accuracy as it is the simplest measure and allows the ability to present that measure of accuracy to the users of the forecasting model.

There are three main techniques for testing a model: using the training data set, using a separate testing set or a split from the testing set or using a cross-validation technique. Using the training set, i.e. the same data used to create the model, is an easy solution but does not allow detecting if the model is over-fitting the data. If the model over-fits the data, its performance will be reduced when new occurrences, which were not in the initial dataset, are presented to the forecasting model. Using a separate testing set is a better solution if enough data is available as it tests the model for new occurrences that were not in the training dataset. The main drawback when the amount of available data is limited is that this method limits the amount of data available for the algorithm to build the model.

For this research, I choose to use cross-validation. This is one of the most common methods [120]. *Cross-validation* is a technique which allows assessing how the results of a forecasting model are generalised to an independent dataset. Here, I used a 10-fold *cross-validation*. In essence, the dataset is split in 10 parts, 9 are used as training data and 1 is used as testing. This is repeated 10 times total so that all data is used as training and testing an equal number of times.

c) Concordance Index (C-Index)

"The *concordance index* [...] quantifies the quality of rankings" [121], [122]. The index is the probability of concordance between the predicted and the real value.

Assuming that the predicted variable has values such as $c_1 < c_2 < \cdots < c_n$. Considering each pair of example e_1 and e_2 with their value v_1 and v_2, where v_1 and v_2 are different, Equation 5

expresses the probability $P(v_1 < v_2)$ that the value v_1 of the predicted variable for e_1 is less than the value v_2 of the predicted variable for e_2 with the corresponding probability $P(v_2 < v_1)$.

Equation 5: Probabilities for Concordance Index

$$\boldsymbol{P(v_1 < v_2) = P(v_1 = c_1 \cap v_2 > c_1) + P(v_1 = c_2 \cap v_2 > c_2) + \cdots + P(v_1 = c_{n-1} \cap v_2 = c_n)}$$
$$\boldsymbol{P(v_2 < v_1) = P(v_2 = c_2 \cap v_1 > c_2) + P(v_2 = c_1 \cap v_1 > c_1) + \cdots + P(v_2 = c_{n-1} \cap v_1 = c_n)}$$

The *Concordance Index* CI is then calculated such that for each pair $(e1, e2)$ where the real values $(v_1 < v2)$:

Equation 6: Concordance Index calculation

$$\boldsymbol{CI = \frac{number\ of\ pairs\ where\ (P(v_1 < v_2) > P(v_2 < v_1))}{total\ number\ of\ pairs}}$$

The higher the *concordance index* (close to 1), the better is the forecast capability.

Chapter Summary: This third chapter provided a review of the state-of-the art techniques for data modelling using Bayesian networks. The focus of this work is centred on search and score methods using the *K2* scoring algorithm. Nature inspired and evolutionary algorithms are explored, with a specific focus on genetic algorithms and ant colony optimisation. In order to provide a benchmark, the standard *Logistic* regression algorithm is also approached in this chapter as well as the metrics of quality used to assess the results.

Chapter 4: Evolved Bayesian Network Models of Rig Drilling Operations in the Gulf of Mexico

I investigate the use of *genetic algorithms* and *ant colony optimisation* to induce a *Bayesian network* model for the real world problem of rig operations management. I sample from a new dataset that I name *WRD1* (section 4.1), then use *K2* with *genetic algorithms* (section 4.2) and *ant colony optimisation* algorithms (section 4.3) in order to learn *Bayesian networks* models (section 4.4).

4.1 Dataset - WRD1

I assembled the *WRD1* dataset in order to provide real-world typical data, relevant to the problem at hand, and to develop and test the algorithms selected for this research. I initially published this dataset in [123]. The name *WRD* comes from 3 sources of data assembled: *Wells* – *Rigs* (specifications) – *Deployments*.

Table 4 shows the variables that have been selected for the initial experiments. The data selected is based on:

- **data availability:** the data is covered sufficiently within the available database,
- **relevance:** suggested by their use in ODS-Petrodata's tools and products and by the information categories published in [17],
- **readiness:** need for transformation, discretisation, filtering or cross-referencing to reach a useable state.

I believe the data in *WRD1* to be a minimum representation of the problem at hand. However, I selected the data to represent what I believe are the best indicative elements of performance in the timeframe available.

Overall I selected 17 key fields with sufficient data coverage as well as a reasonable number of distinct values. I maintained the number of fields selected low in order to have a tractable computation load during the model learning process. When available, the fields selected inform the criteria considered by Osmundsen et al. in [17] and [124]. The computational load was, however, still significant as some of the variables have a large number of values. The fields selected were either taken directly from the database fields or derived from them when not directly usable in a

meaningful way[8]. All the numerical fields[9] are discretised in industry-meaningful categories, established using industry expertise, such as rig operating categories or usual operating ranges of particular equipment. Other fields have been left unprocessed and directly copied over to the dataset. The *Wells-Rigs-Deployments* dataset (version 1) – *WRD1* – produced from this extraction contains 6670 rows containing related values of 17 factors.

Table 4: WRD1 selected fields and variable value count for preliminary experiments

Field Name	Number of distinct values
Well Phase	6
Well Deviated	4
Well Type	6
Well Status	7
Well Result	17
Days On Location[10]	11
Number Of Days To Total Depth[10]	10
Total Vertical Depth[10]	18
Total Footage Drilled[10]	18
Average Feet Drilled Per Day[10]	16
Shore Base	54
Region	59
Water Depth[10]	10
Rig Type	6
Harsh Environment Capability	2
Rig Owner	72
Rig Contractor	70

4.2 K2 and Genetic Algorithms

In [43], Larrañaga et al. proposed a *genetic algorithm* to search the space of node orderings rather than the full space of structures. The initial individuals in the population are randomly created node orderings which are then evolved until a good ordering is found. In each generation, a pair of individuals is selected for crossover and mutation. They are selected according to their calculated fitness score within the population. Only one individual offspring is created and scored at a time and, if it is a better performer, it replaces the worst individual in the current population. If it is not better, it is simply dropped. The fitness of each ordering is calculated by running the greedy search algorithm *K2* on that ordering and returning the score of the network structure found. For the purpose of this paper, I denote Larrañaga's algorithm by *K2GA*. Figure 6 illustrates the *K2GA* algorithm. The algorithm starts by generating a random population. It then evaluates each individual by running a *K2* search through the ordering (genome of the individual). It then performs

[8] For example, start and end dates are transformed into durations.
[9] For example, water depth or footage drilled.
[10] Discretised numerical field.

a selection of two individuals and produces two offspring. If the new individual is fitter than the worst individual, this individual is inserted within the population instead of the worst individual. This is repeated until one of the termination conditions is satisfied (maximum number of generation or target fitness).

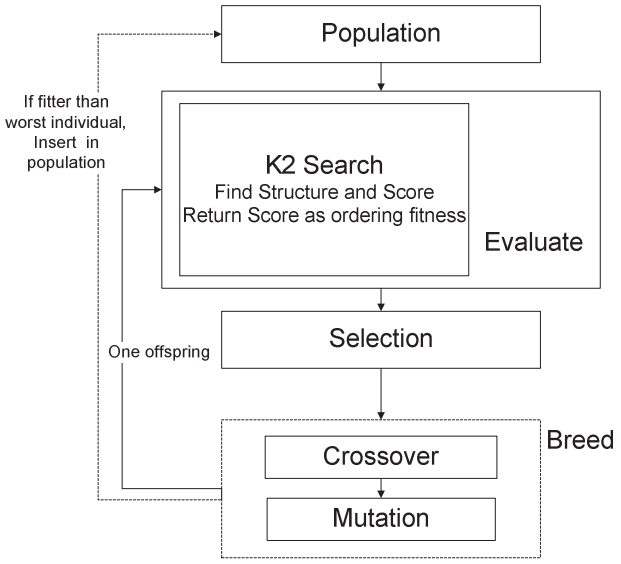


Figure 6: K2GA [43]

In [24], Kabli et al. propose *ChainGA*, an alternative way of reducing the computational cost related to this by using chain structures to evaluate orderings, replacing the *K2* expensive evaluation in *K2GA*. *ChainGA* follows a similar approach to *K2GA*: it searches the space of node orderings and assigns a value to each ordering based on the *CH-score* [28]. However, rather than using *K2* to construct a network on each ordering, *ChainGA* evaluates a fixed chain structure. This low resolution evaluation phase terminates in a set of orderings that have the highest evaluated *CH-score* found with this structure. Figure 7 illustrates the *ChainGA* algorithm. The algorithm starts by generating a random population. It then evaluates each individual by calculating a *CH-score* on the ordering (genome of the individual). It then performs a selection of two individuals and produces two offspring. If the new individual is fitter than the worst individual, this individual is inserted within the population instead of the worst individual. This is repeated until one of the termination conditions is satisfied (maximum number of generation or target fitness). After the evolution has ended, a selection of the best individuals is selected (typically 5 individuals, as recommended by Kabli et al.) and a *K2* search is run with the orderings of those individuals to find the best performing ordering.

In the Gulf of Mexico datasets (*WRD1*), several variables have large value sets, leading to significant computational cost using this approach. This is a cost incurred by all approaches that

use *CH-score* and is not specific to the *K2GA* and *ChainGA* algorithms. In *ChainGA* each variable has at most one parent, whereas in *K2GA* nodes have multiple parents. The fact that in this application parents can have many values means that the savings of *ChainGA* are greater than when a dataset has a limited number of parents, as is usual in standard datasets. For example, *WRD1* has multiple variables with over 50 possible values when other standard datasets from [11] such as *ALARM* has 4 values, *ASIA* has 2 values and *CAR* has also up to 2 values only. Overall, *ChainGA* generally results in a reduced computation time since the number of links to evaluate is fixed and is, in general, much smaller than that required by *K2GA*. In [24], Kabli et al. compared *K2GA* and *ChainGA* on a set of benchmark problems with known networks; trade-offs were observed between computation cost and the quality of the structure found. In the following section, I describe experiments with these algorithms run on the rig operations dataset.

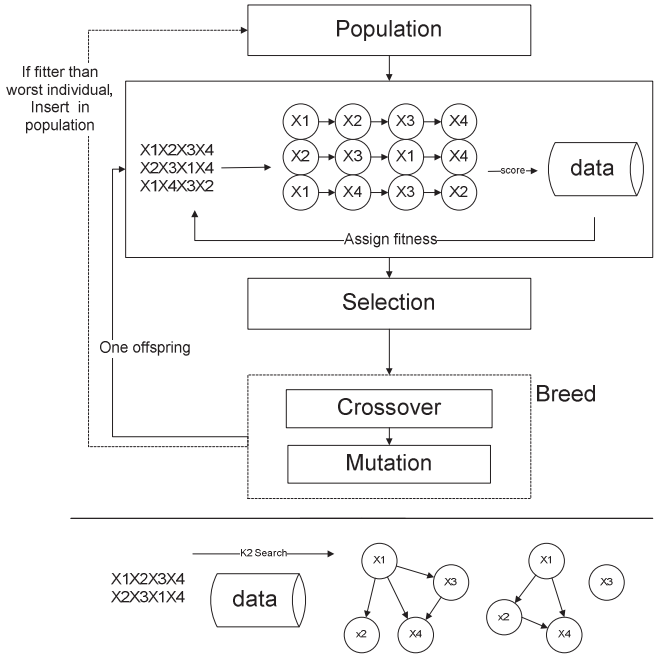


Figure 7: ChainGA [24]

In this research, I used the same algorithms parameters as used by Kabli et al. on benchmark problems. At this point in the research, I decided not to optimise the algorithm parameters to the dataset as I will use the algorithms with other datasets.

4.3 K2 and Ant Colony Optimisation

There are a many published descriptions of *ant colony optimisation (ACO)* algorithms to learn *Bayesian networks* [75], [125], [126], [127], [85]. Typically, these approaches integrate with other greedy construction heuristic algorithms [128], [119], [68]. In this research, two algorithms were used, based on *ant colony optimisation* for *Bayesian network* structure learning, developed by Wu

et al. [26]. Those algorithms are based on two existing approaches: *ChainGA* and *K2GA*. They are named in [26] as *ChainACO* and *K2ACO*. I am using this *ant colony optimisation* approach to find the node ordering for learning an optimal structure. The main idea of the *ChainACO* approach comes from *ChainGA*. *ChainACO* uses the same two phases, depicted in Figure 8. In the first phase of *ChainACO*, the algorithm constructs chains using an *ant colony optimisation* approach instead of *genetic algorithms*. Then, the second phase applies *K2* to the best orderings found and returns the best structure. Figure 8 and Figure 9 illustrate the details of *ChainACO*.

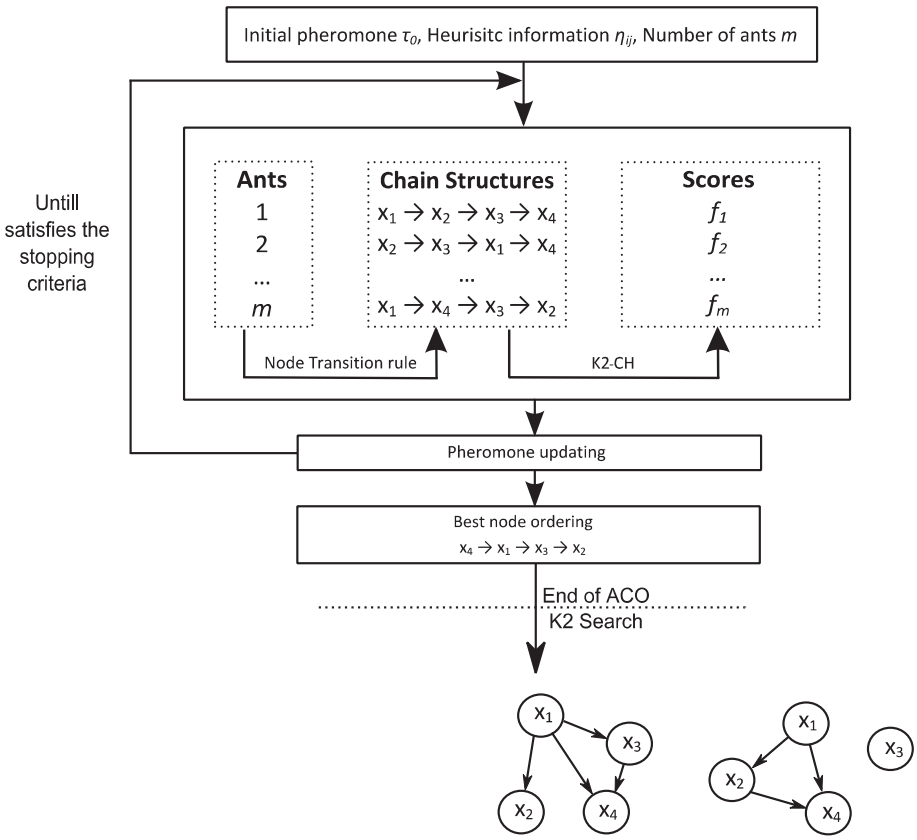


Figure 8: ChainACO [26]

```
ChainACO:
1. Initialise pheromone
    Initialise heuristic information, select the starting nodes.
2. Loop
        Each ant is positioned on a starting node
        Loop
               Each ant applies a state transition rule to incrementally build a
               solution and a local pheromone updating rule
        Until all ants have built a complete solution
    A global pheromone updating rule is applied
    Until termination criterion is met
3. Implement K2 Algorithm on best solution to learn the best structure.
```

Figure 9: ChainACO pseudo-code [26]

In *K2ACO*, the *genetic algorithm* from *K2GA* is similarly replaced by an *ant colony optimisation* search. The initial individuals in the population are the randomly created node orderings, which are then optimized by a colony of *ants* in this space until a good ordering is found. During the *ant colony optimisation* process, the fitness of each ordering is calculated by running the *K2* search algorithm. On completion, the model is learned using the best ordering and *K2*. Figure 10 and Figure 11 illustrate the details of *K2ACO*.

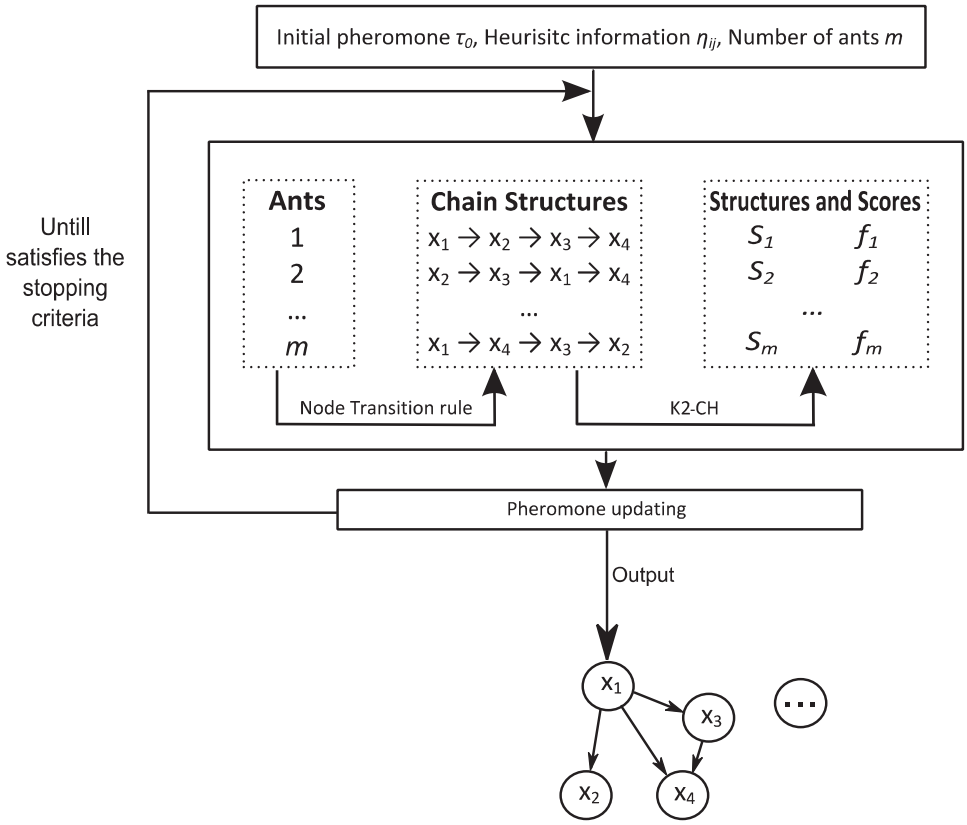


Figure 10: K2ACO [26]

```
ChainACO:
1. Initialise pheromone
    Initialise heuristic information
2. Loop
        Each ant is positioned on a starting node
        Loop
           Each ant applies a state transition rule to incrementally build a
           solution and a local pheromone updating rule
        Until all ants have built a complete solution
    A global pheromone updating rule is applied
    Until termination criterion is met
3. Implement K2 Algorithm on best solution to learn the best structure.
```

Figure 11: K2ACO pseudo-code [26]

4.4 Experimental Results

Following the steps of the *K2GA*, *ChainGA*, *K2ACO* and *ChainACO* algorithms described above, I built the *Bayesian network* model that represents the data I have selected.

The *K2GA* and *ChainGA* algorithm implementations were run 45 times each with 200 generations with a population size of 30 node orderings. *Displacement mutation* and *cycle crossover* rates were 0.05 and 0.9 respectively. The selection used was a tournament selection of size 4. Those values were optimised empirically using test runs with 100 cases randomly selected from the dataset. The best scored resulting network was then chosen as the optimal model for the problem at hand. I ran each algorithm 45 times over the *WRD1* dataset and compared the results using a two-tailed T-test to validate their significance.

In this part, I start by reviewing the performance of each of the algorithms, as measured by the *CH-score*. I assess the structure produced, looking at the variability between algorithms as they are assessed from an industry standpoint. I then review the edges frequencies using node juxtaposition analysis and explain observed differences between the algorithms.

4.4.1 Structures Performances

Figure 12 illustrates the *Bayesian network* models learned from data using the algorithms. In this figure, it is possible to see some matching relationships formed in the models created by *K2GA*, *ChainGA*, *K2ACO* and *ChainACO*.

The mean structure scores for each algorithm are presented in Table 5. Significance tests were carried out on all pairs of means and the results are shown in Table 6. All differences are significant at or beyond a 99.95% confidence level. *K2GA* produces on average significantly better scoring structures than all of the other algorithms on the dataset. The best-ever individual for *K2GA* scored -55534 compared to -60203 for *ChainGA*, -55781 for *K2ACO* and -55976 for *ChainACO* on the relative score scale (log of *CH-score*). Although significantly different, the results from *K2ACO* and *ChainACO* are much closer to *K2GA* than *ChainGA*, and they also benefit from a smaller standard deviation, showing their stability compared to *ChainGA*'s.

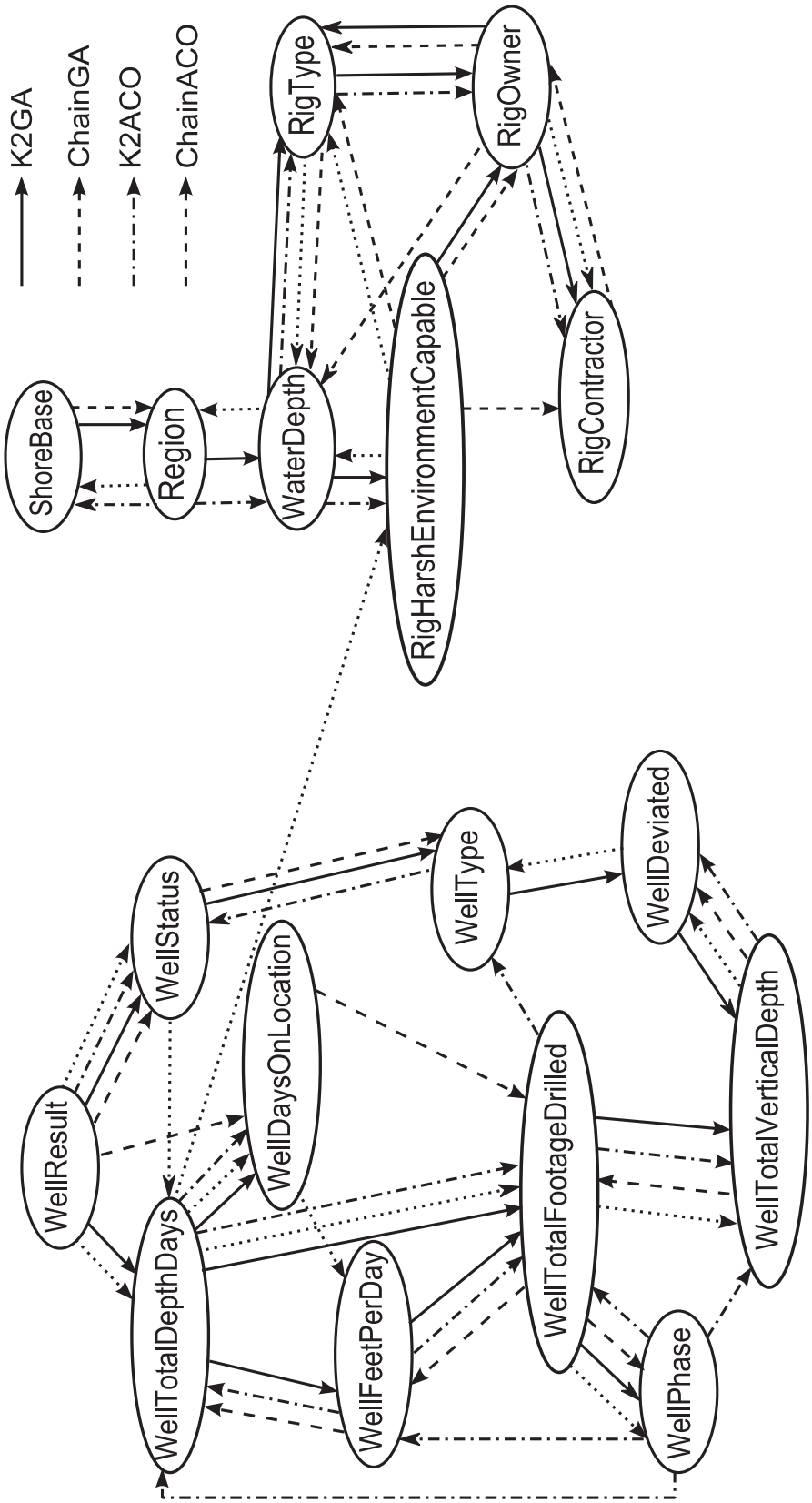


Figure 12: Network representations for K2GA, ChainGA, K2ACO and ChainACO [129]

Table 6 confirms that *K2GA*, *K2ACO* and *ChainACO* are much closer to each other, in terms of scoring, than *ChainGA*. The difference in the mean score of all pairs formed from *K2GA*, *K2ACO* and *ChainACO* is less than 1000, when all pairs involving *ChainGA* have a difference in mean score around 7000. It is to be noted that, as discussed in [24], the performance of *ChainGA* relating to *K2GA* appears to be highly problem-dependent. As confirmed by [26], I expect that the performance of *K2ACO* and *ChainACO* will also be problem-dependent.

Table 5: Means and standard deviations of best individuals K2 scores

	N	Mean Score	Standard Deviation
K2GA	45	-56197.44	205.2
ChainGA	45	-66434.34	1237.7
K2ACO	41	-56265.43	297.8
ChainACO	40	-56556.41	254.7

Table 6: Paired t-test of best individuals K2 score across all runs

Pair	N	Paired Mean Score	Paired Standard Deviation	P
K2GA-ChainGA	43	7721.67	954.36	< 0.0005
K2ACO-ChainACO	40	308.39	109.75	< 0.0005
K2GA-K2ACO	41	410.36	298.73	< 0.0005
ChainGA-ChainACO	40	-6885.66	653.74	< 0.0005
K2GA-ChainACO	40	694.27	234.91	< 0.0005
ChainGA-K2ACO	41	-7220.71	658.14	< 0.0005

Mean runtimes for each algorithm are presented in Table 7. The *ChainGA* runtime is about a quarter of the *K2GA* runtime. *K2ACO* requires a significantly different but closer time to *ChainGA*. However, *ChainACO* completes with runtimes divided by a factor of 10, when compared to *K2ACO* or *ChainGA* and a by a factor of 40, when compared to *K2GA*. There is, therefore, observed trade-offs between quality and computation time.

Table 7: Time statistics per run over all runs

	Average	Standard Deviation
K2GA	42h 28min	5h 9min
ChainGA	11h 1min	1h 11min
K2ACO	11h 50min	0h 41min
ChainACO	1h 39min	0h 5min

The score of the algorithms based on *ant colony optimisation* being much closer to *K2GA* than *ChainGA*, the loss of quality compared to the gain of time is statistically significant, but smaller than the loss of gain obtained by *ChainGA*. The long computation times required on this problem are in a large part due to the number of distinct values taken by many of the variables.

4.4.2 Expert Evaluation of the Model

The best network structures produced by both *K2GA* and *ChainGA* have been presented to Rig and Wells data experts. All the algorithms discovered interactions between *Rig Capabilities*, *Rig Types* and *Water Depth* nodes. The project's experts highlighted that those are linked because specific rig types typically operate at a specific range of water depth. Another group of interactions is identifiable between *Well Result*, *Well Status* and *Well Type*. Only *ChainACO* omitted that link; however, as the search is non-deterministic, another run of *ChainACO* might find it. The *Total Footage Drilled* node also interacts with the node representing the *Drilling Phase* and the one representing the *Footage Drilled per Day*. In addition, there is a strong link between the *Water Depth* and the *Rig Type* nodes. Those will be logically related because of the technical abilities of specific rigs to allow them to work at specified depths. The relationships between the *Rig Type*, the *Rig Owner* and the *Rig Contractor* are justified by the propensity of rig owners and contractors to work together repetitively and to be specialized in specific type of rigs built on the same plans. These specific interactions have consistently been identified by all algorithms. The networks learnt also identify a relationship between the *Shore Base* and the *Region* where the oil drilling rig is operating. This is another logical geographical association showing the abilities of the algorithms to learn valid information and build *Bayesian networks* from data. None of the links uncovered may be novel to data experts but they provide me with a definite specific link that is supported by unbiased data analysis.

The partial separation between *Well*-related and *Rig*-related variables (with the exception of geographical and water depth variables) suggests a potential difficulty in using the model as a predictor for *Rig* variables using *Well* data or for *Well* variables using *Rig* data. However, adding some key variables might solve that problem. *Water Depth*, originating from the *Well* database, has emerged as a key variable that correlates with the rig capabilities and, hence, confirms its position as a significant variable in the choice of a rig. In the Gulf of Mexico, that typically has a uniform geological profile, this may be a reasonable assumption; however, this will have to be explored further and confirmed on worldwide data where a range of geological profiles and water depths will exist. Alternatively, there may be additional variables in the *Wells* and *Rigs* database that do correlate more closely. Furthermore, I would expect geological and other variables to be relevant in more heterogeneous regions but they are usually scarcely available.

4.4.3 Node Juxtaposition Analysis

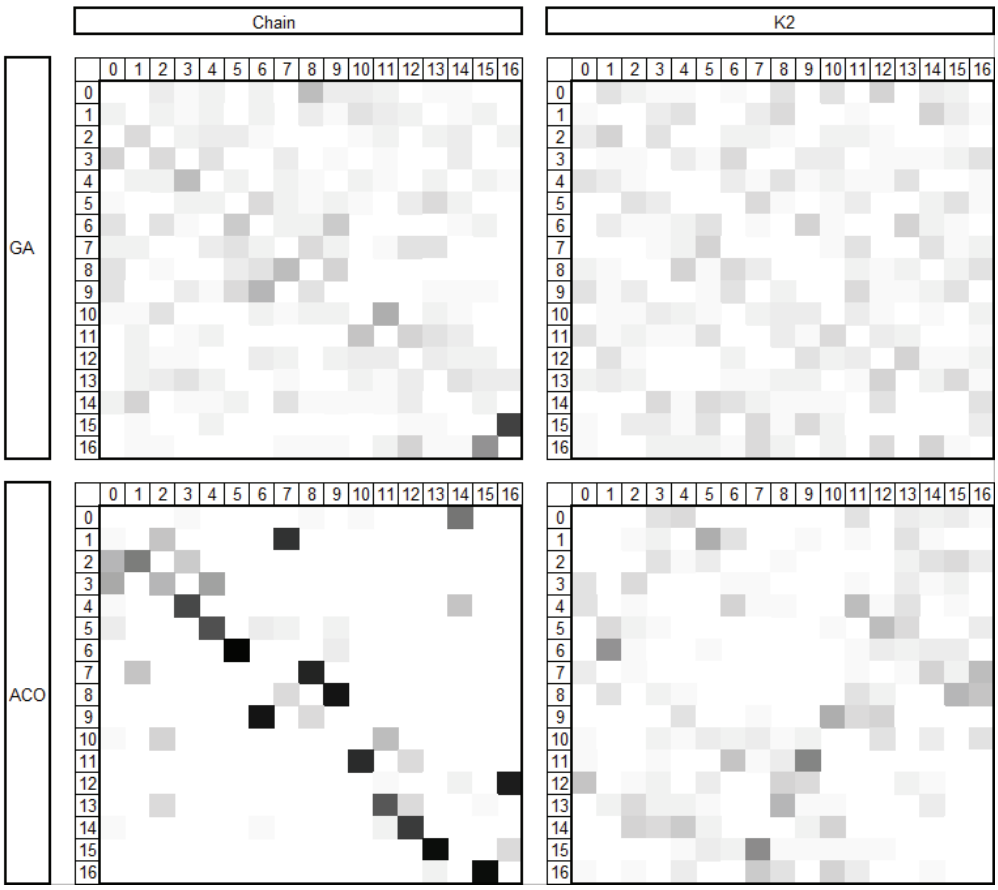


Figure 13: Grayscale representation of node juxtapositions for Genetic Algorithms/Ant Colony Optimisation and K2/Chain algorithms on WRD1

Figure 13 represents the occurrences of node juxtapositions as a greyscale grid. The vertical axis represents the first node; the horizontal axis represents the second node. The shade is darker proportionally to the number of occurrences of links across the graph. The greyscale levels are balanced across the four quadrants of Figure 13.

The precedence of a node in an ordering means its eligibility for being within the parent set of nodes of *Bayesian network* structure. The *Chain*-based algorithms insert a directed edge between each ordered node and its immediate successor, i.e. from node juxtaposition in the ordering. The *K2*-based algorithms, when considering a particular ordered node, will first try inserting an edge from its immediate predecessor and so have a bias in favour of such edges. Therefore, consideration of which of these edges would result from the best orderings found in each run of each algorithm provides statistics which describe the distribution of search outcomes for this problem. Figure 13 shows that *K2GA* explores the search space more broadly, without focusing on

any specific link. This explains why it finds better solutions, but this is an expensive behaviour. *ChainGA* seems to focus the exploration on the most likely chains. However, its score is lower than that of other algorithms. *K2ACO* reduces even further the thoroughness of the search but performs better than ChainGA. The algorithms based on *ant colony optimisation*, on this problem, seem to be more stable than ChainGA, and also focus more quickly onto the most important part of the ordering, compared to *genetic algorithm*-based algorithms. *ChainACO* clearly focuses on some important nodes, converging quickly and consistently towards a good solution. I am observing here the effects of two choices: *K2*/*Chain* and *genetic algorithm*/*ant colony optimisation*.

Given an ordering, *K2* is free to add any parent-child link in its process of constructing a full *Bayesian network* for the purposes of evaluating the ordering. *Chain*, on the other hand, constructs precisely those parent-child links corresponding to nodes immediately juxtaposed in the ordering. Therefore, for *Chain*, the distribution of fitness (in phase 1) will be on those orderings that juxtapose strongly related variables, thus, focusing the search on this restricted set of orderings. The *K2* approach will distribute fitness across a wider set of orderings and so a wider set of variable juxtapositions will still allow variables to be related in the structures *K2* builds. *Genetic algorithms* tends to be a noisier metaheuristic than *ant colony optimisation*. Thus, as expected, *genetic algorithms* have a higher variance than the *ant colony optimisation* and one can observe a wider search distribution as well.

4.4.4 Algorithm Analysis

I applied both *genetic algorithm* and *ant colony optimisation* algorithm based on *K2* and the *Chain* model. This approach provides multiple orderings that the algorithms found when searching for the best networks using *K2*. These orderings are used to study the stability of edges found by the algorithms by counting them out of the best orderings listed for each algorithm at their final stage. The *K2GA* model is the original algorithm and still the best performing when observing the *K2-CH* scores. Its results and performances are, therefore, used as a benchmark for the other algorithms. It appears that *ant colony optimisation* algorithms explore fewer variations of edges than the *genetic algorithms* and converge faster to a good solution. The solutions are significantly less well-performing but the difference between *K2ACO* or *ChainACO* and *K2GA* is much less than the difference between *ChainGA* and *K2GA*. The best improvement brought by *ant colony optimisation* when compared with *genetic algorithms* is the time of execution. *Ant colony optimisation* uses a fraction of the time *genetic algorithms* take to find a plausible network. This is due to *ant colony optimisation* algorithms reducing the number of factor evaluations by converging faster to a good solution.

Chapter Summary: **This fourth chapter introduced the *WRD1* dataset as well as the *K2*-based *genetic* and *ant colony optimisation* algorithms. The experimental results are analysed by reviewing the performances of each algorithms and then by considering expert's evaluation of the model's structures. This chapter also introduces the *node juxtaposition analysis* for viewing the frequency of nodes selected from the search and score approach. For the following parts of this research, even though *ant colony optimisation*-based algorithms performed faster than *genetic algorithm*-based algorithms, *K2GA* was chosen for the further experiments as it provides a higher quality of models and, hence, has a better chance at forecasting oil drilling rig performance accurately.**

Chapter 5: Drilling Performance Models of Rig Operations in the Gulf of Mexico

In this chapter, I investigate the development of a larger model to enable the forecast of the *average footage drilled per day* of an oil drilling rig as a measure of its performance[11]. I start by exposing in detail the assumptions made as well as the work done on the data available to build a more advanced data set that I name *WRD2* (section 5.1). Then I review the forecast abilities of the models in order to support oil drilling rig performance forecast (section 5.2).

5.1 Selection of Data for Model Building – WRD2

For the creation of the second dataset, I consulted more extensively with the project experts from the *Rigs* and *Wells* department in ODS-Petrodata as well as with additional data analysts and reporters. A large number of available fields have been identified. I initially selected 138 fields[12]. Then, the data teams studied the selection in order to identify a number of key fields capable of indicating the performance of an oil drilling rig. Overall, the data selection was designed to have data coverage of the following categories in the datasets: *financial data*, *rig availability measure*, *compliance with regulations*, *operational efficiency*, *rig expertise*, *rig specifications*, *well information*, and *environment*.

Overall, the data selection was designed to have data coverage of the following categories in the datasets: *financial data, rig availability measure, compliance with regulations, operational efficiency, rig expertise, rig specifications, well information, and environment*.

Table 8 presents a summary of the data selected for the creation of the WRD2 datasets. It shows the number of missing data points for a given field and how it relates in percentage to the overall dataset size and the number of distinct values in the field. The number of distinct values should be reasonable for categorical data (about 2 to 15) to keep the problem tractable. In the case of numerical data, the number of distinct values will be high. The four right-most columns of the table are only relevant for numerical data and provide the min, max, mean and standard deviation of the data over the given data field. These measures show the amplitude and key characteristics of the data.

[11] This is only one measure within multiple possible metrics applicable to an oil drilling rig and it is not expected to depict in its entirety the performance of an oil drilling rig.
[12] Some of those are exposed in Table 8 and Figure 15.

Table 8: Data fields selected for WRD2

Data Field	# Missing	% Missing	# Distinct values	Initial type of Data	Min	Max	Mean	Standard Deviation
PreviousWellsPerYear	394	6%	11	Numerical	0	13	1.198	1.392
PreviousWellsSince Upgrade	0	0%	82	Numerical	0	151	2.771	12.47
PreviousWellsInRegion	0	0%	92	Numerical	0	201	26.569	26.535
PreviousWellsInBlock	0	0%	50	Numerical	0	99	6.826	10.193
PreviousWellsInField	0	0%	15	Numerical	0	24	0.416	2.222
AverageUtilisation	854	14%	80	Numerical	0	98	63.055	17.245
AveragePerformance FootagePerDay	1595	25%	3551	Numerical	10.2	5803	515.717	497.921
DaysToTotalDepth	377	6%	162	Numerical	0	390	27.599	26.127
ContractLength	0	0%	711	Numerical	3	6604	351.342	547.333
RigAgeAtTimeOf Drilling	84	1%	47	Numerical	0	56	29.512	7.696
WaterDepthMax	0	0%	115	Numerical	0	10000	1051.781	2148.835
DrillingDepthMax	233	4%	22	Numerical	6500	40000	23642.81	4772.893
RigType	0	0%	11	Categories				
MatOrInd	0	0%	2	Categories				
SlotOrCant	0	0%	2	Categories				
ZeroDischarge	3827	61%[13]	2	Categories				
VariableDeckload Operating	434	7%	212	Numerical	192	28660	9040.393	3318.133
DualActivity	103	2%	2	Categories				
DerrickCapacity	388	6%	44	Numerical	200K	2500K	1254K	309K
DrawworksHP	344	5%	23	Numerical	350	7000	2419.904	854.838
MudPumpNumber	379	6%	4	Numerical	2	5	2.419	0.607
MudPumpHP	155	2%	20	Numerical	310	3000	1613.037	237.098
TopDriveTorque	0	0%	14	Categories				
WellType	0	0%	7	Categories				
WellPhase	0	0%	7	Categories				
WellDeviated	1565	25%	3	Categories				
WellHPHT	2	0%	2	Categories				
WellDaysActive	154	2%	183	Numerical	2	450	42.811	30.874
TotalFootageDrilled	1544	25%	3961	Numerical	11	29519	10564.92	4007.37
TotalMeasuredDepth	464	7%	4052	Numerical	515	33815	11469.09	4250.294
TotalVerticalDepth	1592	25%	3689	Numerical	650	32060	10716.18	4032.1
WaterDepth	1317	21%	929	Numerical	0	9727	552.43	1382.512
HuricaneRisk	0	0%	2	Categories				
WellLocationType	0	0%	2	Categories				

Table 9 lists the data fields that the experts thought could inform the forecasting model but had insufficient data or were complex to populate. Those data are usually spread across multiple data tables and databases or have a large number of distinct values without possible categorisations. For example, this is the case of *RigDesign* that has 335 distinct values without any meaningful categories at that time. In some cases, the data required to populate the information was partially or fully missing. As most cases with those data show, the manipulations would have been too complex to extract meaningful data. For example, the Market category is dependent on the depth

[13] In order to improve coverage, the company data experts suggested that this field's missing values are completed based on the rig building year such as "WHEN NULL If(rig>2005) Y else N".

ratings and on the type of rig to be significant. One approach would be to collate those variables into one but the multiplication of the number of values would then generate too many distinct values for the technology used in this project to handle. It is assumed that the more data are available the better a forecast can be done. By reducing the amount of data used, the accuracy of the models produced is naturally reduced but their generation becomes tractable with the computing power available to this project. I do not think the modelling exercise presented here is invalidated by this manual selection as it provides an easy access to new information on oil drilling rig performance, which was difficultly available but to a few experts before the modelling exercise.

The final assembled dataset contains 12998 data points and covers rigs and drillings from 1983 to 2010. The dataset contains 9528 data points, informing the *AveragePerformanceFootagePerDay* column that I am using in the experiments.

Table 9: Fields not selected and data not available but of potential interest for WRD datasets[14]

Name	Missing data	Complex data
Average Day Rate	X	
Average Price Per Foot Drilled	X	
Market Category		X
Utilisation of Fleet at Time of Contract		X
Utilisation of Fleet for Rig Type at Time of Contract		X
Average Feet Drilled Per Non-(idle/WOW) Day	X	
% Days Waiting on Weather (WOW) for Rig over Period of Time	X	
% Days Waiting on Weather (WOW) in the Region the Well is Operating at the Season the Well is Operating	X	
Design Company		Too many un-groupable distinct categories
Rig Design		
Operator		
Recent Management Change	X	
Time Since Last Well		X
Top Drive Model		X
Bulk Mud	X	
Storage Mud Liquid	X	
Bulk Cement	X	
Last Upgrade Year		X
Overrun Rates	X	
Location (Latitude, Longitude)		X

[14] More details in the Glossary.

5.1.1 The AveragePerformanceFootagePerDay Field: dataset variations and category optimisation of the forecasted performance

Based on a range of utilisation scenarios defined by ODS-Petrodata offshore oil drilling rigs expert Robert Steven (section 2.3), I defined the aim of this work to be the forecast of the average performance of oil drilling rigs (*AveragePerformanceFootagePerDay*). The scenarios are all relying on rig performance or well duration. Performance and duration are intrinsically linked such that the performance of a rig on a given well is the drilled distance divided by its duration (in days). Using *AveragePerformanceFootagePerDay* as the targeted variable to forecast helps maximising the business use of the models. This is also comparable to the measure of drilling efficiency, according to the Norwegian industry standard (*drilling meters per day*) described in [124]. The data is based on the Gulf of Mexico market so the equivalent measure is in '*drilling feet per day*' instead.

AveragePerformanceFootagePerDay being the most important field of the dataset, – as it is the one I want to forecast – I empirically explored a few choices of categories before to select the set of performance to use in the dataset. Some variation of this dataset has been produced for experimental purpose. I call them *WRD2.0, WRD2.1, WRD2.2* and *WRD2.5*. Each sub-version number in the name of the dataset represents a variation of the categories. The datasets are mostly the same, with the only exception of the *AveragePerformanceFootagePerDay* field that has different categories. Those categories were empirically determined in order to find a 'natural' categorisation of the data. Various categories for *AveragePerformanceFootagePerDay* have been experimented with in order to improve the prediction accuracy by identifying categories which would closely match the unknown natural performance categories.

Ultimately, the following categories have been tested:

- *WRD2.0* uses 0-300; 300-700; 700-1000; 1000+.
- *WRD2.1* uses 0-300; 300-600; 600-800; 800+.
- *WRD2.2* uses 0-400; 400-600; 600-900; 900+.
- *WRD2.5* uses 0-200; 200-300; 300-400; 400-500; 500-600; 600-700; 700-800; 800-900; 900-1000; 1000+.

Those variations are the unique difference between *WRD2.0*, *WRD2.1*, *WRD2.2* and *WRD2.5*. Table 10, built using Weka's *BayesNet*[15] algorithm's results, shows that *WRD2.0* promises better results regarding the forecast accuracy abilities of the models. *BayesNet* is Weka's implementation of the *K2* algorithm.

Table 10: WRD2.x Bayesian and Logistic testing

dataset	% classified correctly	
	K2 Bayesian[16]	Logistic
WRD2.0	77.97 %	79.60 %
WRD2.1	73.46 %	74.85 %
WRD2.2	73.05 %	75.40 %
WRD2.5	49.57 %	49.46 %

The *WRD2.5* dataset has a different number of categories (10 categories) and, hence, cannot be directly compared in its prediction accuracy for the purpose of this empirical study of the categories of performance. The accuracy of models built with *WRD2.5* in predicting the exact rig performance category is lower than with *WRD 2.x* but I found it provides more useful information to the user by providing more specific categories. Table 11 is an example of how the categories could be combined when presenting the forecast results to the user in order to maintain a higher level of accuracy than presented for *WRD2.5* in Table 10. This is an improvement open for future experimentation.

Table 11: An example of WRD2.5 categories possible presentation

	0	200	300	400	500	600	700	800	900	1000+
0-300										
200-500										
300-600										
400-700										
500-800										
600-900										
700-1000										
800+										
900+										
1000+										

[15] *BayesNet* was run with the following parameters: 4 parents, no NaiveBayes initialisation, no Markov blanket correction, same random ordering for all 4 datasets.
[16] We note that the ordering was not optimised but chosen at random. We could obtain better accuracy from optimising the ordering but this is not the aim of that specific test.

5.1.2 Categorising the Data

The *Bayesian network* technologies used in this work require discrete categories and do not support continuous data. In this section the ways to create meaningful categories for the algorithms to use are reviewed. In addition to the expert advice, two methods (*discretisation* and *clustering*) have been used to inform the creation of meaningful categories for the data. The output of these categories has then been manually interpreted in order to obtain categories that are as meaningful as possible for the industry, data-wise and for humans. All the continuous fields have been run independently through a *clustering* algorithm and through a standard *discretisation* algorithm. Categories have then been hand-picked based on the 3 sources of information[17]. There are multiple techniques available for discretisation and automated learning of categories. Some are explored by Garcia [130]. "Discretisation is an essential pre-processing technique used in many knowledge discovery and data mining tasks" [130].

a) Expert Guided Data Categorisation

Following the data selection, some suggestions of data categories have been provided by the data experts (Table 12). Those categories are broadly based on the categories used within the company tools. The categories provide a starting point in transforming the continuous data into discrete categories through the process of discretisation. The categories provided are, however, subjective and no clear rationale could be provided by the expert. This is why it was decided to use *discretisation* and *clustering* techniques to complement that information.

[17] Those are listed in Table 13 and Table 14 (pages 82 & 83).

Table 12: Expert suggested categories for WRD2 selected data

Name	Categories
PreviousWells	0-5 / 5-10 / >10 per Wells/per Year
PreviousWellsSinceUpgrade	
PreviousWellsInRegion	
PreviousWellsInBlock	
PreviousWellsInField	
AverageUtilisation	+ / = / -
AveragePerformanceFootagePerDay	+ / = / -
ContractLength	0-3 / 3-6 / 6-12 / 1y+
RigAge	NEW / AVG / OLD (depending on market category)
LastUpgradeYear	
WaterDepthMax (WaterDepthMaxAsOutfittedNow)	Standard = water depth <3000 Deepwater = water depth >3000<7500 Ultradeepwater = water depth >7500
DrillingDepthMax	20K-,20K-25K, 25K-30K, 30K+
MatOrInd	Y/N/NULL
SlotOrCant	Y/N/NULL
RecentManagementChange	Y/N/NULL
ZeroDischarge	Y/N
DualActivity	Y/N
DerrickCapacity	Low,[?], high
MudPumpNumber	2,3,4-5
MudPumpHP	+ / = / -
WellDaysActive	30-,30-60,60-90,90+
TotalFootageDrilled	10K-,10-15K,15-20K,20+
TotalMeasuredDepth	10K-,10-15K,15-20K,20+
OperatorID	- / = / +
WaterDepth	Standard = water depth <3000 Deepwater = water depth >3000<7500 Ultradeepwater = water depth >7500 shallow water categories: 0-400 or 0-300,300-400
HuricaneRisk	Y/N

b) Discretisation

Discretisation is the process of converting continuous data into nominal data, categories or intervals. For this project, I approached the discretisation problem using Weka's [116] discretisation feature. This is an instance filter that discretises a range of numeric attributes in the dataset into nominal attributes using a simple binning algorithm. The range of continuous values is partitioned into segments of the same size. Each segment represents a bin, and numerical values are assigned to the bin representing the segment covering the numerical value. This is similar to the *equal-width discretisation* from [131]. The '*findNumBins*' [132], [133] option was used in order to let the algorithm select an optimal number of categories. This is done by the algorithm using a *leave-one-out cross-validation*. It searches for the best entropy given the data distribution [134].

Table 13's middle section shows the boundaries of categories as found by the Weka discretisation algorithm (page 66). Figure 15 and Figure 16 (pages 62-65) show the pre- and post-discretisation distribution of some of the data.

c) Clustering

A cluster is a "group of similar things [...] occurring closely together" [135]. In this work, I am using a clustering algorithm in order to gain insights of natural occurrences of groups of attribute values in the data. This provides a possibility of discovering natural categories within the data. For example, *TopDriveTorque* in Figure 15 (b) seems to shows 3 natural categories all the data seems to fall into. The clustering algorithm automatically identifies those categories whether they can currently be explained by data experts or not.

I used the *Expectation-Maximisation (EM)* [136] *clustering* algorithm in Weka to cluster values of the dataset variables one by one, independently. The algorithm is detailed in Figure 14. *Expectation-Maximisation* is a method used to find maximum likelihood of parameters in statistical models. It is used to compute the maximum likelihood estimate in the presence of missing or hidden data. It computes a probability distribution for each instance which indicates the probability of the instance belonging to each of the clusters [137]. *Expectation-Maximisation* can decide how many clusters to create by cross validation, or the user may specify how many clusters to generate. In this case, I choose to let the algorithm find the number of clusters to create.

```
The cross validation performed to determine the number of clusters is
done in the following steps:
1. the number of clusters is set to 1;
2. the training set is split randomly into 10 folds;
3. EM is performed 10 times using 10 folds cross-validation;
4. the loglikelihood is averaged over all 10 results;
5. if loglikelihood has increased the number of clusters is increased
by 1 and the program continues at step 2.
The number of folds is fixed to 10, as long as the number of instances
in the training set is not smaller 10. If this is the case the number
of folds is set equal to the number of instances.
```

Figure 14: *Expectation-Maximisation* clustering algorithm [136]

Table 14's top section shows the seeds found by the *Expectation-Maximisation* algorithm and the bottom part of the table shows the cluster boundaries that have been associated with each seed. The boundaries have been calculated by finding the middle value at an equal distance from each seed found by the algorithm. Table 13's top section shows the same category boundaries calculated from the seeds found by the *Expectation-Maximisation* clustering algorithm from Weka [116].

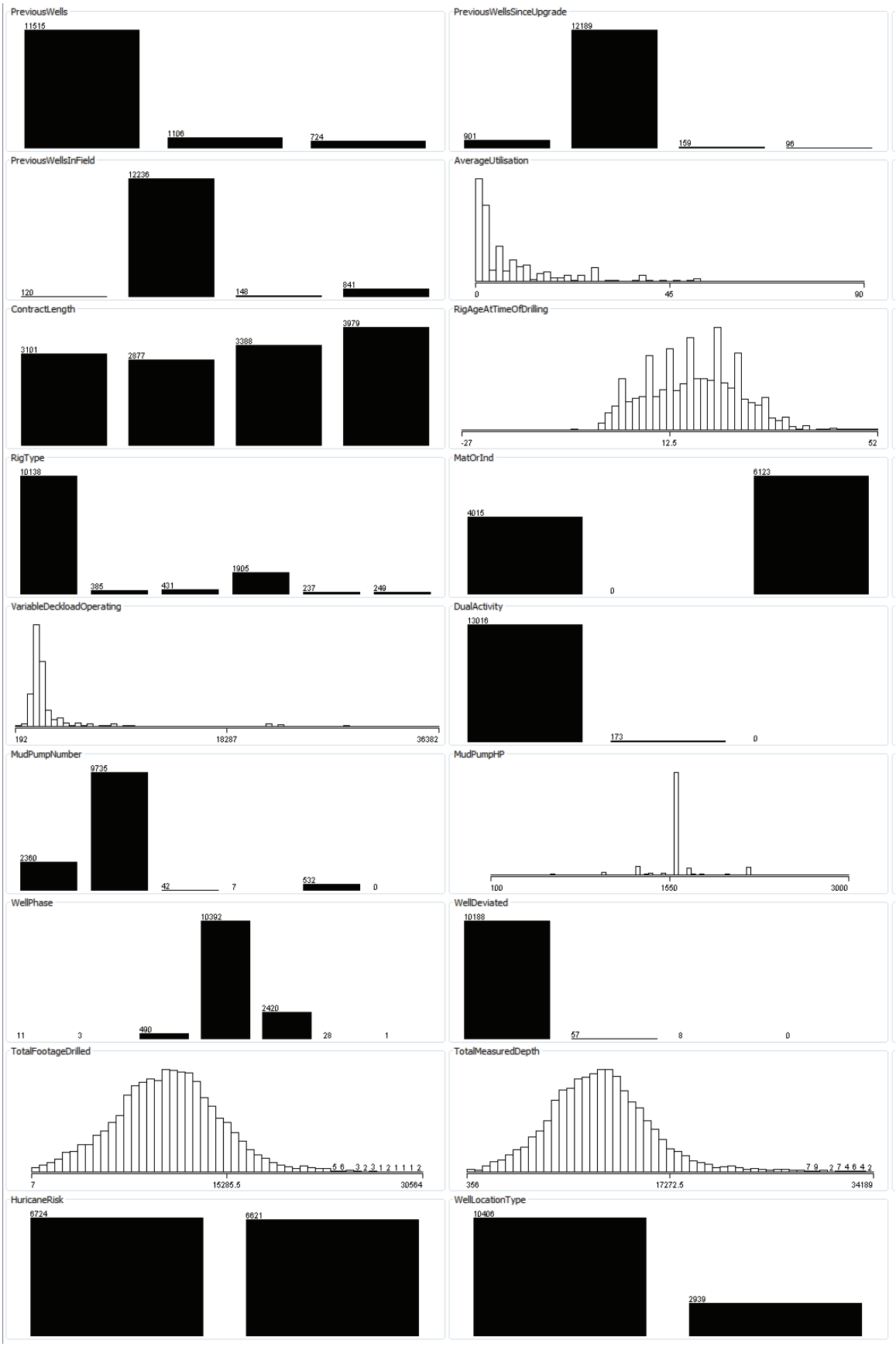


Figure 15: (a) Pre-discretisation data distribution graph as visualised in Weka

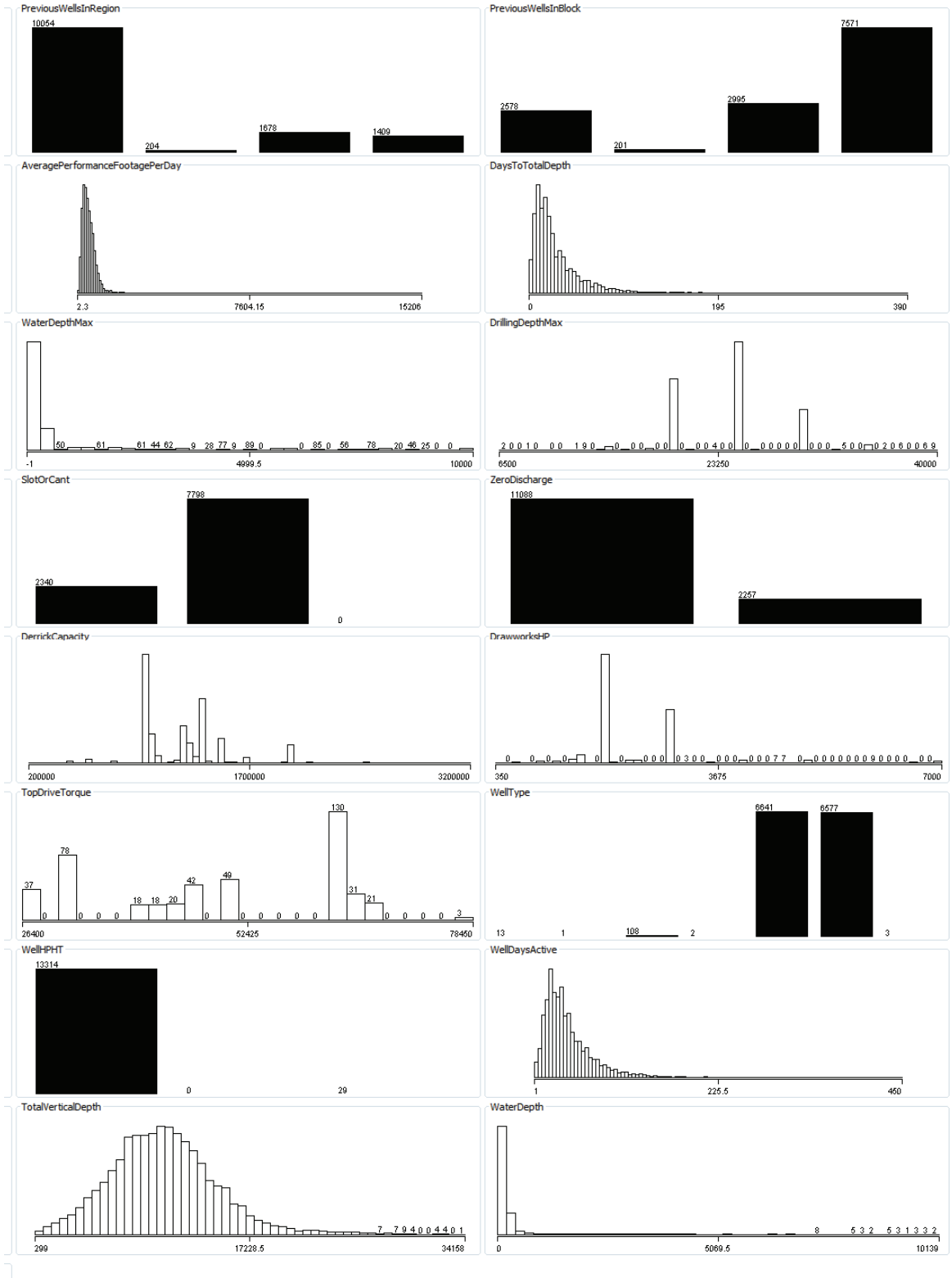


Figure 15: (b) Pre-discretisation data distribution graph as visualised in Weka

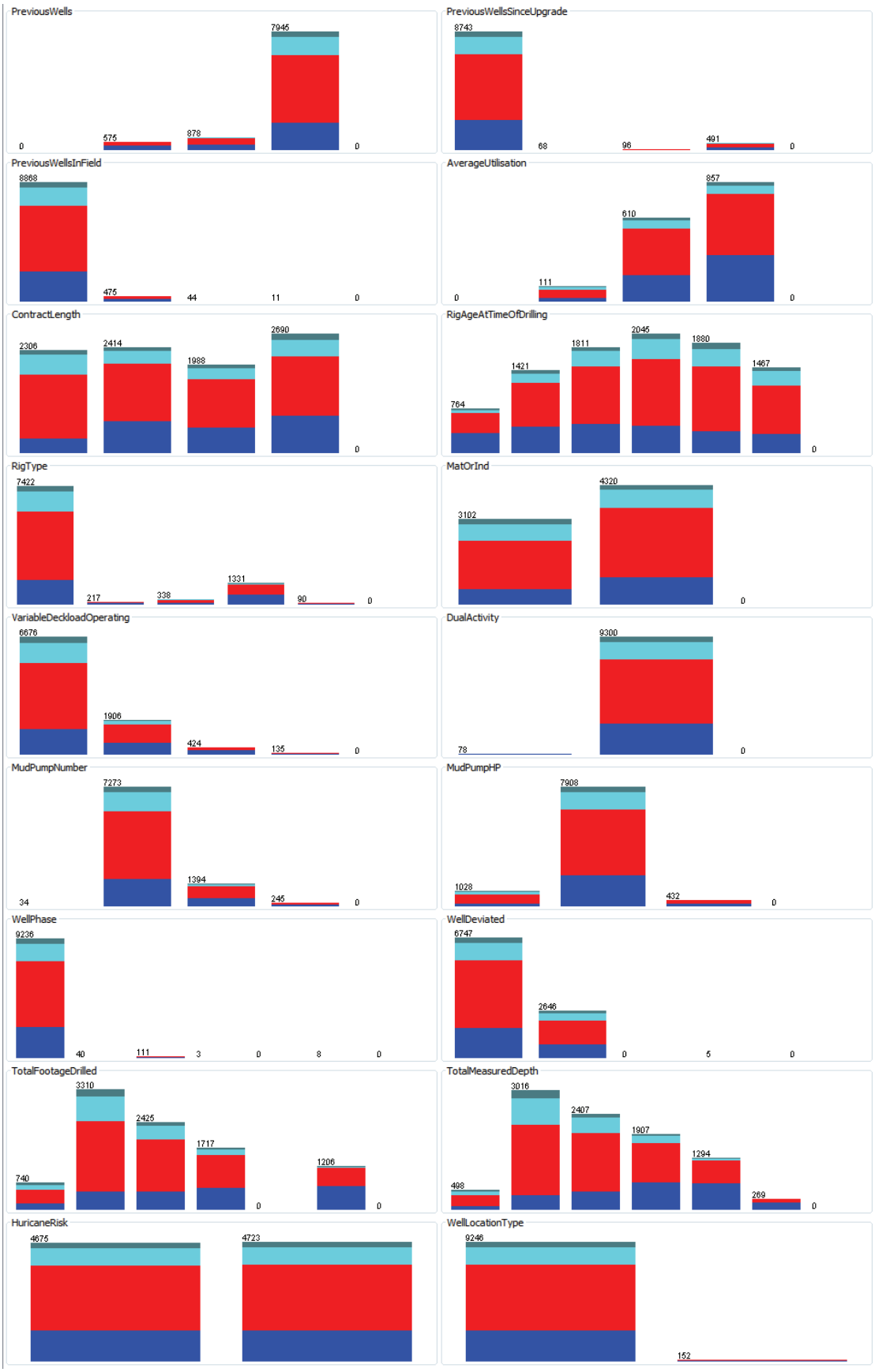


Figure 16: (a) Post-discretisation data distribution graph as visualised in Weka

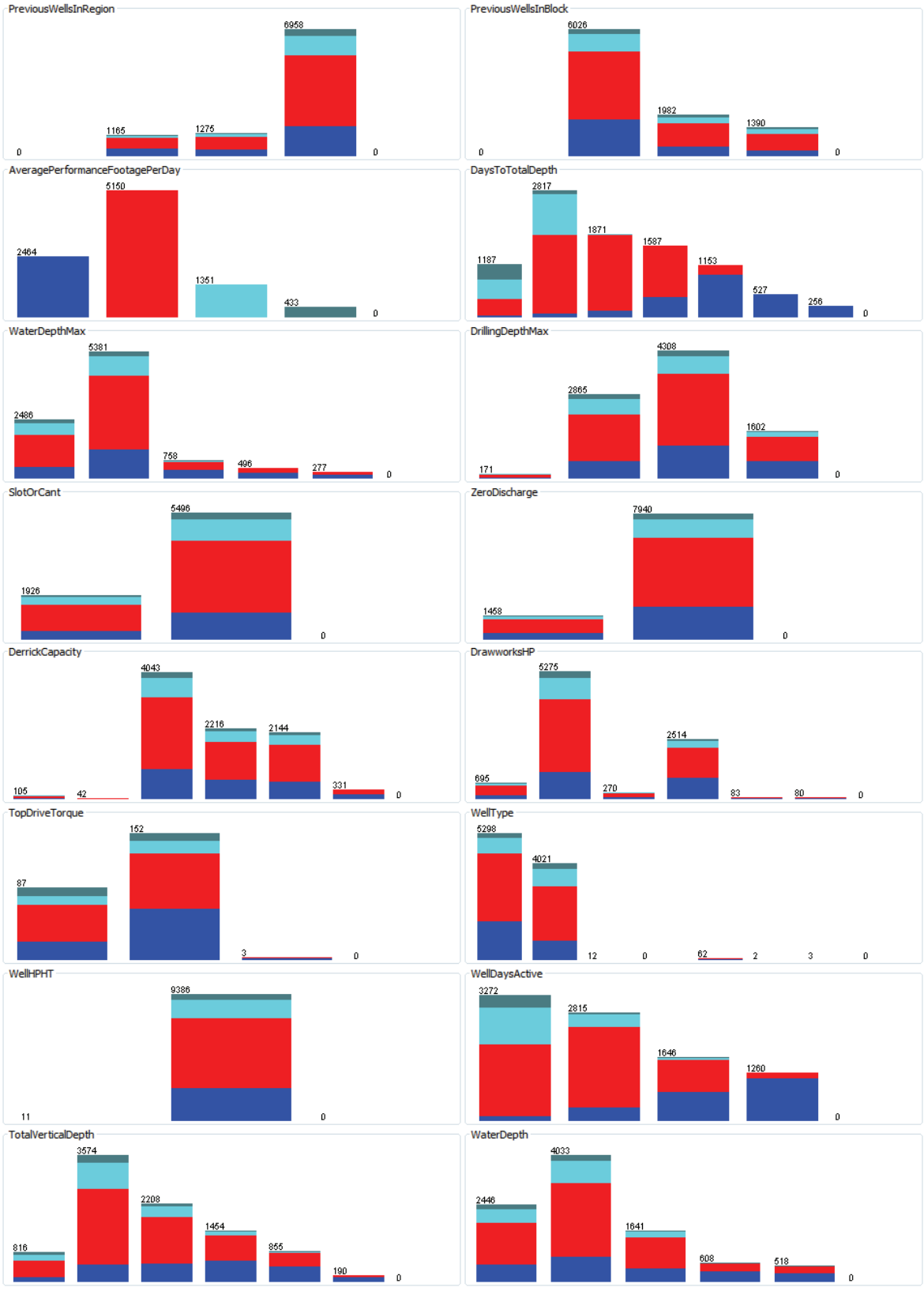


Figure 16: (b) Post-discretisation data distribution graph as visualised in Weka

Table 13: Discretisation and cluster values to manual category selection

field	manual categories	AverageUtilisation	AveragePerformanceFootagePerDay	DaysToTotalDepth	RigAgeAtTimeOfDrilling	WaterDepthMax	DrillingDepthMax	VariableDeckloadOperating	DerrickCapacity	DrawworksHP	MudPumpHP	TopDriveTorque	WellDaysActive	TotalFootageDrilled	TotalMeasuredDepth	TotalVerticalDepth	WaterDepth
		+=-	+=-		New/Avg/Old	0-3-7.5-+	0-20-25-30-+		Low/Avg/High		+=-	+=-	0-30-60-90+	0-10-15-20+	0-10-15-20+	0-10-15-20+	0-3-7.5+
average		8.137	520.303	27.541	17.046	1025.325	24263.272	2977.939	1238760	2381.23	1596.103	49112.416	41.572	10652	11635	10506	592.542
cluster		5	415	13	6	248	23006	2297	800364	2414	1510	43156	33	9495.5	6497.5	6886	124.5
values		11	574	23	12	971		4516	1028859		1866	55231	47	11429.5	9039	9646	391
			885	31	16	3180		14888	1151837				76.5		10824.5	11189	1858
			3104	45	21	6416			1273003						12235.5	13246	1561.5
				69	23				1356902						14797	16016.5	
				99	25				1681957						18862	19704.5	
discretisation		18	3043	78	11.2	1999	13200	7430	800000	1680	680	36810	91	6118	7123	7071	2028
values		36	6084	156	4.6	3999	19900	14668	1400000	3010	1260	47220	181	12230	13889	13843	4056
		54	9125		20.4	6000	26600	21906	2000000	4340	1840	57630	270	18341	20656	20614	6083
		72	12165		36.2	8000	3300	29144		5670	2420	68040		24453	27422	27386	8111
selected		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
value		5	300	10	5	250	20000	2300	800000	1700	1500	43000	30	5000	5000	5000	60
		10	700	20	10	1000	25000	4500	1000000	2400	1800	55000	50	10000	10000	10000	200
			1000	30	15	3000	30000	15000	1200000	3000			75	12500	12500	12500	600
				45	20	7500			1400000	4500				15000	15000	15000	3000
				70	25				2000000	5500					20000	20000	
				100													

Table 14: Clustering seeds and boundary values identified for data categorisation

field	AverageUtilisation	AveragePerformanceFootagePerDay	DaysToTotalDepth	RigAgeAtTimeOfDrilling	WaterDepthMax	DrillingDepthMax	VariableDeckloadOperating	DerrickCapacity	DrawworksHP	MudPumpHP	TopDriveTorque	WellDaysActive	TotalFootageDrilled	TotalMeasuredDepth	TotalVerticalDepth	WaterDepth
cluster seeds	1.7	310	8	3	236	20000	2007	600728	2000	1419	37200	24	8338	5029	4987	60
	8	520	17	9	259	26012	2586	1000000	2828	1600	49112	42	10653	7966	8785	189
	14	627	28	14	1683		6445	1057717		2132	61350	52	12206	10112	10507	593
		1143	33	18	4676		23331	1245957				101		11537	11871	3123
		5065	56	23	8155			1300049						12934	14621	
			82	23				1413755						16660	17412	
			116	27				1950159						21064	21997	
average	8.137	520.303	27.541	17.046	1025.325	24263.272	2977.939	1238760	2381.23	1596.103	49112.416	41.572	10652	11635	10506	592.542
cluster values	5	415	13	6	248	23006	2297	800364	2414	1510	43156	33	9495.5	6497.5	6886	124.5
	11	574	23	12	971		4516	1028859		1866	55231	47	11429.5	9039	9646	391
		885	31	16	3180		14888	1151837				76.5		10824.5	11189	1858
		3104	45	21	6416			1273003						12235.5	13246	1561.5
			69	23				1356902						14797	16016.5	
			99	25				1681957						18862	19704.5	

5.2 Experimental Exploration

In order to build the models, the new *Bayesian* models need to be learned from the *WRD2.0* dataset. As shown in chapter 4, *K2GA* is the algorithm that gives the best chance of obtaining a good result. The drawback is the computational cost associated with running the full score for each individual. The computational cost of learning a bigger model has been greatly reduced by carefully reducing the number of categories for each variable (section 5.1.2). The computational capacity has been greatly reduced by swapping the proprietary *K2* implementation used previously with the one implemented in Weka [138]. Both of these implementations are based on the same publications [28], [139]. There are multiple advantages to use the Weka implementation: it is much more efficient at calculating the *CH-score* (even if it uses much more memory) and, hence, allowed performing more runs in the allocated time. It's diffusion by the University of Waikato makes it more widely accessible to the community and, thus, would allow the reproduction of this project's results more easily. It has been tried and tested for many years now and, hence, supports a high confidence of the results generated. Finally, it provides more parameters and additional features to experiment with. In this research, I have used Weka programmatically to ensure a full control over the workings of the algorithms at any time and in order to have the ability to add the solution space exploration algorithms (*genetic algorithms* and *ant colony optimisation* algorithms) around the Weka implementation of *Bayesian networks* score calculation. In this section, I explore the set of parameters (section 5.2.1) and then run *K2GA* over the dataset with the same *genetic algorithm* parameters as with *WRD1* but with an increased number of individuals and generations (section 5.2.2). This chapter is finished by reflecting on the results obtained (section 0) and comparing the results with a node juxtaposition graph (section 5.2.3).

5.2.1 K2 Parameter Search

The 4 main parameters for Weka's *K2* implementation are:

- the scoring method to be used,
- the initialisation method (no link or *NaiveBayes* initial structure[18]),
- the option for a *Markov blanket* correction,
- the maximum number of parents allowed.

[18] NaiveBayes is a simple Bayesian network with a link from the classifier node to every other node.

Bouchaert [140] specifies about the *Markov blanket* correction that "after a network structure is learned a *Markov blanket* correction is applied to the network structure. This ensures that all nodes in the network are part of the *Markov blanket* of the classifier node."

The maximum number of parents allowed for each node in the network is an important parameter as the more parents a node has, the more complex the calculations are. I tested with 1, 2, 4, 6 and 8 parents. Using 8 parents proved to be not tractable[19] on the systems and was then removed from the list of parameters.

As with the previous implementation of *K2*, the algorithm performance is dependent on a good ordering being provided to the algorithm. The ordering is a description of the order of the variables available to the algorithm that should be considered. As *K2* attempts to build links between nodes, taking them one at a time and linking only forward in the list in order to prevent cyclic graphs, the algorithm is, then, constrained by that ordering and some of the links are impossible with some of the orderings. In this experiment, I picked one random ordering that was kept for the length of the empirical parameter search. The available scoring algorithms are the *CH*[20] (*Cooper-Herskovitz*), *BDeu* (*Bayesian-Dirichlet uniform likelihood-equivalence*), *MDL* (*minimum description length*), *entropy* and *AIC* (*Akaike information criterion*).

Table 15 shows the 80 combinations of parameters that were tested on *K2*. The best combination of parameters for the dataset used in this research seems to be the *CH-score* with neither a *NaiveBayes* initialisation nor a *Markov blanket* correction. On the given ordering (randomly generated), the *CH-score* co-varies with the forecast accuracy from the learned model. Two main factors seem to improve the results of *K2* on the dataset: the choice of measure and the maximum number of parents the algorithm can add. Some parameters seem to have a strong impact on the ability of *K2* to learn the network adequately. The *Bayesian network* initialisation is active in most of the poorly performing combinations. Moreover, the number of parents, when set low[21], prevents a good performance in learning the network.

[19] It was requiring over 10 GB RAM and brought the entire operating system close to collapse.
[20] *CH-score* is named "*BAYES*" in Weka.
[21] It is to be noted that when set to 1, the resulting search is similar to *Chain* from *ChainGA*.

Table 15: Weka K2 parameter search for WRD2.0

measure	init	markovB	parents	measureBayesScore	pctCorrect
BAYES	FALSE	FALSE	6	-187666.9344	78.53798681
BAYES	FALSE	FALSE	4	-189518.8061	78.53798681
AIC	FALSE	FALSE	4	-194005.1915	78.53798681
AIC	FALSE	FALSE	6	-194005.1915	78.53798681
BAYES	FALSE	FALSE	2	-198140.0154	78.53798681
AIC	FALSE	FALSE	2	-198336.7396	78.53798681
MDL	FALSE	FALSE	2	-199799.142	78.53798681
MDL	FALSE	FALSE	4	-199799.142	78.53798681
MDL	FALSE	FALSE	6	-199799.142	78.53798681
BDeu	FALSE	FALSE	6	-189114.4531	78.50606512
BDeu	FALSE	FALSE	4	-190257.7116	78.50606512
BDeu	FALSE	FALSE	2	-198231.7036	78.50606512
ENTROPY	FALSE	FALSE	2	-198581.6153	78.2613322
MDL	FALSE	TRUE	2	-204159.7767	78.16556714
MDL	FALSE	TRUE	4	-204159.7767	78.16556714
MDL	FALSE	TRUE	6	-204159.7767	78.16556714
BAYES	FALSE	TRUE	2	-203084.6611	78.15492658
ENTROPY	FALSE	TRUE	2	-204481.2933	78.11236433
AIC	TRUE	TRUE	4	-202044.0198	78.10172377
AIC	TRUE	TRUE	6	-202044.0198	78.10172377
AIC	TRUE	FALSE	4	-202044.0198	78.10172377
AIC	TRUE	FALSE	6	-202044.0198	78.10172377
BDeu	FALSE	TRUE	2	-203322.6025	78.08044265
AIC	FALSE	TRUE	2	-203845.4576	78.05916152
BDeu	TRUE	TRUE	6	-199015.6047	78.02723984
BDeu	TRUE	FALSE	6	-199015.6047	78.02723984
AIC	FALSE	TRUE	4	-200429.7711	78.00595871
AIC	FALSE	TRUE	6	-200429.7711	78.00595871
BDeu	TRUE	TRUE	4	-199470.0451	77.86763141
BDeu	TRUE	FALSE	4	-199470.0451	77.86763141
BAYES	TRUE	TRUE	6	-197169.7345	77.82506916
BAYES	TRUE	FALSE	6	-197169.7345	77.82506916
BAYES	TRUE	TRUE	4	-198486.8046	77.80378804
BAYES	TRUE	FALSE	4	-198486.8046	77.80378804
BDeu	FALSE	TRUE	4	-200064.4297	77.68674186
ENTROPY	FALSE	FALSE	4	-208861.5574	77.5909768
BAYES	FALSE	TRUE	4	-198355.608	77.54841456
BAYES	TRUE	TRUE	2	-212149.421	77.48457119
BAYES	TRUE	FALSE	2	-212149.421	77.48457119
ENTROPY	FALSE	TRUE	1	-216005.5376	77.48457119
ENTROPY	TRUE	TRUE	2	-212318.7684	77.4526495
ENTROPY	TRUE	FALSE	2	-212318.7684	77.4526495
BDeu	FALSE	TRUE	1	-215968.2032	77.4526495
AIC	TRUE	TRUE	2	-212218.9587	77.44200894
AIC	TRUE	FALSE	2	-212218.9587	77.44200894
MDL	FALSE	TRUE	1	-215838.2021	77.44200894
AIC	FALSE	TRUE	1	-215976.1745	77.44200894
BDeu	TRUE	TRUE	2	-212232.776	77.39944669
BDeu	TRUE	FALSE	2	-212232.776	77.39944669
BDeu	FALSE	TRUE	6	-201182.2421	77.37816557
BAYES	FALSE	TRUE	6	-198985.875	77.33560332
BAYES	FALSE	TRUE	1	-215845.1955	77.30368163
MDL	TRUE	TRUE	4	-211200.4229	77.29304107
MDL	TRUE	TRUE	6	-211200.4229	77.29304107
MDL	TRUE	FALSE	4	-211200.4229	77.29304107
MDL	TRUE	FALSE	6	-211200.4229	77.29304107
MDL	TRUE	TRUE	2	-213477.5362	77.23983826
MDL	TRUE	FALSE	2	-213477.5362	77.23983826
ENTROPY	FALSE	TRUE	4	-229014.6203	76.34603107
ENTROPY	TRUE	TRUE	4	-213479.0238	75.95233028
ENTROPY	TRUE	FALSE	4	-213479.0238	75.95233028
ENTROPY	FALSE	TRUE	6	-283811.4466	74.64354118
ENTROPY	FALSE	FALSE	6	-265344.6662	74.19663758
ENTROPY	TRUE	TRUE	6	-274930.0149	71.85571398
ENTROPY	TRUE	FALSE	6	-274930.0149	71.85571398
BAYES	FALSE	FALSE	1	-216097.6049	71.16407746
MDL	FALSE	FALSE	1	-216112.1024	71.16407746
BDeu	FALSE	FALSE	1	-216121.4323	71.16407746
AIC	FALSE	FALSE	1	-216124.7871	71.16407746
ENTROPY	FALSE	FALSE	1	-216165.693	71.16407746
BAYES	TRUE	TRUE	1	-268280.6007	60.76824856
BAYES	TRUE	FALSE	1	-268280.6007	60.76824856
BDeu	TRUE	TRUE	1	-268280.6007	60.76824856
BDeu	TRUE	FALSE	1	-268280.6007	60.76824856
MDL	TRUE	TRUE	1	-268280.6007	60.76824856
MDL	TRUE	FALSE	1	-268280.6007	60.76824856
ENTROPY	TRUE	TRUE	1	-268280.6007	60.76824856
ENTROPY	TRUE	FALSE	1	-268280.6007	60.76824856
AIC	TRUE	TRUE	1	-268280.6007	60.76824856

5.2.2 K2GA Runs

The *Bayesian network* models that represent the data from *WRD2.0* were built following the steps of *K2GA*, integrating Weka's *K2* implementation (*BayesNet*). As with *WRD1* in chapter 4, the *K2GA* algorithm was run with 200 generations with a population size of 30 node orderings. *Displacement mutation* and *cycle crossover* rates were 0.05 and 0.9 respectively. The selection used was a tournament selection of size 4. The *K2* parameters were to use the *CH-score*, no *NaiveBayes* initialisation, no *Markov blanket* correction and to limit the maximum number of parents for a node to 6. This experiment (*experiment 1*) was run 100 times. In the second experiment, I then ran the same algorithm with 1000 generation and a population size of 100 node orderings. This experiment (*experiment 2*) was run 100 times.

Table 16: WRD2.0 experimental run score results

		Worst Fitness	Average Fitness	Best Fitness	Min / Max
WRD2.0 (30/200) Experiment 1	Mean	-186869	-186499	-186069	-187770/-185019
	St. Deviation	386.23	375.16	389.21	
WRD2.0 (100/1000) Experiment 2	Mean	-185448	-185273	-185064	-186098/-184385
	St. Deviation	295.33	285.65	295.69	

Table 17: t-Test 2-sample assuming unequal variances for WRD2.0

	experiment 1	*experiment 2*
Mean	-186068.90	-185063.74
Variance	151488.03	87434.93
Observations	100	100
df	185	
t Stat	-20.56	
P(T<=t) one-tail	5.68e-50	
t Critical one-tail	1.65	
P(T<=t) two-tail	1.13e-49	
t Critical two-tail	1.97	

Table 18: WRD2.0 experiments, measures of model quality

	CH	Bdeu	MDL	Entropy	AIC	correctly classified	C-Index	Correlation
experiment 1	-185019.0306	-3940840.935	-2132204.403	-549433.2745	-895460.2745	78.1124 %	0.5424772	0.7720156
experiment 2	-184384.5946	-3396656.382	-1879578.172	-503301.3919	-804184.3919	78.6231 %	0.5424772	0.7775245

The structure was assessed, looking at the variability between the best ordering for each experiment from the industry's standpoint. Then, the edges frequency charts were reviewed and the observed differences between the algorithms were explained. The mean structure scores for each algorithm are presented in Table 16. Significance tests were carried out on the pair of means and the results are shown in Table 17. The differences between the two experiments are significant, beyond a 99.95% confidence level. According to the *CH-score* results, *experiment 2* produced on average significantly better scoring structures than *experiment 1*. The best-ever individual for *experiment 1*

scored *-185019* compared to *-184384* for *experiment 2* on the relative score scale (log of *CH-score*). Table 18 shows the other scores for both experiments as well as the ratio of correctly classified instances. This confirms that the improvements provided from the *genetic algorithm* evolutions continue to improve with additional generations and more individuals. As observed in the preliminary experiments, even though I am using only one algorithm in this set of experiments, there is a trade-off between model quality and computation time. Table 18 also shows the *C-index* and the correlation measure of the forecast with the real value for the best ordering of each experiment. The *C-index* remains exactly the same, indicating that the forecast power is stable for each model. For those results, as with the rate of correctly classified instances, the correlation of forecast and real value improves when the *CH-score* improves. Analysis of Data Relationships as Revealed by Bayesian Network Models

One of the advantages *Bayesian networks* models provide, as opposed to *Logistic* models, is the possibility to analyse the strength of links. The best networks from *experiment 1* and *experiment 2* using *K2GA* are displayed in Figure 17, Figure 18, Figure 19 and Figure 20. Both algorithms discovered interactions between specific nodes. *AverageUtilisation↔MudPumpNumber*, *AverageUtilisation↔TopDriveTorque*, are expected to be linked because the difference in the rig specification leads to a higher desirability on the market and to a higher utilisation. *DrawworksHP↔ MudPumpNumber, DerrickCapacity↔MatOrInd, RigType↔SlotOrCant, MatOrInd↔ZeroDischarge* and *DerrickCapacity↔MudPumpNumber* are all related to the specific range of equipment outfitted on certain models of rigs. Similar models of rigs have similar specifications. *DrillingDepthMax↔MudPumpNumber* are related to the capacity of a rig relative to its specification. *WaterDepth↔WaterDepthMax* are linked because a rig will often be used based on its technical capacity. The link between *ContractLength* and *PreviousWellsInBlock* is due to the availability of a rig. It is typical for a rig performing longer contracts to have less availability and, thereby, less previous experience in a given location as it navigates in between blocks less often. *WellDeviated↔WellType* and *WaterDepth↔WellLocationType* are related owing to the nature of oil drilling rig operations. Specific well types such as bypass wells, for example, have more tendencies to be deviated than others. Well location relates to a surface well or a kickoff well[22]. Furthermore, the water depth will impact the feasibility of drilling a kickoff well. *PreviousWellsInBlock↔WellType* are linked because of the nature of the well drilled. Exploration wells tend to be drilled where nobody has drilled before, hence, *PreviousWellsInBlock* will be low for that *WellType*. Finally, *DaysToTotalDepth↔TotalFootageDrilled* is a logical link provided that in most cases the deeper a rig needs to drill, the more time it will take.

[22] Surface well is drilled from the ocean floor, kickoff is drilled from the side or bottom of an existing well.

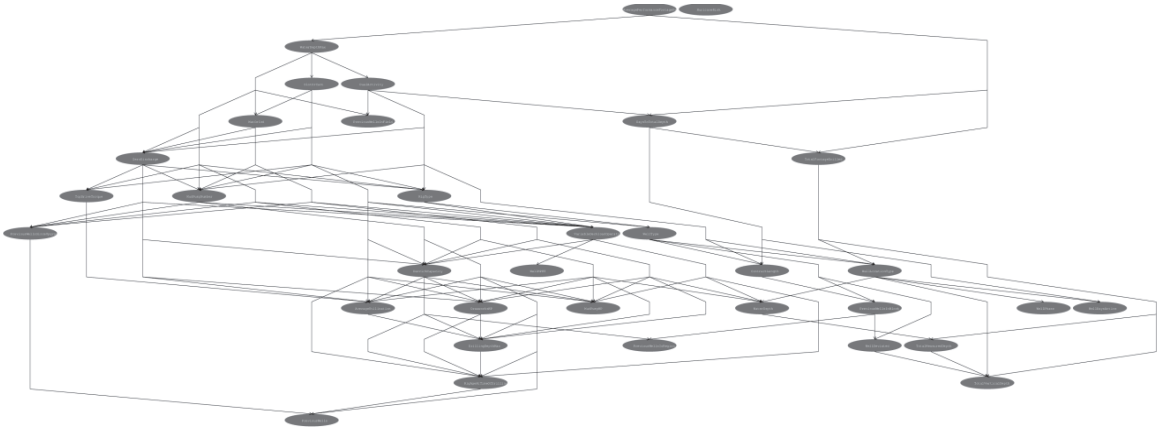


Figure 17: Overview of WRD2.0 experiment 1 best network score results[23]

node / parent	PreviousWells	PreviousWellsSinceUpgrade	PreviousWellsInRegion	PreviousWellsInBlock	PreviousWellsInField	AverageUtilisation	AveragePerformanceFootagePerDay	DaysToTotalDepth	ContractLength	RigAgeAtTimeOfDrilling	WaterDepthMax	DrillingDepthMax	RigType	MatOrInd	SlotOrCant	ZeroDischarge	VariableDeckloadOperating	DualActivity	DerrickCapacity	DrawworksHP	MudPumpNumber	MudPumpHP	TopDriveTorque	WellType	WellPhase	WellDeviated	WellHPHT	WellDaysActive	TotalFootageDrilled	TotalMeasuredDepth	TotalVerticalDepth	WaterDepth	HuricaneRisk	WellLocationType
PreviousWells										■											■														
PreviousWellsSinceUpgrade										■							■																		
PreviousWellsInRegion	■			■																														
PreviousWellsInBlock																								■										
PreviousWellsInField																		■															■		
AverageUtilisation											■			■	■	■		■																	
AveragePerformanceFootagePerDay																																		
DaysToTotalDepth							■																						■					
ContractLength				■				■										■																
RigAgeAtTimeOfDrilling											■			■		■		■	■	■														
WaterDepthMax							■																											
DrillingDepthMax						■								■						■														
RigType																																		
MatOrInd															■	■																		
SlotOrCant																																		
ZeroDischarge													■		■			■																
VariableDeckloadOperating													■	■		■			■		■													
DualActivity																																		
DerrickCapacity												■	■	■	■	■				■														
DrawworksHP						■					■		■	■	■	■					■													
MudPumpNumber						■					■	■	■		■				■															
MudPumpHP											■			■	■		■		■		■													
TopDriveTorque						■					■	■			■																			
WellType													■		■																			
WellPhase																																		
WellDeviated				■																				■										
WellHPHT																	■																	
WellDaysActive								■																										■
TotalFootageDrilled							■																			■								
TotalMeasuredDepth																													■					
TotalVerticalDepth																										■			■	■				■
WaterDepth											■	■	■	■																				
HuricaneRisk																																		
WellLocationType																									■									

Figure 18: Link details of WRD2.0 experiment 1 best network score results

[23] Ordering is 11,18,7,15,14,16,21,13,17,33,5,8,24,2,19,29,9,4,3,23,20,6,34,12,28,26,10,22,32,30,1,27,31,25

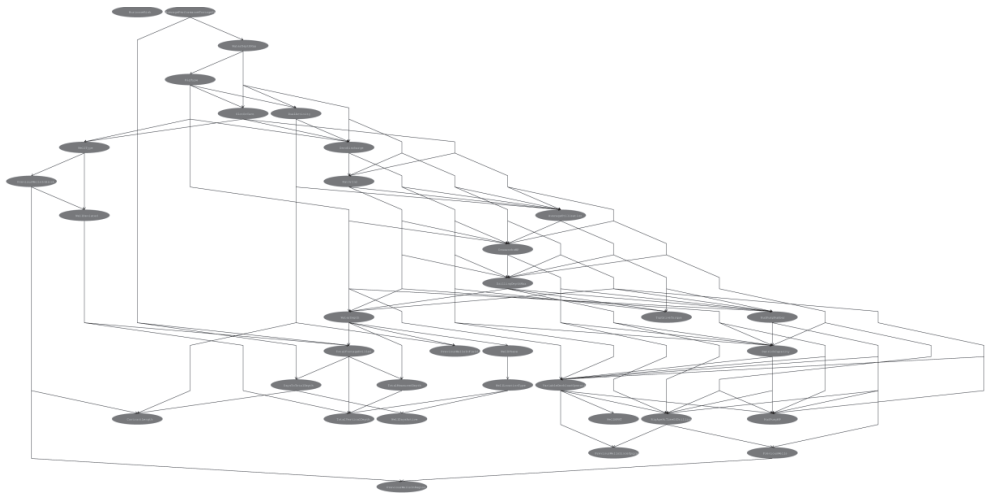


Figure 19: Overview of WRD2.0 experiment 2 best network score results[24]

node / parent	PreviousWells	PreviousWellsSinceUpgrade	PreviousWellsInRegion	PreviousWellsInBlock	PreviousWellsInField	AverageUtilisation	AveragePerformanceFootagePerDay	DaysToTotalDepth	ContractLength	RigAgeAtTimeOfDrilling	WaterDepthMax	DrillingDepthMax	RigType	MatOrInd	SlotOrCant	ZeroDischarge	VariableDeckloadOperating	DualActivity	DerrickCapacity	DrawworksHP	MudPumpNumber	MudPumpHP	TopDriveTorque	WellType	WellPhase	WellDeviated	WellHPHT	WellDaysActive	TotalFootageDrilled	TotalMeasuredDepth	TotalVerticalDepth	WaterDepth	HuricaneRisk	WellLocationType
PreviousWells		■								■											■														
PreviousWellsSinceUpgrade											■			■		■					■														
PreviousWellsInRegion				■															■																
PreviousWellsInBlock									■															■											
PreviousWellsInField											■							■																	
AverageUtilisation											■				■	■			■		■		■												
AveragePerformanceFootagePerDay																																			
DaysToTotalDepth								■										■																	
ContractLength									■									■						■											
RigAgeAtTimeOfDrilling												■			■		■		■	■	■														
WaterDepthMax							■																												
DrillingDepthMax						■					■			■				■	■	■	■														
RigType											■				■	■		■																	
MatOrInd											■				■																				
SlotOrCant											■																								
ZeroDischarge											■			■	■	■		■																	
VariableDeckloadOperating											■		■	■	■	■					■														
DualActivity											■																								
DerrickCapacity											■			■	■	■	■				■														
DrawworksHP											■			■	■	■			■		■														
MudPumpNumber											■			■	■			■																	
MudPumpHP											■			■	■		■		■		■														
TopDriveTorque											■				■	■																			
WellType													■		■																				
WellPhase																																			■
WellDeviated				■																				■											
WellHPHT																	■																		
WellDaysActive									■																										■
TotalFootageDrilled							■		■																										
TotalMeasuredDepth																														■		■			
TotalVerticalDepth																										■				■	■				■
WaterDepth											■			■			■																		■
HuricaneRisk																																			
WellLocationType																		■						■						■				

Figure 20: Link details of WRD2.0 experiment 2 best network score results

[24] Ordering is 33,11,13,18,15,16,14,6,20,24,4,7,12,26,32,29,21,19,30,5,23,8,25,17,9,10,1,27,34,31,22,28,3,2

5.2.3 Node Juxtaposition

Figure 21 represents the occurrences of node juxtapositions as a greyscale grid. The vertical axis represents the first node; the horizontal axis represents the second node. The shade is darker proportionally to the number of occurrences of node juxtapositions within the best ordering of each run for all four algorithms. As expected, *experiment 1* and *experiment 2* are exploring the search space in a similar way, consistent with *K2GA* in Figure 13. On the right hand side of Figure 21, *experiment 2* shows more signs of convergence (more contrasts) of the best individuals to specific node associations. This is explained by the fact that *experiment 2* has a bigger population and evolves for longer. I estimate that this is the start of a convergence and that the best orderings from *experiment 2* are closer to the search space optimal solution than the best orderings from *experiment 1*.

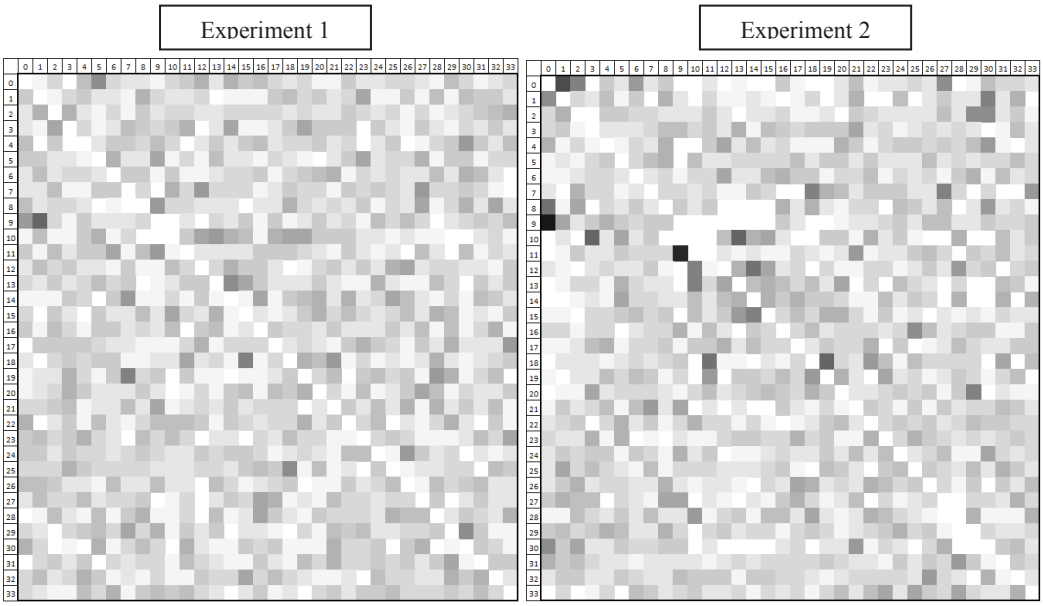


Figure 21: WRD2.0, experiment 1 & 2 best ordering node juxtaposition graphs

---

Chapter Summary: **This fifth chapter introduced the second dataset used in this research as well as its variations: *WRD2.x*. The building of the dataset was a big part of the work done for this section of the research and was described in details, including descriptions of the target field to forecast – *average performance footage per day* – and the methods used for categorising the data. After optimising for algorithms parameters, the chapter then reviewed the results for the *K2GA* runs and analysed the relationships identified by the *Bayesian networks* models in the data.**

Chapter 6: Result Validation, Scoring Variations and Feature Selection

In this chapter, I explore the validation of the models obtained and then perform additional experiments in view of improving the project's results. I first look at the results from lower scored orderings in order to compare the score results with the forecasting abilities of the models, based on a standard 10-fold cross-validation test (section 6.1). I proceed to compare the forecasting abilities provided by alternative modelling techniques (section 6.2). Then, I compare the forecast results with an alternative simplified forecasting technique (section 6.3.1) and with average and majority values from the model (section 6.3.2). Subsequently, the additional experiments performed are reviewed. The first one is a study of the covariance of the landscape of random orderings' *CH-scores* calculated with 5 different structures: *K2*, *Chain*, *Pyramid*, *BlockH* and *BlockV* (section 6.4). The *Pyramid* and *Block* algorithms are novel fixed structure *Bayesian network* scoring algorithms elaborated during this research. The second experiment performs a search for a reduced set of variables to forecast the *AveragePerformanceFootagePerDay* measure of oil drilling rig performance. This is known as feature selection (section 6.5).

6.1 Lower Scored Ordering Cross-validation Forecast Ability

I run a comparative analysis of the accuracy of the models generated using a 10-fold cross-validation and the *CH-scores* from *K2*. Figure 22 shows the scores and the accuracy from each of the best node orderings from *experiment 1* and *experiment 2* from chapter 5. The correlation coefficient between the two curves is -0.0228. I propose the hypothesis that *CH-score* might not be the best measure of adequacy for determining the worth of a network when considering the dataset. However, no better alternative has been identified at this time.

I extracted the most accurate network from the best node from the list of node orderings found in *experiment 1* and *experiment 2*. Its score is -185276 but its accuracy is 79.18%.

From this selection of best scores, I extract the model with the best cross-validation results. This network is shown in Figure 23 and Figure 24. One can first observe that the shape is slightly different to the other networks while the shapes from the networks issued from experiments 1 and 2 are relatively similar. However, all of the links that both *experiment 1* and *experiment 2* best scored networks found were also found by the best accuracy network shown here. All three networks have a similar number of links with 98 and 104 respectively for experiments 1 and 2 best scored networks and 102 for this most accurate network.

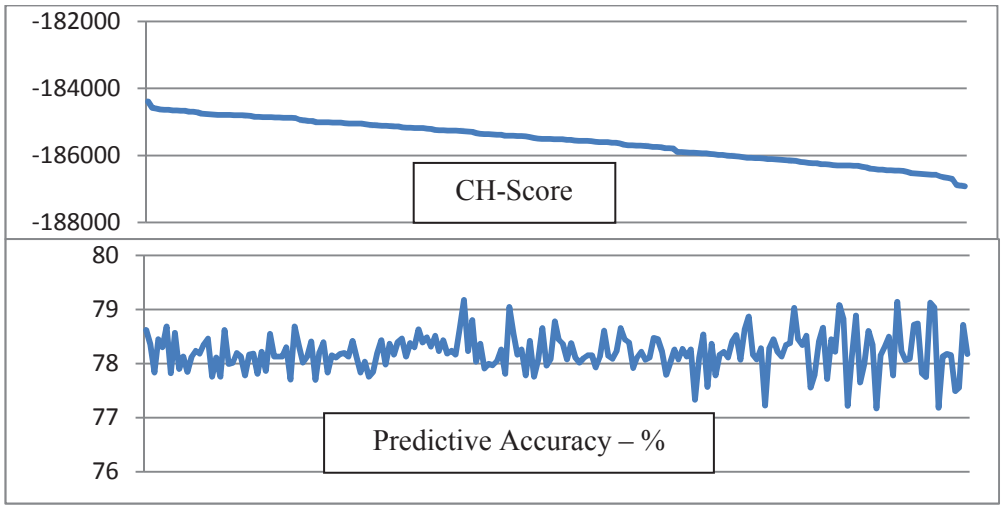


Figure 22: CH-Score and model accuracy correlation analysis (matching instances sorted by descending CH-score)

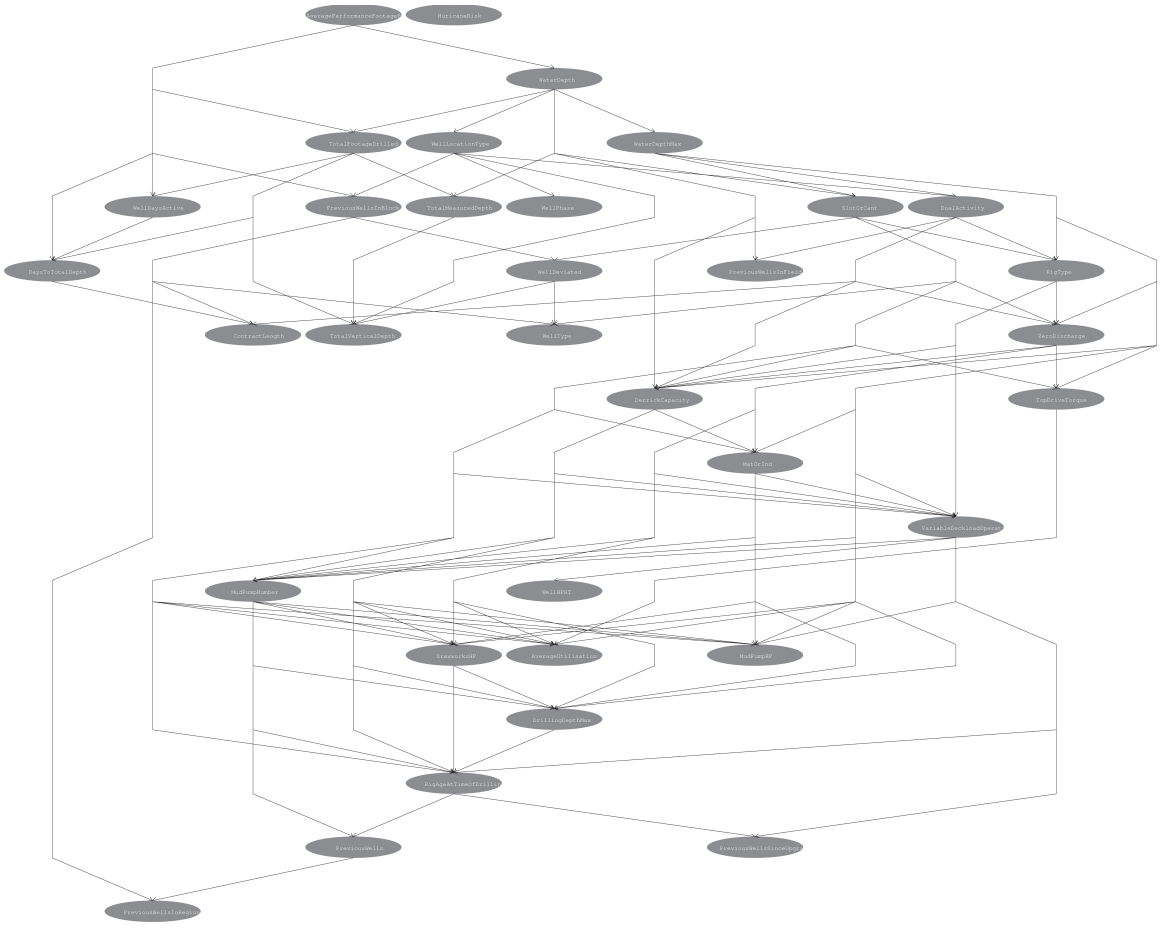


Figure 23: Shape of best network based on % correctly classified instances from a 10-fold cross-validation[25]

[25] Ordering is 32,29,34,11,4,28,15,26,18,13,16,19,14,25,23,24,7,17,21,27,30,8,20,12,31,22,10,9,33,1,6,2,5,3

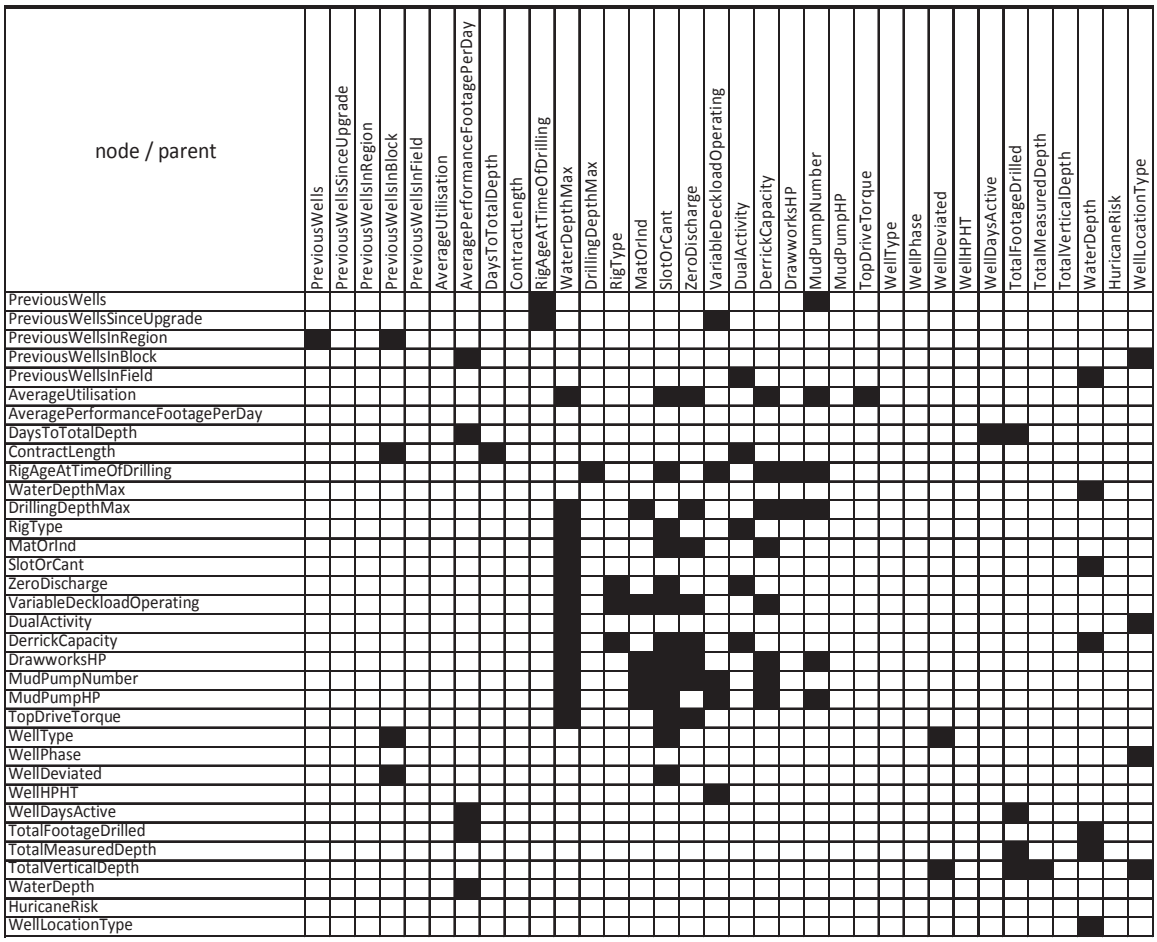


Figure 24: Link details of best network based on % correctly classified instances from a 10-fold cross-validation

6.2 Alternative Modelling Techniques

Using Weka data mining tool, I tested other algorithms on the *WRD2.0* dataset. The results of the testing for all the algorithms are displayed in Table 19 for *WRD2.0* and in Table 20 for *WRD2.5*. The Logistic regression algorithm and the *Bayesian network* learning algorithm are consistently performing better than most other algorithms listed in Table 19 and Table 20. The better accuracy of the forecasts on *WRD2.0*, when compared to *WRD2.5,* can be explained by the complexity of the attributes to predict. *WRD2.5* has 10 categories but *WRD2.0* has only 4 categories making it an easier set of values to work with. The other algorithms are performing well on the datasets: *J48*[26] and *DecisionTable*. They could be investigated in future work. All four algorithms (*Logistic*, *BayesNet*, *J48* and *DecisionTable*) are more complex algorithms than the others in the list with the

[26] J48 is an open source Java implementation of the C4.5 algorithm in Weka. C4.5 is a decision-tree building algorithm described in [215].

exception of *LBR*. However, *LBR* was unable to complete a run in the time available to run this experiment (7 days) when used on the *WRD2.5* dataset.

Table 19: Accuracy on predicting AveragePerformanceFootagePerDay from WRD2.0 with various Weka algorithms and 10-folds cross-validation

Algorithm	Cross validation, predictive accuracy on WRD2.0
weka.classifiers.functions.Logistic	79.60%
weka.classifiers.trees.J48[26]	79.26%
weka.classifiers.bayes.BayesNet[27]	79.18%
weka.classifiers.rules.DecisionTable	78.80%
weka.classifiers.lazy.LBR	77.42%
weka.classifiers.rules.OneR	71.37%
weka.classifiers.lazy.IBk (k=1)	66.95%
weka.classifiers.trees.DecisionStump	64.70%
weka.classifiers.bayes.NaiveBayes	60.70%
weka.classifiers.rules.ZeroR	54.79%

Table 20: Accuracy on predicting AveragePerformanceFootagePerDay from WRD2.5 with various Weka algorithms

Algorithm	Cross validation, predictive accuracy on WRD2.5
weka.classifiers.functions.Logistic	49.46%
weka.classifiers.rules.DecisionTable	49.70%
weka.classifiers.bayes.BayesNet[27]	49.62%
weka.classifiers.trees.J48	49.02 %
weka.classifiers.bayes.NaiveBayes	33.24%
weka.classifiers.lazy.IBk (k=1)	36.54%
weka.classifiers.rules.OneR	39.72%
weka.classifiers.rules.ZeroR	17.08%
weka.classifiers.trees.DecisionStump	22.96%

6.3 Comparing Results with a Simulated 'Manual' Approach

When interviewing the experts, I obtained a list of the most important data items they would use to forecast oil drilling rig performance and compared the results of using only those data items with the results from the models. This reduced collection of fields is presented in Figure 25 and is referred to as '*base fields*' below.

In this part I review a simple forecast method based on the base fields (section 6.3.1). Then I compare the results of the model-based forecast from this research to those simple forecasts (section 6.3.2).

[27] With the best ordering found in part 6.1 pre-set.

- MudPumpHP
- TotalMeasuredDepth
- TopDriveTorque
- WellPhase
- WellLocationType
- PreviousWellsSinceUpgrade
- RigType
- VariableDeckloadOperating
- DerrickCapacity
- MudPumpNumber

Figure 25: Base fields used by experts to evaluate oil drilling rig performance

6.3.1 NoModel Classification

In order to evaluate the approach taken in this research against the current practice of what experts do when performing an examination of the data, I created a '*NoModel*' classification algorithm. This algorithm is designed to simulate the thought-process data experts are using to produce estimates of rig performance for a given situation. This algorithm, first, compares each data row with the average (mean) category of all other rows with same base fields' values, then performs the same calculations but using a '*majority vote*'. The '*majority vote*' method uses the category the most represented in the subset of data rows as the value forecasted.

Figure 26 presents the results from both the average and the 'majority vote' methods. One can observe that using a '*majority vote*' method performs better as a predictor than using the average of the values observed. However, the '*majority vote*' still has a lower result than the models generated by this project as there are 296 cases in the dataset that do not have any additional data matching the '*base fields*'. The models, thus, lack the data to provide any forecast for their performance.

	categories	right	wrong	no data	accuracy
Average NoModel	0-300	1679	649	136	49.72%
	300-700	2982	2040	128	
	700-1000	10	1324	17	
	1000+	2	416	15	
Majority NoModel	0-300	856	1472	136	55.80%
	300-700	4358	664	128	
	700-1000	23	1311	17	
	1000+	7	411	15	

Figure 26: NoModel forecast results

6.3.2 Average and Majority Vote Forecast Validation

This analysis is comparing the forecast from the *Bayesian* model and the '*NoModel'* forecasted solution. As with the previous analysis, one can see that a '*majority vote'* system performs better

than the averaging system. Figure 27 shows that the *Bayesian* model and the '*NoModel*' approach forecast 58 to 66 % of cases similarly. This is a substantially lower success rate than using the 10-fold cross-validation. This suggests that the forecast loses a lot in quality when the data is restricted to these '*base fields*'. In addition, it suggests that the data-model approach can provide better results than the current approach, as devised by the industry experts.

	categories	right	wrong	unique data	match
Average Bayesian Inference	0-300	1932	532	136	58.15%
	300-700	2961	2189	128	
	700-1000	516	835	17	
	1000+	56	377	15	
Majority Bayesian Inference	0-300	1329	1135	136	66.31%
	300-700	4265	885	128	
	700-1000	566	785	17	
	1000+	72	361	15	

Figure 27: Average and Majority vote forecast validation

6.3.3 Model Validation Conclusion

Looking at the results of 200 orderings, the score results can be compared with the forecasting abilities of the models, based on a standard 10-fold cross-validation test. I found that there was no correlation between the two measures. Then I compared the forecasting abilities of the models generated by this project to the forecasting abilities of other models from standard modelling algorithms. *Logistic* is overall the best predictor, but it is more expensive to learn from the data. *J48* and *DecisionTable* might be good competitors but appear to be less stable than the preferred algorithm used in this work (*K2GA* using Weka's *BayesNet* implementation) as they perform unequally on the different versions of *WRD2*. Finally, the results were compared with an alternative forecasting method simulating the decisions that an expert might make. I conclude that the models developed in this research are providing a clear advantage over the current state of decision-support methods when forecasting oil drilling rig performance.

6.4 Study of Fitness Landscape Covariance from Random Orderings Using Fixed Structures and CH-Score

Fitness landscape is a metaphor used to develop an insight about the workings of processes, originating from Wright [141] and Haldane [142]. Jones mentions that "*the landscape metaphor originated with the work of Sewall Wright* [141]. *The idea has received wide attention within biology but has also been adopted by researchers in other fields*" [143]. In this experiment, I am

analysing the correlation of the fitness landscapes representing the space of *CH-score* node orderings, given a fixed structure learning algorithm. For each structure, I am comparing the fitness landscape from the fixed structure *CH-score* to the *K2* learned *CH-score*. The aim of this experiment is to identify if any structure is a natural alternative landscape, capable of providing scores easier to calculate than when using a full *CH-score*. The fixed structures used are *Chain* and *K2*, existing from previous experiments and the new pyramid structure as well as the two versions of Block structures. The pyramid structure is shown in Figure 28. Given an ordering, the chain is shown above the pyramid. The pyramid takes the last node as the ultimate child of the network and builds a succession of layers with 2 parents for each node. Figure 29 and Figure 31 show a vertical and a horizontal block structure. In a block structure, each layer contains the same number of nodes until there are no more nodes available for the last layer. Each node of each layer is the parent of the upper layer, as illustrated in Figure 29 and Figure 31. For this experiment, I chose to set the width of the vertical block structure to 3 and the width of the horizontal bloc structure to 7. Those parameters could be tuned to fit the specificities of each dataset but for the preliminary experiment those arbitrary values have been chosen.

Table 21, Table 23 and Table 25 show the correlation between the scores of each of the algorithms tested. Table 22, Table 24 and Table 26 show the statistical analyses of the scores for datasets of random orderings. Mainly, I observe that none of the scores are highly correlated with each other, meaning that they might not be good predictors of each other. However, I cannot infer that they would be a bad predictor of the final model's performance as the *CH-score* itself is a mechanism to approximate the value of a structure in regard to the modelling exercise, yet it is not designed to predict the model's performance when in use.

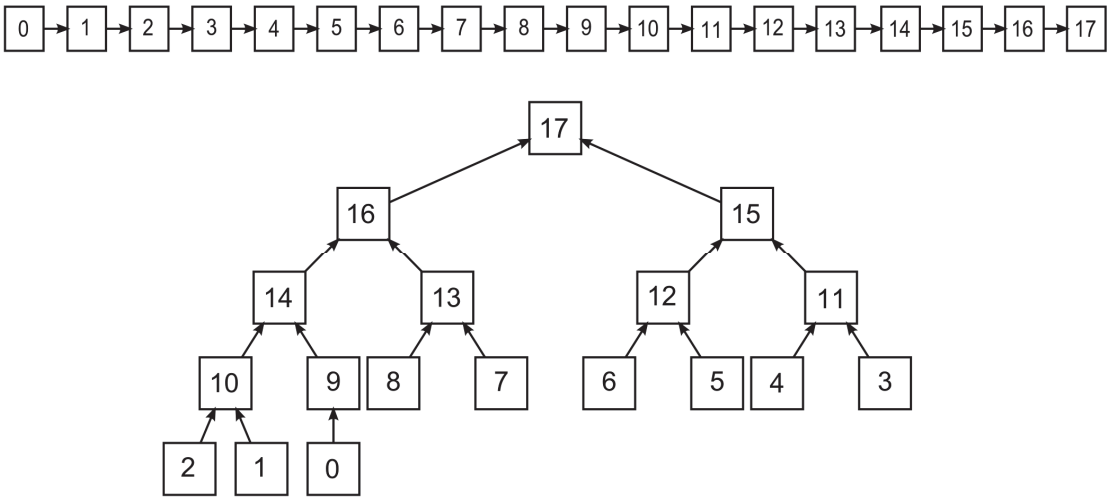


Figure 28: An example of pyramid fixed structure

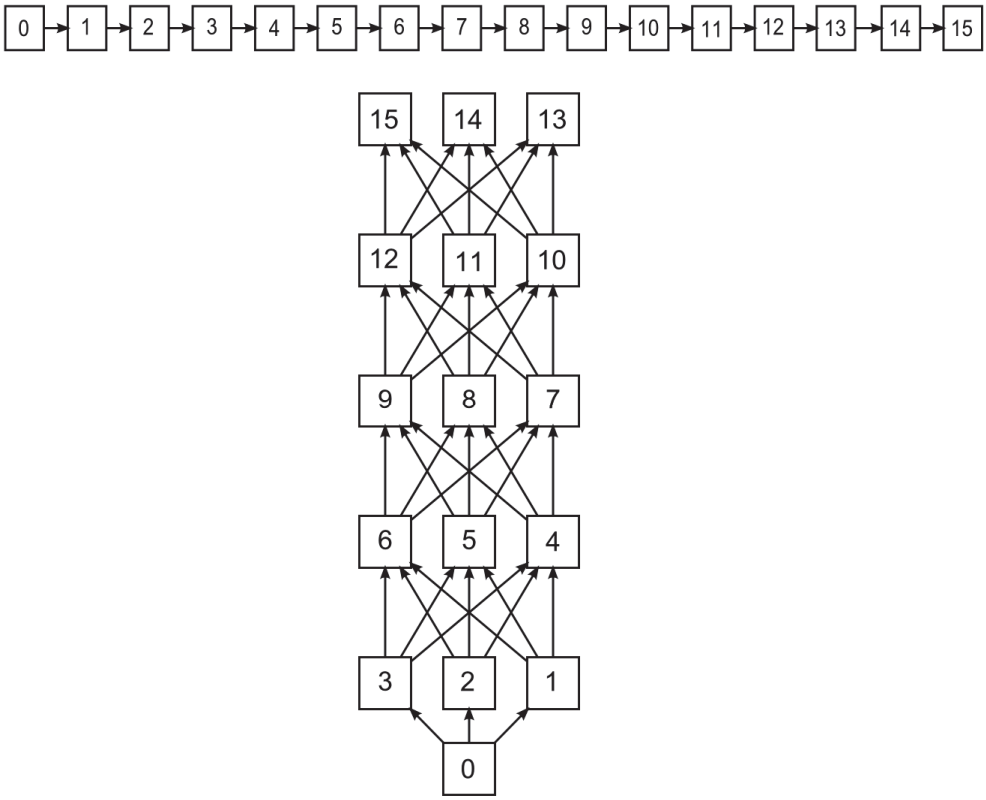


Figure 29: An example of vertical block fixed structure

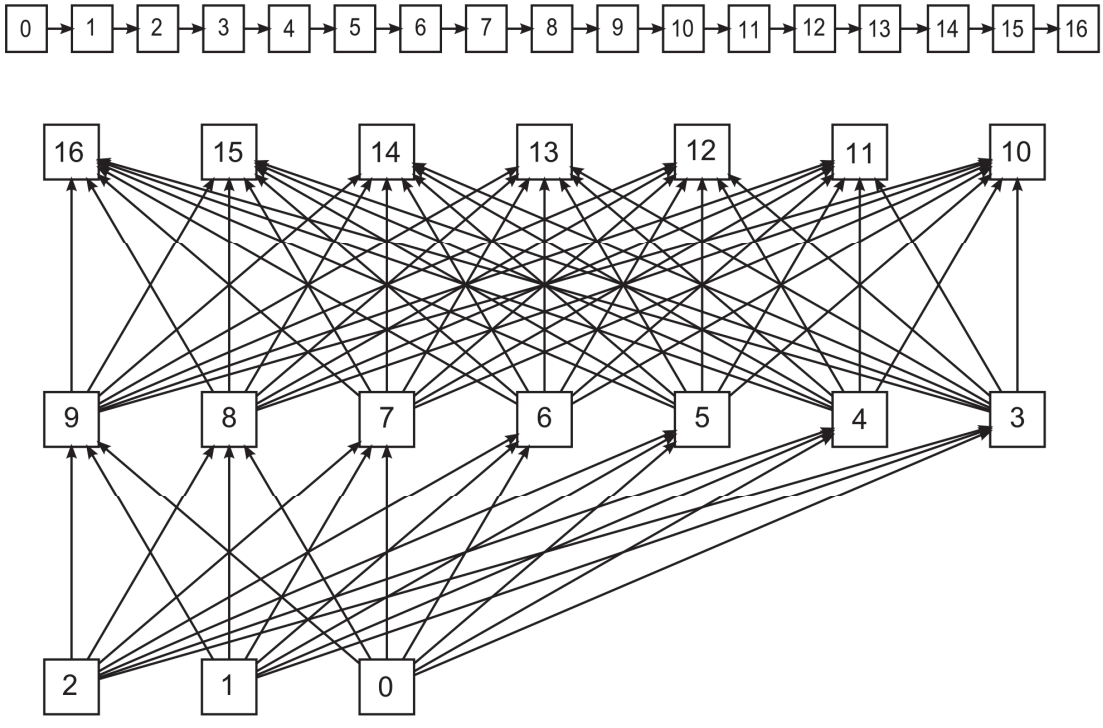


Figure 30: An example of horizontal block fixed structure

Table 21: ASIA10000 random ordering score correlations (40320 random orderings)

Chain	-0.12097			
Pyramid	0.04398	-0.14572		
BlockH	-0.01776	0.22733	0.19337	
BlockV	0.00140	0.00158	-0.01143	-0.00557
	K2	**Chain**	**Pyramid**	**BlockH**

Table 22: ASIA random ordering scores statistic description

	K2	**Chain**	**Pyramid**	**BlockH**	**BlockV**
Min	-22734.4	-30037.0	-30147.6	-29837.4	-29717.3
Max	-22621.2	-23251.0	-23173.5	-22736.0	-22699.5
Average	-22656.0	-27570.3	-27816.8	-26235.9	-25649.5
Stdev	25.19274	1616.155	1361.268	1701.787	1602.567

Table 23: ALARM random ordering score correlations (100 000 random orderings)

Chain	0.00622			
Pyramid	-0.00505	-0.08098		
BlockH	-0.00530	-0.16655	0.07090	
BlockV	0.00153	-0.00245	-0.00294	0.009543
	K2	**Chain**	**Pyramid**	**BlockH**

Table 24: ALARM random ordering scores statistic description

	K2	**Chain**	**Pyramid**	**BlockH**	**BlockV**
Min	-31430.6	-64848.9	-65030.4	-62250.2	-62985.0
Max	-29366.4	-50262.5	-50232.6	-39239.7	-41573.0
Average	-30235.4	-59891.2	-60376.0	-49291.9	-53065.2
Stdev	284.6189	1805.003	1735.753	2573.769	2466.428

Table 25: CAR random ordering score correlations (100 000 random orderings)

Chain	0.00224			
Pyramid	-0.00109	-0.06673		
BlockH	-0.00039	-0.00818	0.045068	
BlockV	-0.00275	0.000416	0.001769	-0.0026
	K2	**Chain**	**Pyramid**	**BlockH**

Table 26: CAR random ordering scores statistic description

	K2	**Chain**	**Pyramid**	**BlockH**	**BlockV**
Min	-23672.5	-40353.3	-40379.1	-40147.3	-40147.3
Max	-23106.9	-26555.4	-27154.1	-23801.4	-23801.4
Average	-23214.4	-37049.2	-37370.3	-32630.8	-32630.8
Stdev	95.26407	2262.88	2199.071	3217.075	3217.075

6.5 Feature Selection

In machine learning and statistics, feature selection (variable selection) is the technique of selecting a subset of relevant variables in order to improve the model building. Two techniques have been applied to the *WRD2.0* and *WRD2.5* datasets in order to reduce their size and, hence, the computation requirement necessary to learn a model from the data.

6.5.1 Weka Feature Selection

The feature selection, as performed in Weka [116], is composed of two algorithms: an evaluation algorithm and a search algorithm. Various implementations of these algorithms are available in the Weka framework for performing a feature selection. For this exercise, the following search algorithms were used:

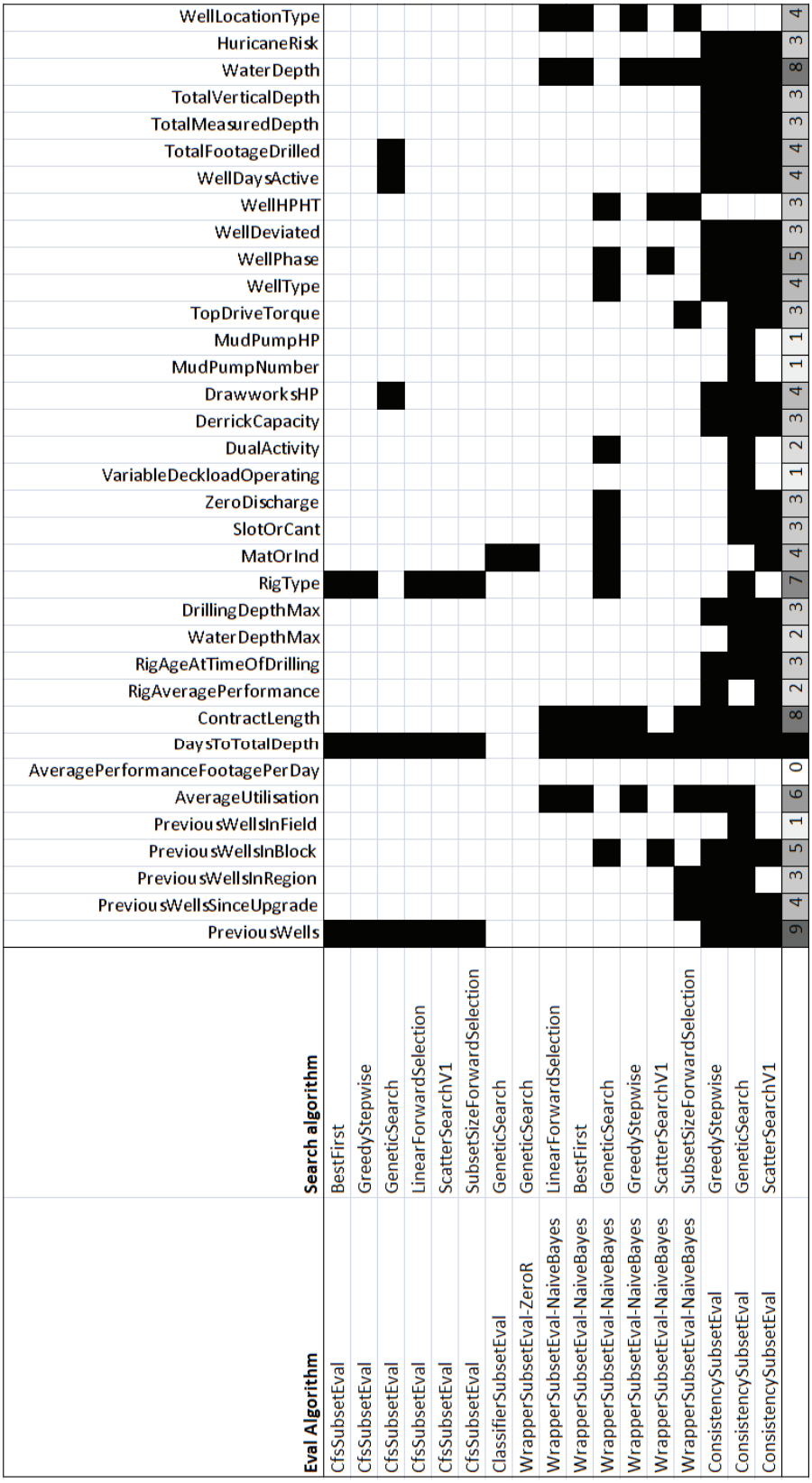
- **BestFirst** [144], [145] searches a space of attributes by greedy hill climbing augmented with a backtracking facility.
- **GreedyStepwise** [146] performs a greedy forward or backward search within the space of attributes.
- **GeneticSearch** [147] performs a search using the simple *genetic algorithm* described in [97].
- **LinearForwardSelection** [148] is an extension of *BestFirst*. It performs a ranking and takes a fixed number of k attributes into account. This wrapper algorithm is described in [144].
- **ScatterSearchV1** [149] performs a *Scatter search* [150] through the space of attributes. It is based on [151].
- **SubsetSizeForwardSelection** [152] is an extension of the *LinearForwardSelection*. In addition, the search performs an internal cross-validation. A *LinearForwardSelection* is performed on each fold to determine the optimal subset size. And another *LinearForwardSelection* is performed on the whole data, up to the determined optimal subset size [153].

The following evaluation algorithms were used:

- **CfsSubsetEval** is a *Correlation based Feature Selection* (CFS) [120]. It is based on the hypothesis that good feature sets contain features that are highly correlated with the class but that may be uncorrelated with each other. It evaluates the worth of a subset of attributes by looking at the individual predictive ability of each attribute along with the degree of redundancy between the attributes.
- **ClassifierSubsetEval** [154] uses a classifier to estimate the value of a set of attributes. By default *ZeroR* is the classifier used by this evaluation [154].
- **WrapperSubsetEval** [155] evaluates attribute sets by using a learning algorithm. Cross-validation is used to estimate the accuracy of the learning scheme for a set of attributes. Kohavi and John provide more information in [156].
- **ZeroR** [157] is a rule-based classifier. It predicts the results of any case as the mean (when numeric) or the mode (when nominal).
- **NaiveBayes** [158] is a simple probabilistic classifier based on the Bayes' theorem. It assumes the independence of all the attributes [159].
- **ConsistencySubsetEval** [160] evaluates a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes. Liu and Setiono provide more information in [161].

The results for the 16 feature selection runs are presented in Table 27. The black square indicates that the feature (data field) was selected, when a white square indicates that the feature was dropped. There are other techniques for features selection, as explored by Guyon [162] and Saeys [163]. The ones used in this work have the advantage to be readily and publicly available in Weka.

Table 27: Weka Feature Selection, variable selected by each algorithm and overall selection counts



The most selected features seem to be *DayToTotalDepth*, *PreviousWells*, *WaterDepth* and *ContractLength*, followed by *RigType*. This seems to suggest that the *CfsSubsetEval*-based algorithms would be more discriminating (3 fields selected for most) for the problem at hand than the *ConsistencySubsetEval*-based evaluation algorithms. *ClassifierSubsetEval* and *WrapperSubsetEval-ZeroR*, by their nature, were more discriminating and selected only one data field. For the problem at hand, those are the least informative of all feature selection algorithms as they are not providing enough features to learn a classifier.

6.5.2 Pearson Correlation

Pearson product-moment correlation is one of the most used [164] measures of dependence between two quantities. The correlation coefficient is obtained by dividing the covariance of the two variables by the product of their standard deviation. The correlation coefficient *r* is calculated by Equation 7. The correlation coefficient may take any value between -1.0 and +1.0. It is assumed that there is a linear relationship between x and y that are both continuous random variables. Both variables must be normally distributed and x and y must be independent of each other. Equation 7 expresses the Pearson product moment correlation, where *r* represents the correlation coefficient.

Equation 7: Pearson product moment correlation

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

In this part of the research, I calculated the general covariance of each continuous data when related to each other continuous data in *WRD2.0*, *WRD2.5* and, as a control, a version of *WRD2.0* pre-discretisation (*WRD2.0-preD*). This provides a matrix (Figure 30 to Figure 35, in appendix) that allows a comparison of similar data fields. In the chosen representation, I use different levels of greys to allow a visualisation of the correlation. The darker the shade, the higher is the correlation between two variables. I produced two sets of visualisations for each of the datasets. The first one (Figure 30, Figure 32 and Figure 34) shows the data fields in the order they came in the dataset. The second one (Figure 31, Figure 33 and Figure 35) is ordered horizontally to display the most correlated column to the rig performance measure on top of the tables and the least correlated column at the bottom of the table. This provides a ranking of the most correlated to the least correlated data fields, in relation to the measure of *rig* performance (*AveragePerformanceFootagePerDay*).

Table 28: Summary of feature selection methods and assembly of ranks into one value

	WRD2.0	WRD2.5	WRD2-preD	SUM	# FSA	SUM+(15-FSA)
RigType	0	0	0	0	7	8
WellPhase	0	0	0	0	5	10
WellType	0	0	0	0	4	11
WellDeviated	0	0	0	0	3	12
WellLocationType	1	1	1	3	4	14
AveragePerformanceFootagel	0	0	0	0	0	15
WaterDepth	3	3	3	9	8	16
TotalFootageDrilled	2	2	2	6	4	17
DaysToTotalDepth	7	4	8	19	15	19
PreviousWellsInBlock	4	5	7	16	5	26
TotalMeasuredDepth	5	7	5	17	3	29
TotalVerticalDepth	6	8	4	18	3	30
PreviousWells	9	9	6	24	9	30
WellDaysActive	10	6	18	34	4	45
PreviousWellsInRegion	11	11	14	36	3	48
DrillingDepthMax	15	12	9	36	3	48
DualActivity	13	14	12	39	2	52
SlotOrCant	17	13	10	40	3	52
PreviousWellsInField	12	16	11	39	1	53
RigAgeAtTimeOfDrilling	8	10	23	41	3	53
PreviousWellsSinceUpgrade	14	17	16	47	4	58
MudPumpHP	19	19	13	51	1	65
DrawworksHP	21	15	19	55	4	66
MatOrInd	23	18	15	56	4	67
AverageUtilisation	22	21	22	65	6	74
RigAveragePerformance	16	30	17	63	2	76
ZeroDischarge	24	22	21	67	3	79
ContractLength	26	27	20	73	8	80
VariableDeckloadOperating	18	24	27	69	1	83
WaterDepthMax	20	26	24	70	2	83
DerrickCapacity	29	20	25	74	3	86
TopDriveTorque	25	23	29	77	3	89
WellHPHT	28	25	26	79	3	91
MudPumpNumber	27	28	28	83	1	97
HuricaneRisk	30	29	30	89	3	101

I performed a Pearson correlation analysis of all orderable variables (the values of each variable were ordered and their rank used as a continuous feature; for example 0-100,100-300,300-500,500+ becomes 0,1,2,3) on *WRD2.0* (Figure 30, Figure 31) and *WRD2.5* (Figure 32, Figure 33). These correlations have also been calculated on the *WRD2-preD* dataset (pre-discretisation, see section 5.1 for the discretisation details) and are shown in Figure 34 and Figure 35. This shows a similarity in the ordering of most-correlated variables to *AveragePerformanceFootagePerDay*, regardless of whether the correlation coefficients are calculated from *WRD2.0*, *WRD2.5* and *WRD2.0-preD*. The experiment with *WRD2.0-preD* (Figure 34 and Figure 35) was performed as a

control. WRD2.0-preD contains the data before they were categorized. As shown in Table 28, the categorization did not impact drastically on the rank of the data fields.

One thing to note is that a correlation measure is not a measure of the causation [50]. These measures only allow to draw a parallel in the occurrence of the variables. This distinction is one of the points that have to be emphasised when explaining or presenting the results of the model. For example, if a user notices that the *DualActivity* attribute of an oil drilling rig is correlated with its performance, the user might be tempted to artificially change the variable to impact on the performance of the rig. As causation cannot be demonstrated using the data models, this cannot be guaranteed to have the desired effect. This ranking provides an indication of the level of information each of the data fields might carry towards the forecast of, among others, oil drilling rig performance.

6.5.3 Feature Selection Methods Combination

Table 28 shows 3 *WRD2.x* columns which correspond to the ranks of each variable from Figure 30, Figure 32 and Figure 34 (0 ranks corresponds to non-continuous variables, which have been selected by default). A sum of those ranks is presented in the 4th column. I define '*#FSA*' to be the number of times a variable has been selected using the Feature Selection Algorithms (FSA) in Table 27 (15 is the maximum possible *#FSA* value for each variable). By grouping all this information together, I obtain the "SUM+(15-FSA)" column in Table 28 that shows a combined rank from all the other columns. The variable names are ordered according to that rank. The variables that could not be ranked according to the co-variance are listed first.

Table 29 shows the cross-validation accuracy results. One can see that the *logistic regression* algorithm consistently performs better than the *NaiveBayes* algorithm on the *WRD2.0* dataset. The performance of *NaiveBayes* consistently reduces as the dataset number of variables is reduced. However, the performance of logistic regression improves by 0.0525% when the number of variables is reduced. The performance then drops as expected when further reducing the number of variables to 5.

6.5.4 Accuracy and Model Learning Time

NaiveBayes is one of the least well performing algorithms. *Logistic* and *BayesNet*[28] perform similarly to less than 0.5% accuracy difference. *Logistic* prediction accuracy is

[28] BayesNet is used here with a pre-optimised ordering.

slightly higher; however, the model learning time is more than 400 times longer, making it more expensive to use at the computational level. There is a slight (0.02-0.08%)[29] increase in the prediction accuracy of both *Logistic* and *BayesNet* when removing variables. This is similar to observations by Devaney et al. [165] and Janecek et al. [166]. This suggests that the models could be simplified and that it might improve their accuracy slightly but, as the change is small, more experiments should be conducted to validate that assumption. Reducing the number of variables used in building the model reduces its complexity and allows for faster calculations. Reducing the number of variables from 34 to 14, the model calculation is divided by a factor superior to 110, when using *Logistic* regression, and by a factor superior to 23, when using *BayesNet*. However, *BayesNet* is still the best-performing algorithm when comparing the model calculation times. When the number of nodes is reduced further there is the expected drastic reduction in the predictive accuracy of the algorithms.

Table 29: Test of reduced WRD2.0 dataset (using the feature selected) with the main Weka algorithms used previously

Algorithm	Number of variables	Prediction accuracy	Model learning time
NaiveBayes	34	60.70 %	0.03 seconds
Logistic	34	79.63 %	776.6 seconds
BayesNet	34	79.18 %	1.87 seconds
NaiveBayes	14	59.94 %	< 0.01 seconds
Logistic	14	79.75 %	6.94 seconds
BayesNet	14	79.20 %	0.08 seconds
NaiveBayes	5	56.40 %	< 0.01 seconds
Logistic	5	56.41 %	1.53 seconds
BayesNet	5	55.37 %	0.02 seconds

[29] Equivalent to 2 to 8 forecast instances on our 10-fold cross-validations across the entire data available.

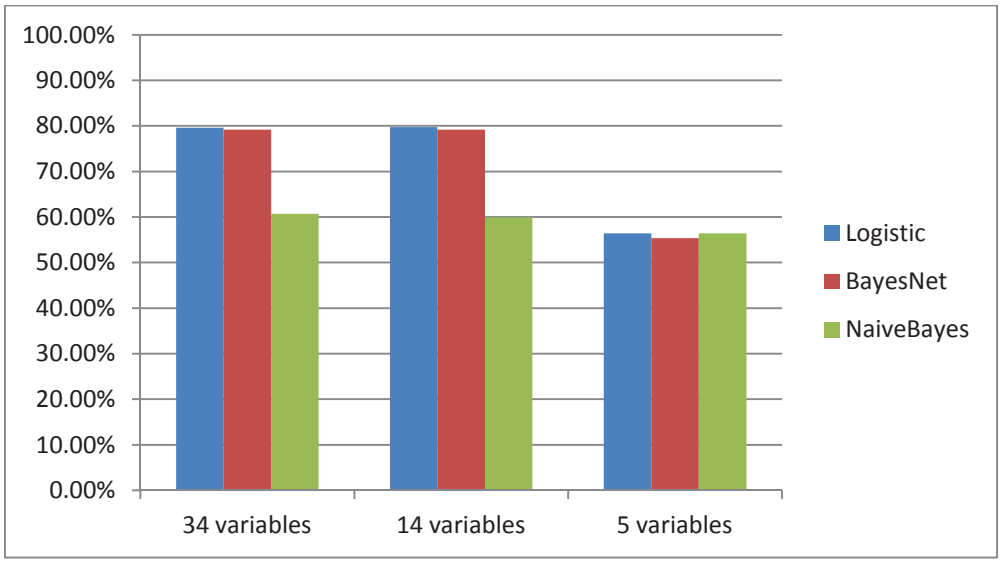


Figure 31: Accuracy for the tests of reduced WRD2.0 dataset

Chapter Summary: **The sixth chapter focused on the validation of the results and the comparison to the benchmark algorithm. Comparing the results from lower scored orderings with the forecasting abilities of the models, based on a standard 10-fold cross-validation test, I hypothesise that *CH-score* might not be the best measure of adequacy for determining the worth of a network when considering the dataset. Comparing the forecasting abilities provided by alternative modelling techniques showed that other standard algorithms such as *Logistic*, *J48* and *DecisionTable* have the ability to perform similarly well to *Bayesian networks* on the dataset. Comparing the forecast results to the alternative simplified forecasting technique mimicking an expert's forecast showed that a data-modelling approach can provide better results than the approach devised by the industry experts. The study of the covariance of the landscape of random orderings' *CH-scores* calculated with *K2*, *Chain*, *Pyramid*, *BlockH* and *BlockV* showed that none of the scores for each algorithm are highly correlated with each other, meaning that they might not be good predictors of each other but that more work is necessary to show if those new scoring mechanisms would be a good predictor of the *CH-score*. The feature selection experiment showed that *Logistic* and *Bayesian networks* perform similarly to less than 0.5% accuracy difference but that *Bayesian networks*' learning time is more than 400 times less expensive to process and that the models could potentially be simplified and that it might improve their accuracy slightly.**

Chapter 7: Value Creation and Commercial Applications

In this chapter, I review the application of *computational intelligence* as a competitive tool for commercial development in view of some recent literature (section 7.1). I then present the user interface that was developed to demonstrate the models in the case of developing a rig performance forecasting tool (section 7.2). Subsequently, I introduce a review of *recommender systems* as they have been investigated as a potential application of the models to benefit the oil and gas industry (section 7.3) and a review of scheduling technologies as an additional potential application (section 7.4).

7.1 Creating Value from Computational Intelligence

One of the main issues encountered while working on this research was the barriers raised in the course of interaction between different disciplines: abstract computing research and the real world of engineering and stakeholders. In his book "Applying computational intelligence" [27], Kordon explores the problems and issues associated with technology transfer between the world of advanced theory of *computational intelligence* and the world of practical applications of engineering methodologies to the industry. He explores the specificities of various *computational intelligence* techniques, their typical applications and the creation of value, and reviews application strategies. Throughout the book, Kordon's focus on 'competitive advantage' helps expanding the industrial side of this research. In addition to this short summary, some more materials from Kordon's work have been reviewed in the appendix.

The adoption of the technology I develop in this research relies on the credibility associated with it at the end of the project. Kordon highlights that "one of the differences between *computational intelligence* and the other high-tech alternatives is that it has already demonstrated its potential for value creation in many application areas" [27] (page 221). This confirms observations in the energy sector regarding the use of *computational intelligence* for wind power systems [167], electric power systems [168], thermal plants [169], and the oil industry (including reservoir characterization, gas storage, seismic inversion, engine oil development, oil field development, production scheduling) [170] as well as for biology [171]. Kordon explains that a variety of technology such as "search engines, word-processor, spell checkers and [...] rice cooker" are also everyday examples of the application of the technology [27]. One highly publicised event,

demonstrating further avenue of value creation of *computational intelligence*, was the "chess battle between Kasparov and Big Blue" [27].

In order to ensure the credibility of the approach taken in this project, the results are tested against historical data. I am using the multiple standard validation techniques of the model to ensure the quality of the results. This is the measure used to determine the reliability of the model produced.

In view of the work presented here, these theories suggest that when introducing the technology developed, it should not be forced onto the users but added as a benefit over the existing systems. The user should be able to opt out and still use the other tools and facilities previously available with no change if such is the user's need. The "*recommender system*" approach here is ideal as it provides the user with suggestions and insights without forcing an automated decision onto the user. This approach provides the benefits of reducing the "pain of adoption" as well as providing insights in a complex and uncertain environment.
Kordon [27] (page 316) proposes a "methodology for applying *computational intelligence* in a business". The key steps are:

- Introducing *computational intelligence* → Proof of concept projects.
- Applying *computational intelligence* → Several successful business projects.
- Leveraging *computational intelligence* → Growing value creation.

Within this methodology, the project stands in phase 1, with its purpose consisting in validating the technology potential based on pilot projects. In order to summarise the "factors that may influence the decision-making process of initiating a computational intelligence application", [27] provides a checklist which I am reviewing in view of this project:

- **Define appropriate application:** I developed models in order to forecast oil drilling rig performance.
- **Define competitive advantage:**
    - I created a measure and ability to measure the performance of oil drilling rigs.
    - I create the possibility to forecast that measure based on tangible decision elements.
- **Get management support:** From the start of the project, I secured management and field expert's support for the project. The results encouraged multiple product proposals which are now being considered for development at medium and long terms.
- **Allocate available stakeholders:** I secured the support of data engineers, field experts and modelling consultants in order to secure the success of the project. They regularly advised me along the project. This ensured an adequate support all along my research.

- **Check data quality:** I performed a study on the validation of the results which is presented in this document. I obtained good results and I am encouraged to continue developing better, more complete models to improve even more the quality of the forecasts.
- **Identify infrastructure needs:** I developed some demonstration software independently of any productisation. This step may be done within ODS-Petrodata's product portfolio as an option on some of the current products.
- **Estimate user attitudes:** The user acceptance will depend on the productisation phase. The suggested approach as it stands is to add a forecasting function without taking anything away from the current system. At worst, the system might encounter indifference, but no negative reactions and no loss of business. The productisation plan should aim to present the user with sufficient and clear proofs of the forecast provided and should highlight the benefits gained by having access to additional information to support decision making.
- **Estimate training needs:** A basic training to adopt the new technology related to c*omputational intelligence* will be required for software developers in order to perform regular maintenance. The user should not need any formal training if the user interface is correctly designed and explained as well as intuitively supportive of the user's needs.
- **Propose incentives for all stakeholders:** Some of the incentives identified are: *management* support can be maintained with regular updates, *data experts* can be provided with additional insight gained from the model analysis and *users* can be incentivised to use the new forecasting abilities with a free trial. This provides the additional advantage to obtain additional feedback in return for the free trial.

7.2 Rig Performance Forecasting: Demonstration Interface

The baseline of this research and development project was to extract knowledge from ODS-Petrodata's databases by identifying the inter-relationships implicit in the data. This knowledge about inter-relationships can then be used for forecasting. I developed various models in order to forecast oil drilling rig performance (average feet per day) and offshore well plan outcomes (days to total depth). I obtained an accuracy approaching 80% on a standard cross-validation test for some of the models from the Gulf of Mexico data.

Rig performance forecasting is a particularly interesting application of modelling technology. This application would support businesses in their decision to hire a specific oil drilling rig for a specific job by using performance expectation forecast. The prototype tool, showing the prediction of

AveragePerformanceFootagePerDay from a set of user-defined parameters is presented in the figures below.

1. A start screen allows the user to select one type of forecasting between two performance models and the number of days to total depth (Figure 32). This latter one is calculated using a model created by the data modelling algorithm as well.
2. The software then displays the main display screen with the first step of the wizard allowing selection of a rig name and some geographical data (Figure 33). The data on the main display is recalculated dynamically (Figure 34). The remaining wizard steps allow selecting some geographical information. The data selected are not the ones directly used in the models but they allow the retrieval of the data. For example, setting the rig name allows the software to retrieve the specification data from the company database and then to set them in the model to help the forecast.
3. The second step of the wizard allows selecting well information (Figure 35).
4. Finally, the software displays the performance forecast of the oil drilling rig (Figure 36). The screen then allows the user to adapt some of the specifications that the rig could hypothetically acquire.

This tool is designed to be easy to use as a demonstration tool. It is not designed to be a production tool as it is not integrated in the current company product suite and it is not designed to learn and update the underlying data and models to follow the market evolutions.

Figure 32: Starting screen for the demonstration allowing the user to select one type of forecasting.

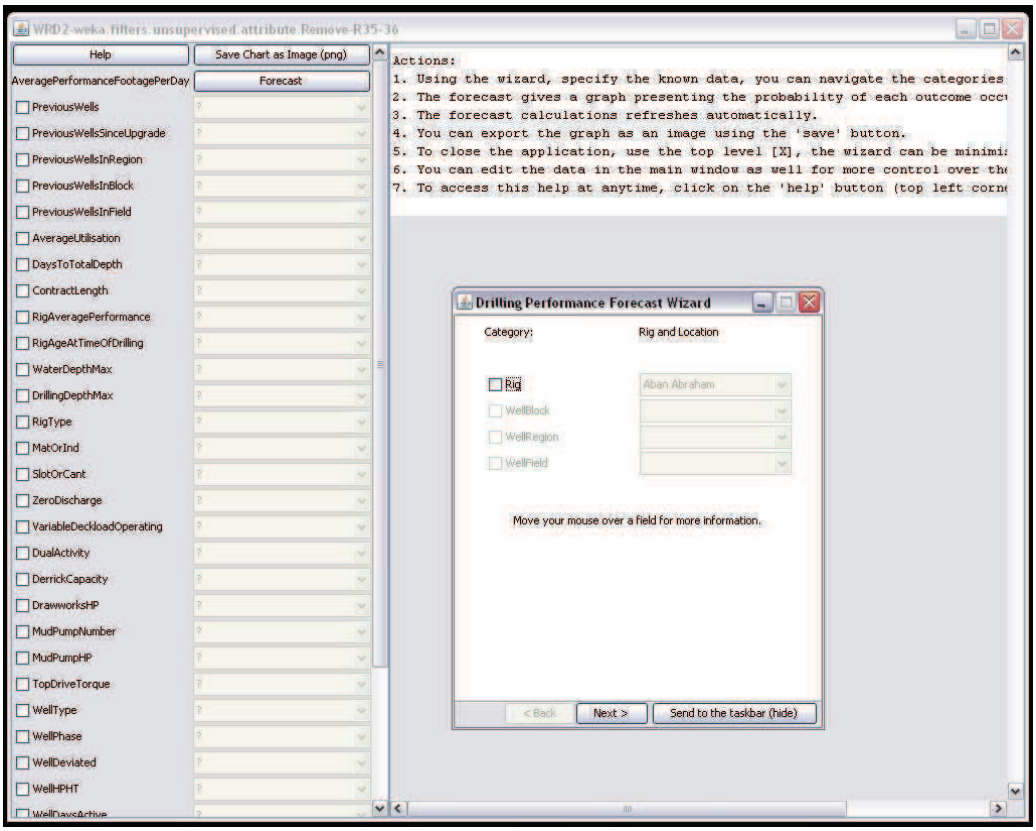


Figure 33: First step of the demonstration wizard allowing to select a rig and some geographical data

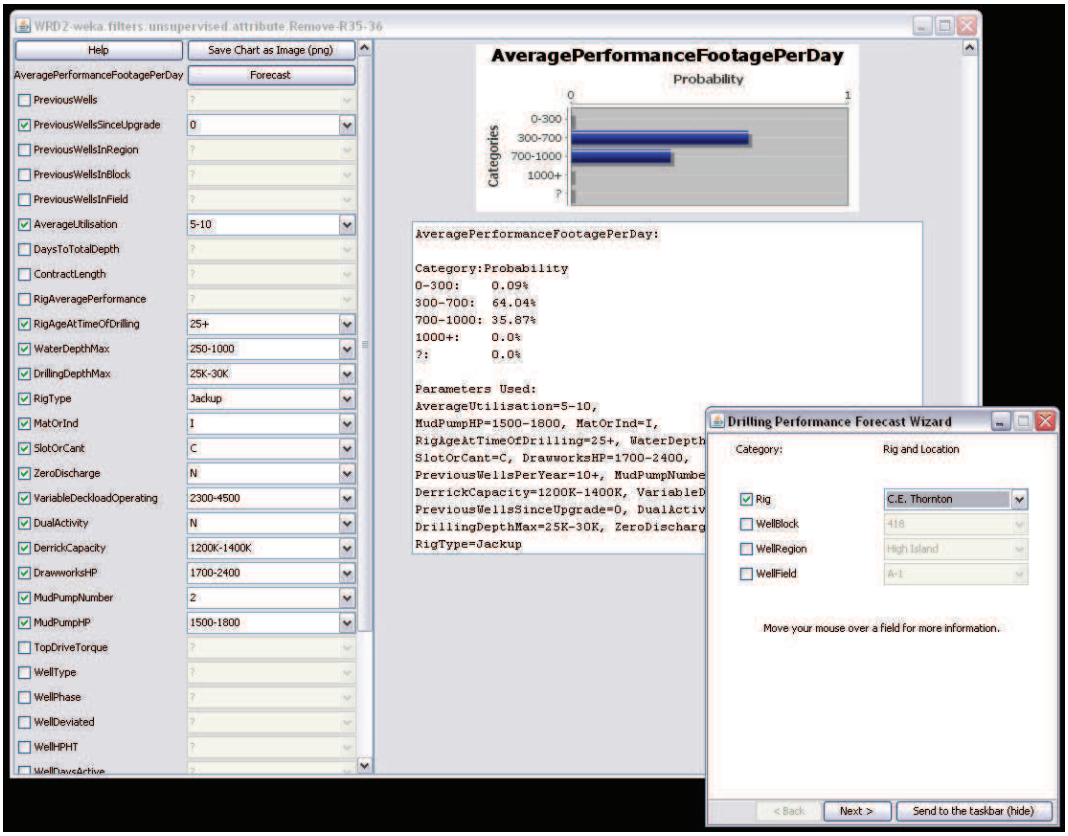


Figure 34: One selected rig when using the demo

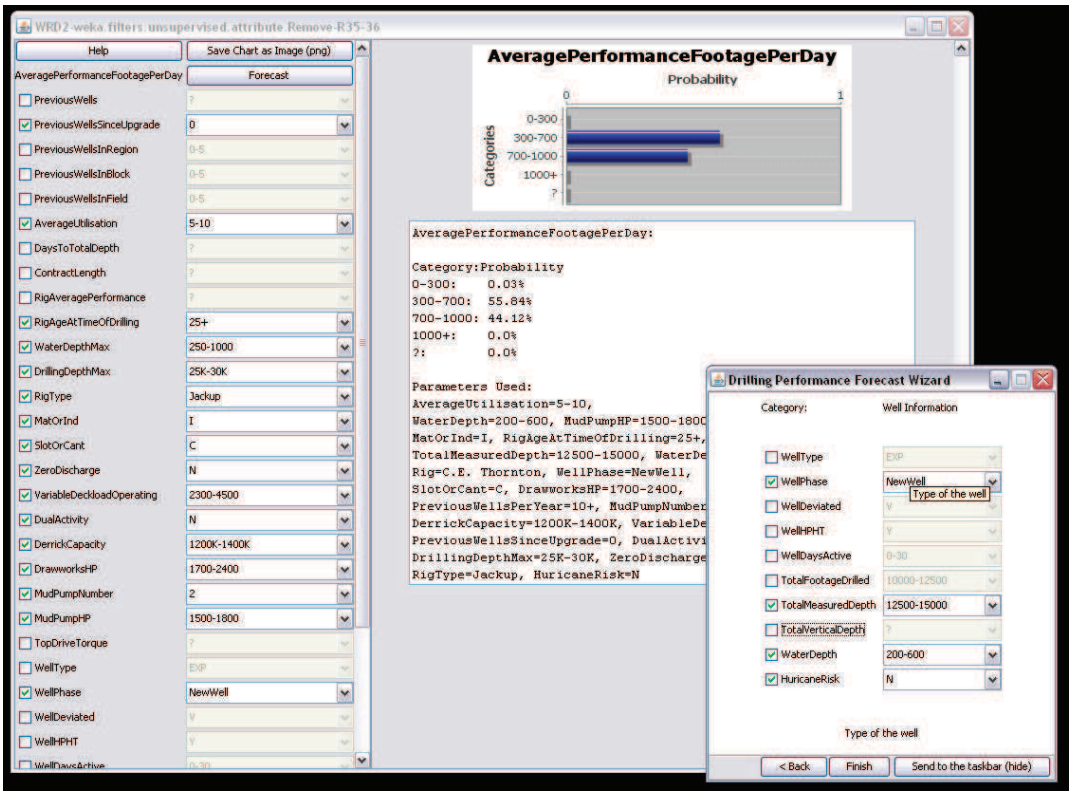


Figure 35: Step two of the demonstration wizard, selecting well information

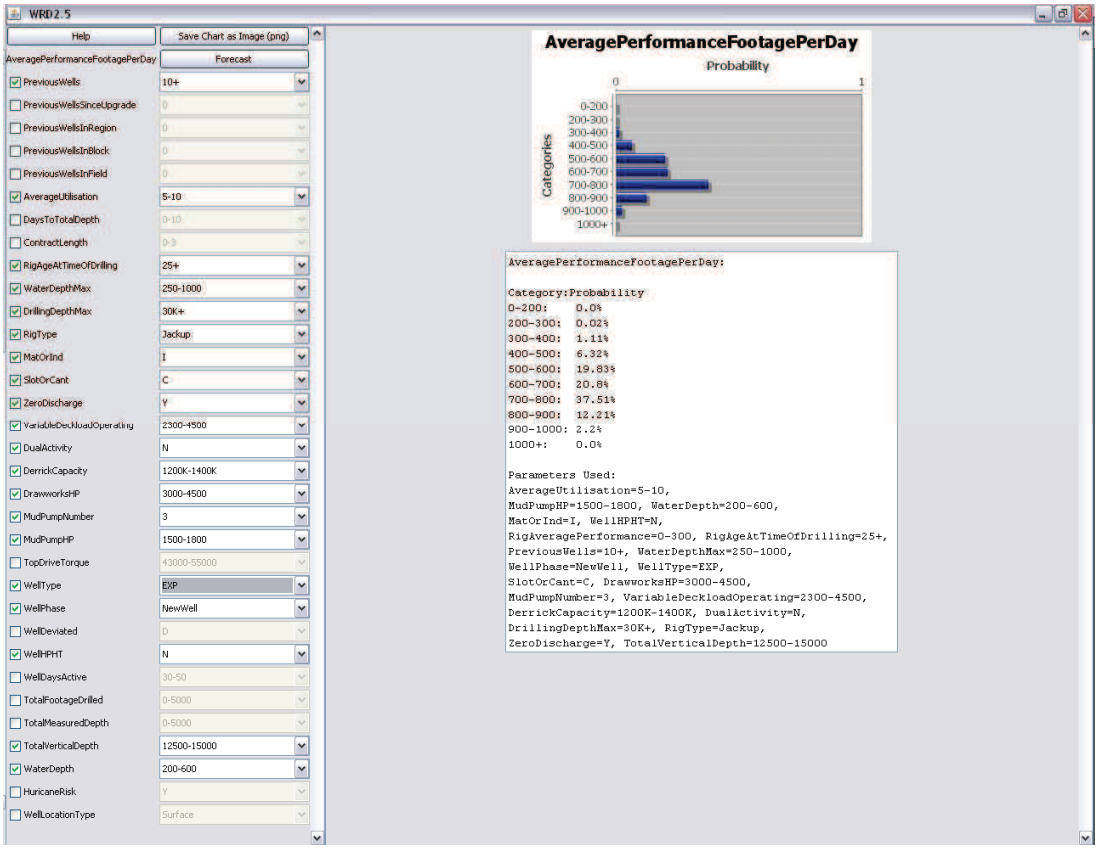


Figure 36: A display of the performance forecast of an oil drilling rig in the demonstration software

7.3 Recommender System for Oil Drilling Rig Selection

An application of a reliable model would be to recommend specific items to a user based on previous choices within the system and on the information content within the item [33,34,172]. A *recommender system* is a system performing information filtering to bring information items to a user; this information is filtered in a way that it is likely to interest the user [25]. This technology aims at providing users with relevant advice on the selection of items and could be applied to oil drilling rigs. The models obtained can constitute the basis for robust and flexible tools for assisting businesses in finding the best rig for a particular job. Consulting the model for optimal match, it enables the identification of rigs suitable for a specific operational demand. Adding variables related to the intended drilling task and user preferences in the model allows filtering relevant rig recommendations, using the *Bayesian* network-based model to provide an expectation of performance on a given task.

7.3.1 Recommender Systems: Technology Review

A *recommender system* is a system performing information filtering to bring information items such as movies, music, books, news, images, web pages or, in general, any item to a user. This information is filtered so that it is likely to interest the user. The aim of a *recommender system* is often to "help consumers learn about new products and desirable ones among myriad of choices" [173], [174].

Information filtering systems, more broadly, aim at removing redundant or unwanted information from a large information base. They aim at presenting relevant information and reducing the information overload, while improving the signal-to-noise ratio at the semantic level. This feature is relevant to the domain of application because the selection of a rig needs to fit the requirements of a demand (request for tender). In the world, there are over 1000 mobile offshore rigs available and not all of them are available or suitable for all demands. In addition, different companies (users) will have different preferences in regard to which parameters should influence their selection process for rigs. Those preferences could be, for example, cost-effectiveness, time of execution or health and safety ratings. Filtering the information to recommend rigs to users based on their preferences will help users to focus on the most relevant rigs for their needs.

According to Ujjin [175], "it seems that the definition of '*recommender system*' varies depending on the author. Some researchers use the concepts '*recommender system*', 'collaborative filtering' and 'social filtering' interchangeably" [175], [176], [177]. He also adds that others regard

'*recommender system*' as a generic descriptor that represent various recommendation and prediction techniques including collaborative, social and content based filtering, *Bayesian networks* and association rules [178]. Ujjin concludes his discussion by stating that he will assume the second definition in the rest of his publication. This seems to be the current assumption in the field and it is also the definition chosen by Herlocker et al. [179]. Therefore, that is the definition I will be using in this research.

Information of many types can be collected. "A simplified taxonomy separates recommender [systems] into content-based versus collaborative-filtering-based systems" [173]:

- Content based approach: the characteristics originate from the information item.
- Collaborative filtering approach: the characteristics originate from the user environment (social, user preferences, patterns, etc.).

One of the main issues for both approaches is the cold-start problem. New users have to interact with the system before a profile can be built up and the system becomes efficient for their needs [34]. Hybrid approach is often considered, by combining features from collaborative and content-filtering methods, to prevent such limitations. By asking a few targeted questions to the user, both systems are able to reinforce each other and learn the user preferences faster by comparing the responses to other user's responses and to items content to infer a potential profile. In this case, the cold-start problem is not a major issue as there is historical data available on most of the user's choices. The system would then be able to pre-establish their priorities and create their profile. One powerful feature of this approach is that any new choice from the user keeps improving their profile and allows following their preference variations over time.

The content based approach consists in analysing the content of the items being recommended. Each user is treated individually. There is no assumption of group or community [175]. The system works mainly by analysing items and the proximity of the selected items to others, selected by the user. Then, these items are selected to be recommended as they might interest the user. This approach is heavily based on which items are being considered by a user and their environment. The assumption is that a user interested in one item will be interested in a similar item. This effect happens in the rig selection problem. Users used to a successful work practice with a company will favour the same provider for similar projects. In item based filtering, items are rated and used as parameters for the matching instead of users. The items are grouped together and proposed to users. Users can then compare and rate them. User preferences are collected explicitly. Those preferences allow to group users by interest. The items are then selected using the ratings of a similar user.

Collaborative filtering "mimics word-of-mouth recommendations" [173]. Herlocker et al. [179] states that "one of the most successful technologies for *recommender systems* [is] called collaborative filtering". Collaborative filtering systems come from the earlier information filtering systems. Those systems were developed in order to bring only relevant information to the user by observing previous behaviours and, thus, build a user profile. This system is based on the collection of taste information from many users [175]. It assumes that a group of users will have a similar appreciation of items, then aims to "predict the unobserved preferences of an active user based on a linear weighted combination of other people's preference" [173].

7.3.2 Recommender Systems and Bayesian Networks

Various models are used to represent the underlying data within *recommender systems*. For example, Shani [180] uses a *Markov Decision Process* (MDP) and Zukerman [181] explores various predictive statistical models such as *linear models*, *TF-IDF*-based models (*Term Frequency - Inverse Document Frequency*), *Markov models*, *Neural networks*, various *n-dimensional classification models*, *Rule-based models* and *Bayesian networks*. Condliff [182] also chooses *Bayesian network*-based models.

One possibility is to use inference on *Bayesian networks* to generate predictions. Those predictions can be recommended to the user, ordered by the most probable to the least probable. Usually, just a few choices are sufficient, but the system can offer the possibility to the user to obtain more recommendations. Based on the variables used during the inference, the recommendation can be accompanied by a justification or explanation of the recommendation.

Bayesian networks and various derivations are still increasingly popular in the *artificial intelligence* community [183]. They have been used for a variety of modelling tasks relative to user preferences [184]. *Bayesian networks* are more flexible than most models [181]. They provide a compact representation of any probability distribution and explicitly represent causal relations. They allow predictions to be made about a number of variables. Also, *Bayesian networks* can be extended to include temporal information as is shown by Dean and Wellman [185], who are using dynamic *Bayesian networks*. Howard and Matheson add influence diagrams to the model to reinforce it [186], [187].

Zuckerman [181] reviews *Bayesian network*-based models that have been used to perform a variety of predictive tasks. For example, Horvitz et al. [188] used a *Bayesian network* to predict the type of assistance required by users performing spreadsheet tasks, Lau and Horvitz [189] built a *Bayesian network* that models search queries and predicts the type of query-related action a user will perform

next (generalise or specify a query). Albrecht et al. [190] used *dynamic Bayesian networks* to predict a user's next action, next location and current quest in a multi-user adventure game. Horvitz et al. [191] also used a *dynamic Bayesian network* to predict a user's attention and the interval between inspections of their email. These predictions allowed the design of a system able to decide whether to alert the user about incoming email and through which means. Gmytrasiewicz et al.'s system [192] considered various models that predict an agent's actions in an air defence scenario, and incrementally updated the probability assigned to each model according to its predictive accuracy. The system built by Jameson et al. [193] predicted the error rates of users when following instructions given in various styles. Some of those examples are not considered *recommender systems* but provide a similar service to a meta-system instead of a user: the modelling task is thought to be a similar challenge.

7.4 Rig Scheduling

Another example of an application for the models is oil drilling rig scheduling. To assist a user or software in scheduling rigs, the application would need to estimate a range of expected completion times for a coordinated set of rig operations parameters. Using the *Bayesian network* to provide the expectations of the various task times would enable such an application. *Bayesian network* for Scheduling has been partially investigated in [194], using multi-agent systems.

7.4.1 Automatic Scheduling in the Industry

The availability of literature relating to the topic of automatic scheduling applied in the oil and gas industry is sparse. However, two mentions have been encountered and used as initial enlightenment of the subject. The first publication is an MBA dissertation by Kenny MacLeod [195]. This paper provides a literature review of the management side of scheduling. The second is a commercial paper about the advantages of Actenum scheduling tool [196], [197]. This publication is interesting for the analysis and insight it provides of the application of novel scheduling technology into a field that has a long history of manual scheduling and difficult constraints such as timing constraints, sequencing constraints and environmental constraints (weather, geology). Actenum [196] states that "production division of an oil company might need to prepare a schedule that satisfies the following requirements: Rig A (a resource) is assigned to drill at Well 31 (a task) for a three month period (a timing constraint), but must be moved to drill at Well 68 by November 7 (a sequencing constraint)". This is a good definition of the problem I am trying to solve. MacLeod [195] refers to Proud (1994) [198] when stating that "resources with limited capacity are the controlling variable requiring schedulers to focus on process bottlenecks to maximise efficiency and minimise

backlog." Actenum confirms that planned and unplanned changes in production operations are a cause of concern to scheduling staff. They add that "linking schedules to key production metrics" enables schedulers to "make informed decisions".

MacLeod [195] identifies 5 major processes of time management: activity definition, activity sequencing, activity duration estimation, schedule development, schedule control. He suggests that the method of capacity planning from Wallace [199] "allows to manage priorities in capacity restricted organisation". Other methods, as the critical path analysis proposed by Kerzner [200] allow to prevent bottlenecks before clashes or over-lapping requirements occur during periods of change. MacLeod [195] explains some requirements and result from these scheduling theories. Actenum adds additional information about the consequences of improved scheduling and how it could apply to an organisation. Most requirements revolve around enhancing communication, training personnel, trust and engagement in accepting the change. Actenum lists a number of advantages. Some of them are enhanced reactivity to change, improved production output ("more reliable and predictable" [196]) and enhanced level of information included in the decision process.

7.4.2 Scheduling Technologies

The study of *scheduling* dates back to the 1950s when researchers and industrial managers were to schedule activities in workshops. *Scheduling* is defined as the "problem of the allocation of resources over time to perform a set of tasks" [201]. In the late 1960s, computer scientists encountered *scheduling* problems when developing operating systems. The scarcity of the resources provided an economic argument to realise research in *scheduling*. Many different approaches have been tried. Most of them were based on branch-and-bound methods and created exponential time consuming algorithms when the complexity of the problem increased. Later, *stochastic scheduling* has been considered. "The field of *stochastic scheduling* is motivated by the design and operational problems arising in systems where scarce service resources must be allocated over time" (Niño-Mora) [202]. *Scheduling* is a combinatorial optimisation problem. Its complexity makes *genetic algorithms* suitable for the *scheduling* problem resolution, unlike deterministic approaches. However, some studies show that *genetic algorithms* are not well suited to the fine tuning of sub-optimal solutions [203,204]. More information on *scheduling* with *genetic algorithms* can be found in research on the "Jobshop scheduling problem" by Mesghouni and Hammadi [205], Hindi et al. [206], Garrido et al. [207] and Kim [208]. In addition, Zhang and Gen [209] consider the resource allocation problem in more details.

Chapter Summary: **In this seventh chapter, I reviewed the application of *computational intelligence* as a competitive tool for commercial development. Following that, I introduced a user interface that was developed to demonstrate the models in the case of developing a rig performance forecasting tool. Finally, the chapter provided a review of *recommender systems*, investigated as a potential application of the models and a review of scheduling technologies.**

Chapter 8: Conclusions

In this final chapter, I review the work outlined in this thesis (section 8.1). I then list the items proposed for future work (section 8.2). Finally, I conclude and list main contributions made by this thesis (section 8.3).

8.1 Summary of Chapter Conclusions

The first chapter introduces the overall research target for this work: investigating the use of statistical tools and Bayesian networks learning algorithms and developing a novel operational tool for forecasting success of rig operations. The chapter exposes the objective and motivation for this work, exposes the research questions and approaches ethical considerations of technological developments. Finally, the chapter lists earlier publications produced as part of this research and provides an overview of the thesis organisation.

The second chapter introduced the commercial background of offshore oil drilling rigs. It reviewed the basics of the offshore drilling background and the rig tendering process used by the industry to select oil drilling rigs. The problems at hand are exposed by reviewing a list of scenarios provided by industry experts in order to guide the progress of this research. The chapter then provided a review of the *Gulf of Mexico* dataset including the available data, the data selection and the work done to prepare the data.

The third chapter provided a review of the state-of-the art techniques for data modelling using Bayesian networks. The focus of this work is centred on search and score methods using the *K2* scoring algorithm. Nature inspired and evolutionary algorithms are explored, with a specific focus on genetic algorithms and ant colony optimisation. In order to provide a benchmark, the standard *Logistic* regression algorithm is also approached in this chapter as well as the metrics of quality used to assess the results.

The fourth chapter introduced the *WRD1* dataset as well as the *K2*-based *genetic* and *ant colony optimisation* algorithms. The experimental results are analysed by reviewing the performances of each algorithms and then by considering expert's evaluation of the model's structures. This chapter also introduces the *node juxtaposition analysis* for viewing the frequency of nodes selected from the search and score approach. For the following parts of this research, even though *ant colony optimisation*-based algorithms performed faster than *genetic algorithm*-based algorithms, *K2GA* was chosen for the further experiments as it provides a higher quality of models and, hence, has a better chance at forecasting oil drilling rig performance accurately.

The fifth chapter introduced the second dataset used in this research as well as its variations: *WRD2.x*. The building of the dataset was a big part of the work done for this section of the research and was described in details, including descriptions of the target field to forecast – *average performance footage per day* – and the methods used for categorising the data. After optimising for algorithms parameters, the chapter then reviewed the results for the *K2GA* runs and analysed the relationships identified by the *Bayesian networks* models in the data.

The sixth chapter focused on the validation of the results and the comparison to the benchmark algorithm. Comparing the results from lower scored orderings with the forecasting abilities of the models, based on a standard *10-fold cross-validation test*, I hypothesise that *CH-score* might not be the best measure of adequacy for determining the worth of a network when considering the dataset. Comparing the forecasting abilities provided by alternative modelling techniques showed that other standard algorithms such as *Logistic*, *J48* and *DecisionTable* have the ability to perform similarly well to *Bayesian networks* on the dataset. Comparing the forecast results to the alternative simplified forecasting technique mimicking an expert's forecast showed that a data-modelling approach can provide better results than the approach devised by the industry experts. The study of the covariance of the landscape of random orderings' *CH-scores* calculated with *K2*, *Chain*, *Pyramid*, *BlockH* and *BlockV* showed that none of the scores for each algorithm are highly correlated with each other, meaning that they might not be good predictors of each other but that more work is necessary to show if those new scoring mechanisms would be a good predictor of the *CH-score*. The feature selection experiment showed that *Logistic* and *Bayesian networks* perform similarly to less than 0.5% accuracy difference but that *Bayesian networks*' learning time is more than 400 times less expensive to process and that the models could potentially be simplified and that it might improve their accuracy slightly.

In the seventh chapter, I reviewed the application of *computational intelligence* as a competitive tool for commercial development. Following that, I introduced a user interface that was developed to demonstrate the models in the case of developing a rig performance forecasting tool. Finally, the chapter provided a review of *recommender systems*, investigated as a potential application of the models and a review of *scheduling* technologies.

8.2 Future Work

This section lists the opportunities for further work which have been identified during this research.

a) Data Linking

Section 2.4.3 emphasised that, when linking data, some cases were found where more than one match was possible. In a future iteration and for production purpose, a manual check of the data might be possible by a trained data expert as the number of data rows remaining to match is limited (634 data rows with matching conflict remaining at the time of writing). However, this remains a tedious task and it might be possible to develop the heuristic to find more links.

b) Problem-dependent Algorithm Performance

Regarding the performance of the algorithms in section 4.4.1, Table 6 confirmed that *K2GA*, *K2ACO* and *ChainACO* were much closer to each other, in terms of scoring, than *ChainGA*. As discussed by Kabli et al. in [24], the performance of *ChainGA* relating to *K2GA* appears to be highly problem-dependent. As confirmed by [26], I expect that the performance of *K2ACO* and *ChainACO* will also be problem-dependent, however, this is to be explored in more detail.

c) WRD2.5: Overlapping Categories

Section 5.1.2 highlighted the possibility of defining overlapping categories. The *WRD2.5* dataset has a different number of categories (10 categories) and, hence, cannot be directly compared in its prediction accuracy for the purpose of this empirical study of the categories of performance. The accuracy of models built with *WRD2.5* in predicting the exact rig performance category is lower than with other *WRD 2.x* but as the granularity of the information is smaller, I found it provides more useful information to the user. Table 11 is an example of how the categories could be combined when presenting the forecast results to the user in order to maintain a higher level of accuracy than currently presented for *WRD2.5* in Table 10. More experiments should be conducted on this in order to measure the accuracy of the information provided to the user satisfactorily.

d) CH-score as a Measure of Bayesian Network Predictive Accuracy

In section 6.1, I ran a comparative analysis of the accuracy of the models generated using a 10-fold cross-validation and the *CH-scores* from *K2*. Figure 22 shows the scores and the accuracy from each of the best node orderings from *experiment 1* & *experiment 2* from chapter 5. The correlation coefficient between the two curves is -0.0228. I propose the hypothesis that *CH-score* might not be

the best measure of adequacy for determining the worth of a network when considering the dataset. However, no better alternative has been identified at this time and this remains a problem open for further research.

e) Alternative Modelling Techniques

In section 6.2, using Weka data mining tool, I tested widespread algorithms on the *WRD2.0* dataset. The results of the testing for all the algorithms are displayed in Table 19 and in Table 20. As shown, the Logistic regression algorithm and the *Bayesian network* learning algorithm are consistently performing better than most other algorithms. Two other algorithms are performing well on the datasets: *J48* and *DecisionTable*. They could be investigated in further work.

f) Study of Fitness Landscape Covariance from Random Orderings Using Fixed Structures and CH-Score

In section 6.4, when studying the fitness landscapes representing the space of *CH-score* node orderings, I am comparing the fitness landscape from the fixed structure *CH-score* to the *K2* learned *CH-score*. The aim of this experiment is to identify if any structure is a natural alternative landscape, capable of providing scores easier to calculate than when using a full *CH-score* which requires learning the entire network each time it is used. Table 22, Table 24 and Table 26 show the statistical analyses of the scores for datasets of random orderings. I observe that none of the scores are highly correlated with each other, meaning that they might not be good predictors of each other. However, I cannot infer that they would be a bad predictor of the final model's performance as the *CH-score* itself is a mechanism to approximate the value of a structure in regard to the modelling exercise but it is not designed to predict the model's performance when in use. This should be determined in a future study.

g) Accuracy, Complexity and Model Learning Time

In section 6.5.4, regarding the feature selection and the exclusion of variables from the model-building, there is a slight increase in the prediction accuracy of both *Logistic* and *BayesNet* when removing variables. This is similar to observations by Devaney et al. [165] and Janecek et al. [166]. This suggests that the models could be simplified and that it might improve their accuracy slightly. More experiments should be conducted to validate that assumption. Reducing the number of variables used in building the model reduces its complexity and allows for faster calculations but, in the case of the *Bayesian networks*, reducing the number of variables might also reduce its versatility and its ability to cope with uncertainty and unknown values.

8.3 Conclusion and Summary of Contributions

Overall, in this work I explored the use of statistical tools and especially *Bayesian networks* learning algorithms applied to a new domain of application (oil drilling rig operational data) in order to forecast performance (average feet drilled per day). I demonstrated a basic tool based on the developed models to industry experts who approved of the forecast delivered. The insights into the data inter-dependencies provided by my research have provided new heuristics to improve the quality of the data in the commercial databases of the project's partners.

One of the main strengths of this research is its application to a real-world problem. I have, in my work, highlighted all the steps taken to process the data clearly and I have explained the reasoning behind each decision. This creates the possibility to apply the technology to other problems in the industry. A similar approach can be used on multiple subjects with a wide range of data. This use of theoretical research to improve techniques and methods applied in the real world contributes to the global and continuous optimisation the industry needs on a wide range of issues and limitations in order to maintain a rate of progress, improvement and development.

In addition to this approach to the real-world problem that I investigated, I also contributed to the body of knowledge on the application of *computational intelligence* algorithms and, more particularly, to the developments of *Bayesian networks* classifiers using *genetic algorithms*.

The following is a summary of the contributions made over the course of this research.

- **Applied K2GA and *Bayesian networks* to a large industry problem:** All through this thesis, I explored how to learn *Bayesian networks* using *K2GA* and to apply this technology to the real-world industry problem at hand. I developed a well-performing and adaptive solution to forecast oil drilling rig performance [13,123,129].

- **Used the knowledge from industry experts to guide the creation of competitive models:** In chapter 5 and with further analysis in chapter 6, I created models able to forecast oil drilling rig performance consistently with close to 80% forecast accuracy, using either *Logistic* regression or *Bayesian network* learning using *genetic algorithms* (K2GA). This also allowed identifying some of the desirable factors necessary to forecast the oil drilling rig performance such as *water depth*, *total footage drilled*, the number of *days to total depth* and the number of *previous wells in the same block*.

- **Introduced node juxtaposition analysis for ordering frequency visualisation:** Based on my publication [129], I introduced the node juxtaposition analysis graph that allows the visualisation of the frequency of nodes links appearing in a set of orderings. This visualisation method provides new insights when analysing node ordering landscapes and is used in chapter 4 and chapter 5.

- **CH-score provides limited information on accuracy:** In chapter 6, I explored the correlation factors between model score and model predictive accuracy and showed that the model score does not correlate with the predictive accuracy of the model, when using the Pearson product-moment correlation coefficient measure.

- **Explored a method for feature selection using multiple algorithms:** In chapter 6, I used feature selection algorithms to attempt reducing the number of nodes and simplify the model. I showed that within limits and with specific algorithms, the model can be simplified with no loss of predictive accuracy. As expected, the reduction of the number of input variables reduces drastically the modelling time by multiple factors.

- **New fixed structure *Bayesian network* learning algorithms:** In chapter 6, I introduced new fixed structure network learning algorithms (*Pyramid*, *BlockV* and *BlockH*) following the idea behind *ChainGA*. The initial tests show that the new structures provide results which do not correlate with *K2GA*; however, *ChainGA* scores do not correlate with *K2GA* scores either. I conclude that additional experiments involving *evolutionary algorithms* should be performed to further the idea.

- **Proposed real-world applications of the models developed, based on current industry needs:** In chapter 8, I reviewed and proposed real-world applications for the models such as *recommender systems*, an oil drilling rig selection tool, a user-ready rig performance forecasting software and rig scheduling tools [129].

Relating to my original objectives, I have investigated the use of *Bayesian networks* learning algorithms and compared it to other statistical and machine learning methods. I approached the model building exercise from real-world data from oil drilling rigs, wells, deployments and specifications and created well-performing models. Furthermore, I investigated various model usages for the oil and gas industry and I reviewed the techniques for applications of *Bayesian* models in commercial and research fields such as forecasting, scheduling and *recommender systems*.

In general, this research project has developed the use of *Bayesian networks* in the real-world for application to commercial objectives and reduced the uncertainty around oil drilling rig performance for the oil and gas industry. This research was carried out using an empirical approach with real-world data from the industry. The data sourcing and preparation was one of the most expensive and time-consuming activity of this research. Using state-of-the-art probabilistic model-building algorithms, we have created an approach to developing models forecasting the performance of industry operations.

I am convinced that these results present a great potential for a better understanding of the variables, influencing oil drilling rig performance, ultimately offering a solution to the industry for improving the selection and operation of oil drilling rigs. Additionally, I am persuaded that the wider range of data analysis and the analytics it supports will play an important part and provide a solid foundation for the development of most technological landscapes in the next five to ten years.

References

1. Knowledge Transfer Partnerships. Knowledge Transfer Partnerships (KTP) - Accelerating Business Innovation [Internet]. 2008 [cited 2012 Aug 28]. Available from: http://www.ktponline.org.uk/

2. Society of Naval Architects and Marine Engineers Singapore, The Joint Branch of the RINA, the IMarEST (Singapore) C for OR& E (CORE). Evening Lecture Invitation. 2008.

3. Wittig F, Jameson A. Exploiting qualitative knowledge in the learning of conditional probabilities of Bayesian networks. Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference. 2000.

4. Markham IS, Mathieu RG, Wray BA. Setting through artificial intelligence: a comparative study of artificial neural networks and decision trees. Integrated Manufacturing Systems. 2000;11(4):239–46.

5. Feelders A, Daniels H, Holsheimer M. Methodological and practical aspects of data mining. Information & Management. 2000;37(5):271–81.

6. Sterritt R, Liu W. Constructing bayesian belief networks for fault management in telecommunications systems. 1st Europen Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems. 2001;149–54.

7. Cummings M. Automation bias in intelligent time critical decision support systems. AIAA 1st Intelligent Systems Technical Conference. Citeseer; 2004. p. 33–40.

8. Bergemann D, Ozmen D. Efficient recommender systems. papers.ssrn.com. 2007;(1196).

9. Horowitz D. Damon Horowitz calls for a “moral operating system” TED talk [Internet]. TED talk. 2011 [cited 2012 Aug 2]. Available from: http://www.ted.com/talks/damon_horowitz.html

10. ODS-Petrodata Ltd. http://rigpoint.ods-petrodata.com/ [Internet]. 2010. Available from: http://rigpoint2.ods-petrodata.com/

11. Harris J. Selection an offshore drilling rig - The competitive tendering process. Offshore Europe. 1989;

12. Nergaard A. Offshore drilling technology. Offshore (Conroe, TX). Carita, Banten, Indonesia: University of Stavanger and Smedvig offshore; 2005. p. 1–17.

13. Fournier FA, Mccall J, Petrovski A, Barclay PJ. Evolved Bayesian network models of rig operations in the Gulf of Mexico. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2010). 2010.

14. Freudenrich C. How oil drilling works [Internet]. howstuffworks.com. 2001 [cited 2011 Mar 22]. p. 1–7. Available from: http://www.howstuffworks.com/oil-drilling.htm

15. National Commission on the BP Deepwater Horizon Oil Spill and Offshore Drilling. A brief history of offshore oil drilling. National Commission on the BP Deepwater Horizon Oil Spill and Offshore Drilling; 2010. p. 18.

16. National Commission on the Deepwater Horizon Oil Spill and Offshore Drilling. National commission on the Deepwater Horizon oil spill and offshore drilling. 2010.

17. Osmundsen P, Sørenes T, Toft A. Drilling contracts and incentives. Energy Policy. Elsevier; 2008;36(8):3128–34.

18. Hoffmann GA, Salfner F, Malek M. Advanced failure prediction in complex software systems. Proc. of SRDS. Citeseer; 2004. p. 1–19.

19. Fayyad U, Piatetsky-shapiro G, Smyth P. The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM. ACM New York, NY, USA; 1996;39(11):27–34.

20. Fayyad U, Piatetsky-shapiro G, Smyth P. The KDD process for extracting useful knowledge from volumes of data. Proceeding of the AAAI 96 conference. ACM New York, NY, USA; 1996;1277–84.

21. Neapolitan RE. Learning bayesian networks. Prentice Hall, Upper Saddle River, NJ. Prentice Hall Upper Saddle River, NJ; 2003.

22. Department of Computer Science Iowa State University. ComS 472/572 Principles of artificial intelligence: Bayesian networks (syntax, semantics, and modeling). 2011.

23. Kjærulff UB, Madsen AL. Probabilistic networks-an introduction to bayesian networks and influence diagrams. Aalborg University. 2005;(May).

24. Kabli R, Herrmann F, Mccall J. A chain-model genetic algorithm for Bayesian network. Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM; 2007. p. 1264–71.

25. Kabli R, McCall J, Herrmann F, Ong E. Evolved bayesian networks as a versatile alternative to partin tables for prostate cancer management. Proceedings of the 10th annual conference on Genetic and evolutionary computation. ACM; 2008. p. 1547–54.

26. Wu Y, Mccall J, Corne D. Two novel ant colony optimization approaches for Bayesian network structure learning. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2010). 2010.

27. Kordon A. Applying computational intelligence: how to create value. Springer; First Edition edition; 2009.

28. Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Machine Learning. 1992 Oct;9(4):309–47.

29. Burgard W, Raedt L De, Kersting K, Nebel B. Bayesian networks, intro. Graphical Models. Freiburg, Germany: Albert-Ludwigs University; 2001.

30. Doguc O, Ramirezmarquez J. A generic method for estimating system reliability using Bayesian networks. Reliability Engineering & System Safety. 2009;94(2):542–50.

31. Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning. Redmond; 1995;20(3):197–243.

32. Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Machine learning. Springer; 1997;29(2):131–63.

33. Adomavicius G, Tuzhilin A. Recommendation technologies: Survey of current methods and possible extensions. Stern School of Business, New York University; 2004. p. 1–39.

34. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering. 2005 Jun;17(6):734–49.

35. Adomavicius G, Tuzhilin A. Expert-driven validation of rule-based user models in personalization applications. Data Mining and Knowledge Discovery. Springer; 2001;5(1):33–58.

36. Thiele L, Miettinen K, Korhonen PJ, Molina J. A preference-based evolutionary algorithm for multi-objective optimization. Evolutionary computation. 2009 Jan;17(3):411–36.

37. Niedermayer D. An introduction to bayesian networks and their contemporary applications. http://www.niedermayer.ca/papers/bayesian/index.html. 1998;

38. Jensen F V., Nielsen TD. Bayesian networks and decision graphs. 2nd editio. Springer; 2007. p. 447.

39. Davies S, Moore AW. Bayesian networks : independencies and inference [Internet]. Carnegie Mellon tutorials. Pittsburgh: Carnegie Mellon; 2008 [cited 2012 Sep 15]. p. 1–21. Available from: http://www.autonlab.org/tutorials/bayesinf.html

40. Robinson RW. Counting unlabeled acyclic digraphs. Combinatorial mathematics V: proceedings of the Fifth Australian Conference. Melbourne; 1977. p. 28.

41. Wong ML, Lee SY, Leung KS. A hybrid data mining approach to discover Bayesian networks using evolutionary programming. Proceedings of the Genetic and Evolutionary Computation Conference. 2002. p. 214–22.

42. Habrant J. Structure learning of Bayesian networks from databases by genetic algorithms-application to time series prediction in finance. Proceedings of the 1st International Conference on Enterprise Information Systems. 1999. p. 225–31.

43. Larrañaga P, Kuijpers CMH, Murga RH, Yurramendi Y. Learning Bayesian network structures by searching for the bestordering with genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, Part A. 1996;26(4):487–93.

44. Novobilski AJ. The random selection and manipulation of legally encoded Bayesian networks in genetic algorithms. The 2003 International Conference on Artificial Intelligence (ICAI). 2003.

45. Van Dijk S, Thierens D, Van Der Gaag L. Building a GA from design principles for learning Bayesian networks. Genetic and Evolutionary Computation — GECCO 2003. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003. p. 198.

46. Buntine WL. Operations for learning with graphical models. Journal of Artificial intelligence Research 2. 1994 Nov;2:159–225.

47. Chickering DM, Geiger D, Heckerman D. Learning Bayesian networks: search methods and experimental results. Preliminary Papers of the Fifth International Workshop on Artificial Intelligence and Statistics. 1995. p. 112–28.

48. Bouckaert RR. Bayesian belief networks: from construction to inference. PhDthesis, University Utrecht. 1995;

49. Murphy K. A brief introduction to graphical models and Bayesian networks [Internet]. 1998 [cited 2012 Aug 2]. Available from: http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html

50. Judea Pearl. Causality: models, reasoning, and inference. Causality: Models, Reasoning, and Inference. Cambridge University Press; 2000. p. 400.

51. DanLi, Yang H, Liang X. Application of Bayesian networks for diagnosis analysis of modified sequencing batch reactor. Advanced Materials Research. 2012;Progress i(610-613):1139–45.

52. Millán E, Descalço L, Castillo G, Oliveira P, Diogo S. Using Bayesian networks to improve knowledge assessment. Computers & Education. 2013;60(1):436–47.

53. Shin C, Hong J-H, Dey AK. Understanding and prediction of mobile application usage for smart phones. Proceedings of the 2012 ACM Conference on Ubiquitous Computing. 2012. p. 173–82.

54. Rigaux C, Ancelet S, Carlin F, Nguyen-thé C, Albert I. Inferring an augmented Bayesian network to confront a complex quantitative microbial risk assessment model with durability studies: application to Bacillus Cereus on a courgette purée production chain. Risk Analysis. 2012;

55. Liang W, Zhuang D, Jiang D, Pan J, Ren H. Assessment of debris flow hazards using a Bayesian Network. Geomorphology. 2012;171-172:94–100.

56. Wiegerinck WAJJ, Kappen B, Burgers W. Bayesian Networks for expert systems, theory and practical applications. In: Babuska R, Groen F, editors. Interactive Collaborative Information Systems. Springer; 2009.

57. Kannan PR. Bayesian networks: application in safety instrumentation and risk reduction. ISA transactions. Elsevier; 2007;46(2):255–9.

58. Martinelli G, Eidsvik J, Hauge R, Førland MD. Bayesian networks for prospect analysis in the North Sea. AAPG bulletin. American Association of Petroleum Geologists (AAPG); 2011;95(8):1423–42.

59. Abdollahzadeh A, Reynolds A, Christie M, Corne D, Davies B, Williams G. Bayesian optimization algorithm applied to uncertainty quantification. SPE Journal. Society of Petroleum Engineers; 2012;17(3):865–73.

60. Masoudi P, Tokhmechi B, Jafari MA, Moshiri B. Application of fuzzy classifier fusion in determining productive zones in oil wells. Energy, Exploration & Exploitation. Multi-Science; 2012;30(3):403–16.

61. Cooper HM. The structure of knowledge synthesis. Knowledge in Society. 1988;1:104–26.

62. Duespohl M, Frank S, Doell P. A Review of Bayesian Networks as a Participatory Modeling Approach in Support of Sustainable Environmental Management. Journal of Sustainable Development. 2012 Nov 8;5(12).

63. Chickering DM, Heckerman D, Meek C. Large-sample learning of Bayesian networks is NP-Hard. The Journal of Machine Learning Research. 2004;

64. Spirtes P, Glymour CN, Scheines R. Causation, prediction, and search. The MIT Press; 2001.

65. De Campos LM, Huete JF. A new approach for learning Bayesian networks using independence criteria. International Journal of Approximate Reasoning. 2000;24:11–37.

66. De Campos LM, Huete JF. On the use of independence relationships for learning simplified belief networks. International Journal of Intelligent Systems. Citeseer; 1997;12(7):495–522.

67. McAuley J, Caetano T, Buntine W. Graphical models. Encyclopedia of Machine Learning. Springer; 2010.

68. Tsamardinos I, Brown LE, Aliferis CF. The max-min hill-climbing Bayesian network structure learning algorithm. Machine learning. Springer; 2006;65(1):31–78.

69. Larrañaga P, Yurramendi Y. Symbolic and quantitative approaches to reasoning and uncertainty. Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Berlin/Heidelberg: Springer-Verlag; 1993. p. 227–32.

70. Larrañaga P, Murga R, Poza M, Kuijpers C. Structure learning of Bayesian networks by hybrid genetic algorithms. Lecture notes in statistics. New York: Springer Verlag KG; 1996. p. 165–74.

71. Singh M, Valtorta M. Construction of Bayesian network structures from data: a brief survey and an efficient algorithm. International Journal of Approximate Reasoning. 1995;12(2):111–32.

72. Larrañaga P, Poza M, Yurramendi Y, Murga RH, Kuijpers CMH. Structure learning of Bayesian networks by genetic algorithms : Performance Analysis of Control Parameters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1996;18(9):912–26.

73. Wang T, Touchman JW, Xue G. Applying two-level simulated annealing on Bayesian structure learning to infer genetic networks. IEEE Computational Systems Bioinformatics Conference. 2004. p. 647–8.

74. Glover F. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research - Special issue: Applications of integer programming. 1986;13(5):533–49.

75. De Campos LM, Gámez Martín JA, Puerta Castellón JM. Learning Bayesian networks by ant colony optimisation: searching in two different spaces. Mathware & soft computing. 2002;9(2-3).

76. De Campos LM, Fernandez-Luna JM, Gámez Martín JA, Puerta Castellón JM. Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning. Elsevier; 2002;31(3):291–311.

77. Cowie J, Oteniya L, Coles R. Particle swarm optimisation for learning Bayesian networks. World Congress on Engineering. 2007. p. 5–10.

78. Sahin F, Devasia A. Distributed particle swarm optimization for structural Bayesian network learning. In: Chan FTS, Tiwari MK, editors. Swarm Intelligence: Focus on Ant and Particle Swarm Optimization. Itech Education and Publishing; 2007. p. 532.

79. Heng X, Qin Z. Research on learning bayesian networks by particle swarm optimization. Information Technology Journal. 2006;5(3):540–5.

80. Correa ES, Freitas A a., Johnson CG. Particle swarm and bayesian networks applied to attribute selection for protein functional classification. Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation. New York, New York, USA: ACM Press; 2007;2651.

81. Larrañaga P, Yurramendi Y. Structure learning approaches in Causal Probabilistics Networks. Symbolic and quantitative approaches to reasoning and uncertainty: European Conference ECSQARU'93, Granada, Spain, November 8-10, 1993: proceedings. 1993. p. 227.

82. Bäck T, Hoffmeister F, Schwefel H-P. A survey of evolution strategies. Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann; 1991. p. 2–9.

83. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Computational Intelligence Magazine. IEEE Computational Inteligence Magazine; 2006;(November):28–39.

84. DeCampos L, Fernandez-Luna J, Gamez J, Puerta J. Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning. 2002;31(3):291–311.

85. Daly R, Shen Q. Learning Bayesian network equivalence classes with ant colony optimization. Artificial Intelligence. AI Access Foundation,; 2009;35:391– 447.

86. Ortiz-Boyer D, Hervás-Martınez C, Garcıa-Pedrajas N. Cixl2: A crossover operator for evolutionary algorithms based on population features. Journal of Artificial Intelligence Research. AI Access Foundation; 2005;24(1):1–48.

87. Myers JW, Laskey KB, DeJong KA. Learning bayesian networks from incomplete data using evolutionary algorithms. Proceedings of the Genetic and Evolutionary Computation Conference. Citeseer; 1999. p. 458–65.

88. Pelikan M, Goldberg DE, Cantu-Paz E. BOA: The Bayesian optimization algorithm. Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99. Citeseer; 1999. p. 525–32.

89. Brownlee A. Multivariate Markov networks for fitness modelling in an estimation of distribution algorithm. Evolutionary Computation. The Robert Gordon University; 2009.

90. Luke S. Essentials of metaheuristics. Lulu; 2009.

91. Robbins H, Monro S. A stochastic approximation method. The Annals of Mathematical Statistics. 1951;400–7.

92. Fermi E, Metropolis N. Los Alamos unclassified report LA-1492. Los Alamos, NM: Los Alamos National Laboratory; 1952.

93. Davidon W. Variable metric method for minimization. SIAM Journal on Optimization. 1991;1(1):1–17.

94. Barricelli NA. Esempi numerici di processi di evoluzione. Methodos. 1954;6:45–68.

95. Michalewicz Z, Fogel DB. How to Solve It: Modern Heuristics. Springer-V. New York; 2004.

96. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press; 1992.

97. Goldberg DE. Genetic algorithms in search, optimization and machine learning. Kluwer Academic Publishers; 1989.

98. Larrañaga P, Kuijpers CMH. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial Intelligence Review. 1999;13:129–70.

99. Eberhart RC, Shi Y. Comparison between genetic algorithms and particle swarm optimization. Lecture Notes in Computer Science. 1998;1447:611–6.

100. Akbari R, Ziarati K. A multilevel evolutionary algorithm for optimizing numerical functions. International Journal of Industrial Engineering Computations. 2011 Apr 1;2(2):419–30.

101. Vassilas N, Miaoulis G, Chronopoulos D, Konstantinidis E, Ravani I, Makris D, et al. MultiCAD-GA: A system for the design of 3D forms based on genetic algorithms and human evaluation. Lecture Notes in Computer Science, Methods and applications of artificial intelligence. 2002;2308:743–4.

102. Ahn L von. Human computation. International Conference On Knowledge Capture. 2007;

103. Dorigo M. Optimization, learning and natural algorithms. Politecnico di Milano, Italy; 1992.

104. Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. actes de la première conférence européenne sur la vie artificielle. Paris, France: Elsevier Publishing; 1991. p. 134–42.

105. Bonabeau E. Editor's introduction: stigmergy. Special issue of Artificial Life on Stigmergy. 1999;5(2):95–6.

106. Grassé PP. La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. Insectes sociaux. Springer; 1959;6(1):41–80.

107. Dorigo M, Socha K. An introduction to ant colony optimization. In: Gonzalez TF, editor. Approximation Algorithms and Metaheuristics. Chapman & Hall/CRC; 2007.

108. Hosmer DW, Hosmer T, Le Cessie S, Lemeshow S. A comparison of goodness-of-fit tests for the logistic regression model. Statistics in medicine. 1997 May 15;16(9):965–80.

109. Cramer JS. The origins of logistic regression. Tinbergen Institute; 2002.

110. Verhulst P-F. Notice sur la loi que la population poursuit dans son accroissement. Garnier JG, Quetelet A, editors. Correspondance mathématique et physique. Impr. d'H. Vandekerckhove; 1838;10:113–21.

111. Verhulst P-F. Recherches mathématiques sur la loi d'accroissement de la population. Nouveaux Mémoires de l'Académie Royale des Sciences et Belles-Lettres de Bruxelles. 1845;18:1–42.

112. Verhulst P-F. Deuxième mémoire sur la loi d'accroissement de la population. Mémoires de l'Académie Royale des Sciences, des Lettres et des Beaux-Arts de Belgique. 1847;20:1–32.

113. Gabriel J-P, Saucy F, Bersier L-F. Paradoxes in the logistic equation? Ecological Modelling. 2005;185(1):147–51.

114. Xu X. Class Logistic. Weka - University of Waikato; 2011.

115. Le Cessie S, Van Houwelingen JC. Ridge estimators in Logistic regression. Applied Statistics. 1992;41(1):191–201.

116. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: An update. SIGKDD Explorations. 2009;11(1).

117. Eibe F. Class ReplaceMissingValueFilter. Weka - University of Waikato; 2011.

118. Carvalho AM. Scoring functions for learning Bayesian networks. Lisboa: INESC-ID; 2009.

119. Buntine W. Theory refinement on Bayesian networks. Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence. 1991. p. 52–60.

120. Hall M. Correlation-based feature selection for machine learning. University of Waikato; 1999.

121. Raykar V, Steck H. On ranking in survival analysis: Bounds on the concordance index. Advances in Neural Information Processing Systems. 2007;20:1209–16.

122. Roy B. The outranking approach and the foundations of ELECTRE methods. Theory and decision. Springer; 1991;31(1):49–73.

123. Fournier FA, Mccall J, Petrovski A, Barclay PJ. Evolved bayesian network models of rig operations in the Gulf of Mexico: preliminary experiments. Aberdeen: SICSA/SEABIS Workshop; 2010.

124. Osmundsen P, Roll KH, Tveterås R. Productivity in exploration drilling. IAEE International Conference. 2009;1–17.

125. Pinto PC, Nagele A, Dejori M, Runkler TA, Sousa J. Learning of Bayesian networks by a local discovery ant colony algorithm. IEEE World Congress on Computational Intelligence. 2008. p. 2741–8.

126. Pinto PC, Nagele A, Dejori M, Runkler TA, Sousa JMC. Using a local discovery ant algorithm for Bayesian network structure learning. IEEE Transactions on Evolutionary Computation. IEEE; 2009;13(4):767–79.

127. Daly R. Using ant colony optimisation in learning bayesian network equivalence classes. Proceedings of the 2006 UK Workshop on Computational Intelligence. 2006;111–8.

128. De Campos L, Puerta J. Stochastic local algorithms for learning belief networks: Searching in the space of the orderings. Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Springer; 2001;228–39.

129. Fournier FA, Wu Y, Mccall J, Petrovski A, Barclay PJ. Application of evolutionary algorithms to learning evolved bayesian network models of rig operations in the Gulf of Mexico. Proceedings of the UKCI Conference. 2010. p. 1–10.

130. García S, Luengo J, Sáez JA, López V, Herrera F. A Survey of discretization techniques: taxonomy and empirical analysis in supervised learning. IEEE Transactions on Knowledge and Data Engineering. IEEE; 2013;25(4):734–50.

131. Fu LD, Tsamardinos I. A comparison of Bayesian network learning algorithms from continuous data. AMIA Annual Symposium Proceedings. 2005. p. 960.

132. Trigg L, Frank E. Class Discretize. Weka - University of Waikato; 2011.

133. Hall M, Rockett P. "Algorithm used by weka.filters.unsupervised.attribute.Discretize?" Forum Post [Internet]. nabble.com. 2010 [cited 2012 Aug 2]. Available from: http://old.nabble.com/Algorithm-used-by-weka.filters.unsupervised.attribute.Discretize--td29525562.html

134. Witten IH, Frank E. Data mining: practical machine learning tools and techniques. Morgan Kaufmann; 2005.

135. Oxford Dictionaries. “cluster” [Internet]. Oxford Dic. Oxford Dictionaries. Oxford University Press.; 2010 [cited 2012 Aug 2]. Available from: http://oxforddictionaries.com/definition/english/cluster

136. Hall M, Frank E. Class EM. Weka; 2011.

137. Borman S. The expectation maximization algorithm a short tutorial. Unpublished paper available at http://www. seanborman. com/publications. Citeseer; 2004;1–9.

138. Bouckaert R. Bayesian network classifiers in Weka. Department of Computer Science, University of …. Hamilton, NZ: University of Waikato; 2004. p. 1–23.

139. Cooper GF, Herskovits E. A Bayesian method for constructing Bayesian belief networks from databases. Proceedings of the Conference on Uncertainty in AI. 1990. p. 86–94.

140. Bouckaert R. Class K2. Weka - University of Waikato; 2011.

141. Wright S. The roles of mutation, inbreeding, crossbreeding and selection in evolution. Proceedings of the sixth international congress on …. 1932;1(6):356–66.

142. Haldane JBS. A mathematical theory of natural and arti cial selection, part viii: Metastable populations. Transactions of the Cambrige Philosophical Society. 1931;17:137–42.

143. Jones T. Evolutionary algorithms, fitness landscapes and search. The University of New Mexico; 1995. p. 249.

144. Hall M, Guetlein M. Class BestFirst. Weka - University of Waikato; 2011.

145. Pearl J. Heuristics: intelligent search strategies for computer problem solving. Addison-Wesley Pub. Co., Inc., Reading, MA; 1984;

146. Hall M. Class GreedyStepwise. Weka - University of Waikato; 2011.

147. Hall M. Class GeneticSearch. Weka. Weka - University of Waikato; 2011.

148. Guetlein M. Class LinearForwardSelection. Weka. Weka - University of Waikato; 2011.

149. Pino A. Class ScatterSearchV1. Weka. Weka - University of Waikato; 2011.

150. Glover F. Heuristics for integer programming using surrogate constraints. Decision Sciences. Wiley Online Library; 1977;8(1):156–66.

151. García López F, García Torres M, Melián Batista B, Moreno Pérez JA, Moreno-Vega JM. Solving feature subset selection problem by a Parallel Scatter Search. European Journal of Operational Research. 2004 Mar;169(2):477–89.

152. Guetlein M. Class SubsetSizeForwardSelection. Weka. Weka - University of Waikato; 2011.

153. Guetlein M, Frank E, Hall M, Karwath A. Large-scale attribute selection using wrappers. Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on. 2009. p. 332–9.

154. Hall M. Class ClassifierSubsetEval. Weka. Weka - University of Waikato; 2011.

155. Hall M. Class WrapperSubsetEval. Weka. Weka - University of Waikato; 2011.

156. Kohavi R, John GH. Wrappers for feature subset selection. Artificial Intelligence. 1997;97(1-2):273–324.

157. Frank E. Class ZeroR. Weka. Weka; 2011.

158. Trigg L, Frank E. Class NaiveBayes. Weka. Weka - University of Waikato; 2011.

159. John GH, Langley P. Estimating continuous distributions in bayesian classifiers. In: Kaufmann M, editor. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. San Mateo; 1995. p. 339–45.

160. Hall M. Class ConsistencySubsetEval. Weka. Weka - University of Waikato; 2011.

161. Liu H, Setiono R. A probabilistic approach to feature selection - A filter solution. 13th International Conference on Machine Learning. 1996. p. 319–27.

162. Guyon I, Elisseeff A. An introduction to variable and feature selection. The Journal of Machine Learning Research. 2003;3:1157–82.

163. Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. Oxford Univ Press; 2007;23(19):2507–17.

164. Massa P, Avesani P. Trust-aware collaborative filtering for recommender systems. On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. Springer; 2004;492–508.

165. Devaney M, Ram A. Efficient feature selection in conceptual clustering. Machine Learning: Proceedings of the Fourteenth International Conference. 1997;1997(July).

166. Janecek A, Gansterer W. On the relationship between feature selection and classification accuracy. JMLR: Workshop and Conference Proceedings. 2008;4:90–105.

167. Wang L, Singh C, Kusiak A, editors. Wind Power Systems: Applications of Computational Intelligence (Green Energy and Technology). Springer; 2010.

168. Saxena D, Singh S., Verma K. Application of computational intelligence in emerging power systems. International Journal of Engineering, Science and Technology. 2010 Sep 7;2(3):1–7.

169. Swain SC, Panda S, Mohanty AK, Ardil C. Application of computational intelligence techniques for economic load dispatch. 2010;497–505.

170. Velez-Langs O. Genetic algorithms in oil industry: An overview. Journal of Petroleum Science and Engineering. 2005;47(1-2):15–22.

171. Tadeusiewicz R. Using neural models for evaluation of biological activity of selected chemical compounds. In: Smolinski T, Milanova M, Hassanien A-E, editors. Applications of Computational Intelligence in Biology. Springer Berlin Heidelberg; 2008. p. 135–59.

172. Park H, Yoo J, Cho S. A context-aware music recommendation system. Lecture Notes in Artificial Intelligence. 2006;4223:970 – 979.

173. Oh H. Literature review on advisor selection [Internet]. University of Minnesota. University of Minnesota; 2009 [cited 2012 Sep 15]. Available from: http://misrc.umn.edu/workshops/2009/spring/Oh.pdf

174. Resnick P, Varian HR. Recommender systems. Communications of the ACM. ACM; 1997;40(3):58.

175. Ujjin S, Bentley P. Building a Lifestyle Recommender System. Poster Proceedings of the 10th International World Wide Web Conference, Hong Kong. Citeseer; 2001. p. 3–7.

176. Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. San Francisco, CA; 1998.

177. Goldberg K, Roeder T, Gupta D, Perkins C. Eigentaste: a constant time collaborative filtering algorithm. Information Retrieval. 2001;(August).

178. Terveen LG, Hill W. Beyond recommender systems: helping people help each other. HCI in the New Millennium. Addison Wesley. Citeseer; 2001;(1):487–509.

179. Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems. ACM; 2004;22(1):5–53.

180. Shani G, Heckerman D, Brafman RI. An MDP-based recommender system. Journal of Machine Learning Research. Citeseer; 2006;6(2):1265.

181. Zukerman I, Albrecht DW. Predictive statistical models for user modeling. User Modeling and User-Adapted Interaction. Springer; 2001;11(1):5–18.

182. Condli MK, Lewis DD, Madigan D, Posse C, Talaria I. Bayesian mixed-effects models for recommender systems. Conference SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation. Citeseer; 1999.

183. Pearl J, Shafer G. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann. Morgan Kaufmann San Mateo, CA; 1988.

184. Jameson A. Numerical uncertainty management in user and student modeling: An overview of systems and issues. User Modeling and User-Adapted Interaction. Springer; 1995;5(3):193–251.

185. Dean TL, Wellman MP. Planning and control. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA; 1991.

186. Howard RA, Matheson JE. Influence diagrams, In the Principles and Applications of Decision Analysis. Strateg Decis Group. 1984;2:719–26.

187. Howard RA, Matheson JE. Influence diagrams. Decision Analysis. INFORMS; 2005;2(3):127–43.

188. Horvitz E, Breese J, Heckerman D, Hovel D, Rommelse K. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. Proceedings of the fourteenth Conference on Uncertainty in Artificial Intelligence. 1998. p. 256–65.

189. Lau T, Horvitz E. Patterns of search: Analyzing and modeling web query refinement. Courses and Lectures - International centre for mechanical sciences. Citeseer; 1999;119–28.

190. Albrecht DW, Zukerman I, Nicholson AE. Bayesian models for keyhole plan recognition in an adventure game. User modeling and user-adapted interaction. Springer; 1998;8(1):5–47.

191. Horvitz E, Jacobs A, Hovel D. Attention-sensitive alerting. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc; 1999.

192. Gmytrasiewicz PJ, Noh S, Kellogg T. Bayesian update of recursive agent models. User Modeling and User-Adapted Interaction. Springer; 1998;8(1):49–69.

193. Jameson A, Großmann-Hutter B, March L, Rummer R. Creating an empirical basis for adaptation decisions. Proceedings of the 5th international conference on Intelligent user interfaces. 2000. p. 149–56.

194. Macho S, Torrens M, Faltings B. A multi-agent recommender system for planning meetings. Fourth International Conference on Autonomous Agents, Workshop on Agent-based Recommender Systems (WARS2000). Citeseer; 2000.

195. MacLeod K. Global vessel scheduling in the subsea engineering & construction industry. Robert Gordon University; 2008.

196. Actenum. Scheduling in asset-intensive organizations : is there a better way ? Vancouver: Actenum Corporation; 2006. p. 1–6.

197. Israel M. Advances in rig scheduling techniques. Upstream Technology. 2008;3(4).

198. Proud JF. Master scheduling: a practical guide to competitive manufacturing. Wiley; 2007.

199. Wallace TF, Kremzar MH. ERP: making it happen: the implementers' guide to success with enterprise resource planning. John Wiley & Sons Inc; 2001.

200. Kerzner H. Project management: a systems approach to planning, scheduling, and controlling. Wiley; 2009.

201. Blazewicz J. Scheduling in computer and manufacturing systems. Springer-Verlag New York, Inc. Secaucus, NJ, USA; 1996.

202. Nino-Mora J. Stochastic scheduling. Encyclopedia of Optimization. 2001;5:367–72.

203. Bierwirth C. A generalized permutation approach to job shop scheduling with genetic algorithms. OR Spectrum. Springer; 1995;17(2):87–92.

204. Dorndorf U, Pesch E. Evolution based learning in a job shop scheduling environment. Computers & Operations Research. Elsevier; 1995;22(1):25–40.

205. Mesghouni K, Hammadi S, Borne P. Evolutionary algorithms for job-shop scheduling. International Journal of Applied Mathematics and Computer Science. Citeseer; 2004. p. 91–104.

206. Hindi KS, Yang H, Fleszar K. An evolutionary algorithm for resource-constrained project scheduling. Evolutionary Computation, IEEE Transactions on. IEEE; 2002;6(5):512–8.

207. Garrido A, Salido M, Barber F, López M. Heuristic methods for solving job-shop scheduling problems. ECAI-2000 Workshop on New Results in Planning, Scheduling and Design. Berl'\in. Citeseer; 2000. p. 36–43.

208. Kim J. Permutation-based elitist genetic algorithm using serial scheme for large-sized resource-constrained project scheduling. Proceedings of the 39th conference on Winter simulation. IEEE Press; 2007. p. 2112–8.

209. Zhang H, Gen M. Effective genetic approach for optimizing advanced planning and scheduling in flexible manufacturing system. Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06. New York, New York, USA: ACM Press; 2006;1841.

210. Coburn P. The change function: why some technologies take off and others crash and burn. A & C Black Publishers Ltd; 2007. p. 224.

211. Christensen CM, Raynor ME. The innovator's solution: Creating and sustaining successful growth. Harvard Business Press; 2003.

212. Levitt TM. The marketing imagination. Expanded. Free Press; 1986. p. 238.

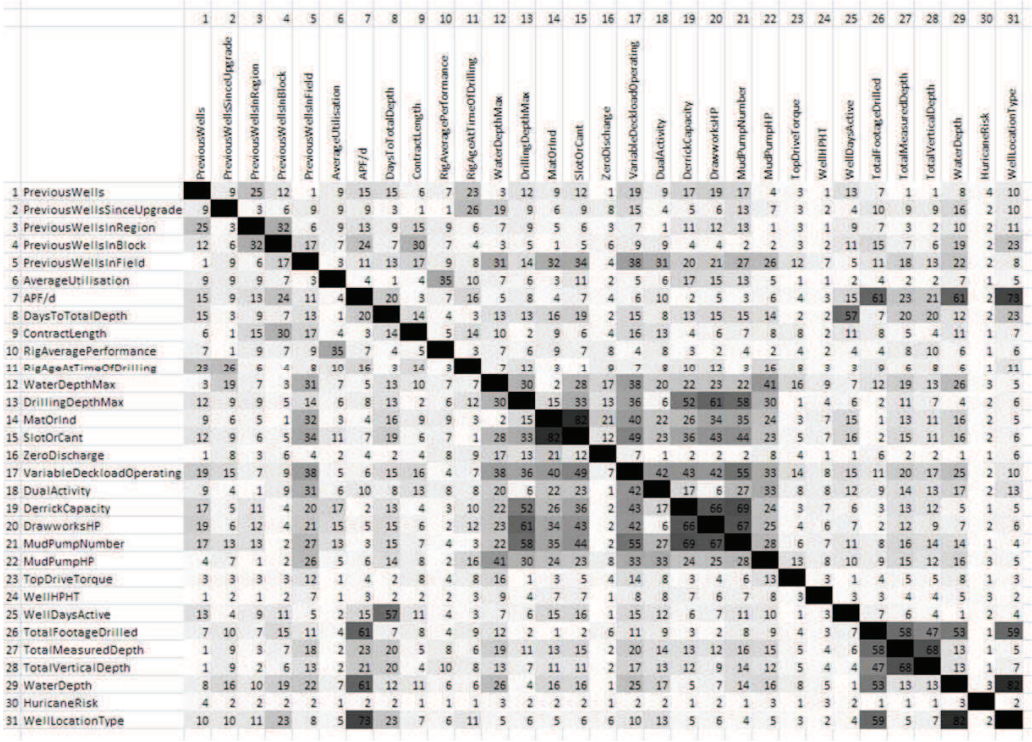
213. Araki M. PID control. Control systems, robotics and automation. 2002;2.

214. McLendon R. Types of offshore oil rigs [Internet]. Mother Nature Network - Earth Matters - Energy. 2010 [cited 2012 Sep 14]. Available from: http://www.mnn.com/earth-matters/energy/stories/types-of-offshore-oil-rigs

215. Quinlan J. C4.5: Programs for machine learning. Morgan Kaufmann Publishers; 1993.

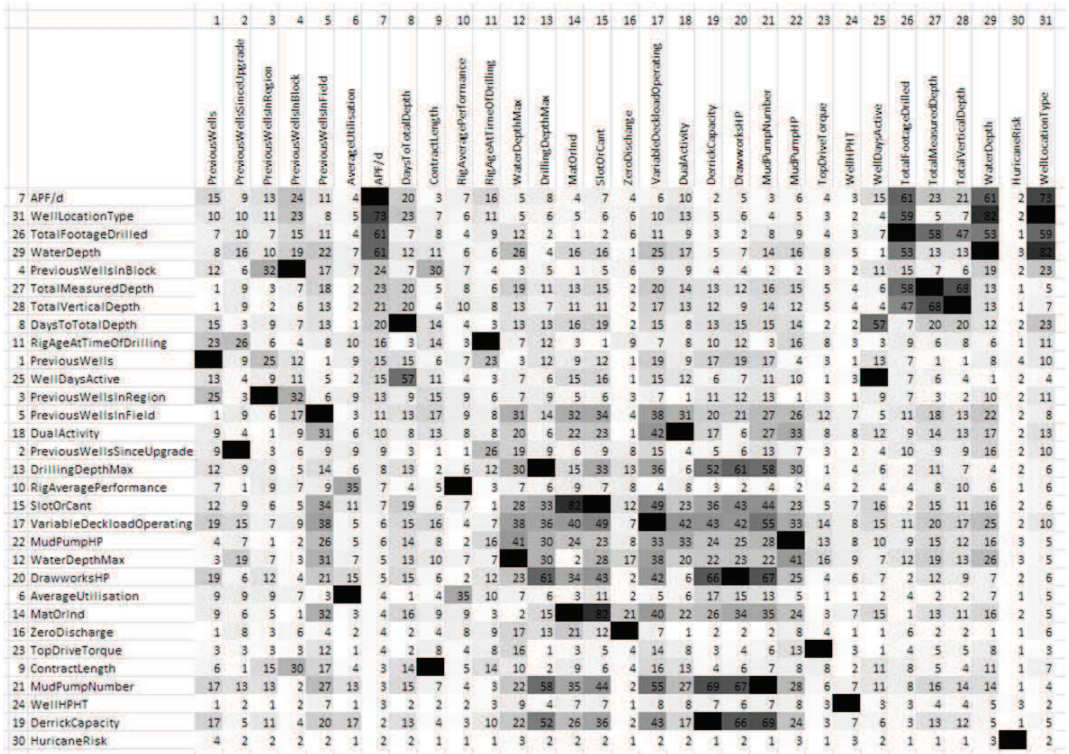
Appendix

Figure 30: Pearson Correlations of all continuous variables for WRD2.0



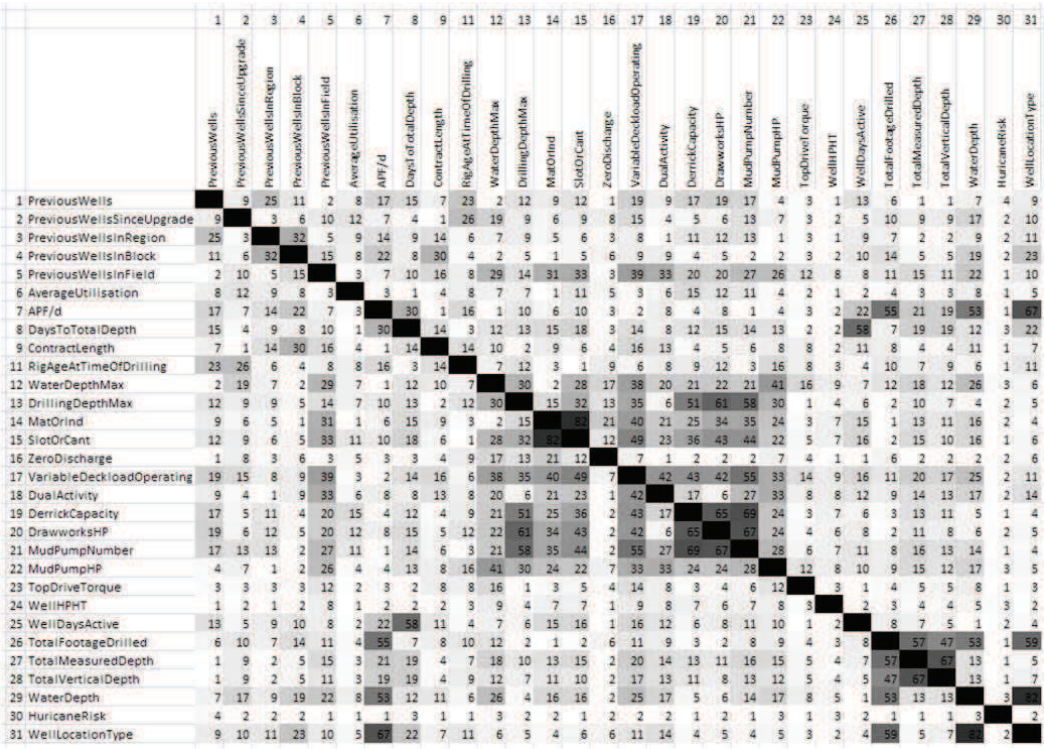
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
		PreviousWells	PreviousWellsSinceUpgrade	PreviousWellsInRegion	PreviousWellsInBlock	PreviousWellsInField	AverageUtilisation	APF/d	DaysToTotalDepth	ContractLength	RigAverageePerformance	RigAgeAtTimeOfDrilling	WaterDepthMax	DrillingDepthMax	MatOrInd	SlotOrCant	ZeroDischarge	VariableDeckloadOperating	DualActivity	DerrickCapacity	DrawworksHP	MudPumpNumber	MudPumpHP	TopDriveTorque	WellHPHT	WellDaysActive	TotalFootageDrilled	TotalMeasuredDepth	TotalVerticalDepth	WaterDepth	HuricaneRisk	WellLocationType
1	PreviousWells		9	25	12	1	9	15	15	6	7	23	3	12	9	12	1	19	9	17	19	17	4	3	1	13	7	1	1	8	4	10
2	PreviousWellsSinceUpgrade	9		3	6	9	9	9	3	1	1	26	19	9	6	9	8	15	4	5	6	13	7	3	2	4	10	9	9	16	2	10
3	PreviousWellsInRegion	25	3		32	6	9	13	9	15	9	6	7	9	5	6	3	7	1	11	12	13	1	3	1	9	7	3	2	10	2	11
4	PreviousWellsInBlock	12	6	32		17	7	24	7	30	7	4	3	5	1	5	6	9	9	4	4	2	2	3	2	11	15	7	6	19	2	23
5	PreviousWellsInField	1	9	6	17		3	11	13	17	9	8	31	14	32	34	4	38	31	20	21	27	26	12	7	5	11	18	13	22	2	8
6	AverageUtilisation	9	9	9	7	3		4	1	4	35	10	7	6	3	11	2	5	6	17	15	13	5	1	1	2	4	2	2	7	1	5
7	APF/d	15	9	13	24	11	4		20	3	7	16	5	8	4	7	4	6	10	2	5	3	6	4	3	15	61	23	21	61	2	73
8	DaysToTotalDepth	15	3	9	7	13	1	20		14	4	3	13	13	16	19	2	15	8	13	15	15	14	2	2	57	7	20	20	12	2	23
9	ContractLength	6	1	15	30	17	4	3	14		5	14	10	2	9	6	4	16	13	4	6	7	8	8	2	11	8	5	4	11	1	7
10	RigAveragePerformance	7	1	9	7	9	35	7	4	5		3	7	6	9	7	8	4	8	3	2	4	2	4	2	4	4	8	10	6	1	6
11	RigAgeAtTimeOfDrilling	23	26	6	4	8	10	16	3	14	3		7	12	3	1	9	7	8	10	12	3	16	8	3	3	9	6	8	6	1	11
12	WaterDepthMax	3	19	7	3	31	7	5	13	10	7	7		30	2	28	17	38	20	22	23	22	41	16	9	7	12	19	13	26	3	5
13	DrillingDepthMax	12	9	9	5	14	6	8	13	2	6	12	30		15	33	13	36	6	52	61	58	30	1	4	6	2	11	7	4	2	6
14	MatOrInd	9	6	5	1	32	3	4	16	9	9	3	2	15		82	21	40	22	26	34	35	24	3	7	15	1	13	11	16	2	5
15	SlotOrCant	12	9	6	5	34	11	7	19	6	7	1	28	33	82		12	49	23	36	43	44	23	5	7	16	2	15	11	16	2	6
16	ZeroDischarge	1	8	3	6	4	2	4	2	4	8	9	17	13	21	12		7	1	2	2	2	8	4	1	1	6	2	2	1	1	6
17	VariableDeckloadOperating	19	15	7	9	38	5	6	15	16	4	7	38	36	40	49	7		42	43	42	55	33	14	8	15	11	20	17	25	2	10
18	DualActivity	9	4	1	9	31	6	10	8	13	8	8	20	6	22	23	1	42		17	6	27	33	8	8	12	9	14	13	17	2	13
19	DerrickCapacity	17	5	11	4	20	17	2	13	4	3	10	22	52	26	36	2	43	17		66	69	24	3	7	6	3	13	12	5	1	5
20	DrawworksHP	19	6	12	4	21	15	5	15	6	2	12	23	61	34	43	2	42	6	66		67	25	4	6	7	2	12	9	7	2	6
21	MudPumpNumber	17	13	13	2	27	13	3	15	7	4	3	22	58	35	44	2	55	27	69	67		28	6	7	11	8	16	14	14	1	4
22	MudPumpHP	4	7	1	2	26	5	6	14	8	2	16	41	30	24	23	8	33	33	24	25	28		13	8	10	9	15	12	16	3	5
23	TopDriveTorque	3	3	3	3	12	1	4	2	8	4	8	16	1	3	5	4	14	8	3	4	6	13		3	1	4	5	5	8	1	3
24	WellHPHT	1	2	1	2	7	1	3	2	2	2	3	9	4	7	7	1	8	8	7	6	7	8	3		3	3	4	4	5	3	2
25	WellDaysActive	13	4	9	11	5	2	15	57	11	4	3	7	6	15	16	1	15	12	6	7	11	10	1	3		7	6	4	1	2	4
26	TotalFootageDrilled	7	10	7	15	11	4	61	7	8	4	9	12	2	1	2	6	11	9	3	2	8	9	4	3	7		58	47	53	1	59
27	TotalMeasuredDepth	1	9	3	7	18	2	23	20	5	8	6	19	11	13	15	2	20	14	13	12	16	15	5	4	6	58		68	13	1	5
28	TotalVerticalDepth	1	9	2	6	13	2	21	20	4	10	8	13	7	11	11	2	17	13	12	9	14	12	5	4	4	47	68		13	1	7
29	WaterDepth	8	16	10	19	22	7	61	12	11	6	6	26	4	16	16	1	25	17	5	7	14	16	8	5	1	53	13	13		3	82
30	HuricaneRisk	4	2	2	2	2	1	2	2	1	1	1	3	2	2	2	1	2	2	1	2	1	3	1	3	2	1	1	1	3		2
31	WellLocationType	10	10	11	23	8	5	73	23	7	6	11	5	6	5	6	6	10	13	5	6	4	5	3	2	4	59	5	7	82	2	

Figure 31: Pearson Correlations of all continuous variables for WRD2.0, ordered by correlation with AveragePerformanceFootagePerDay



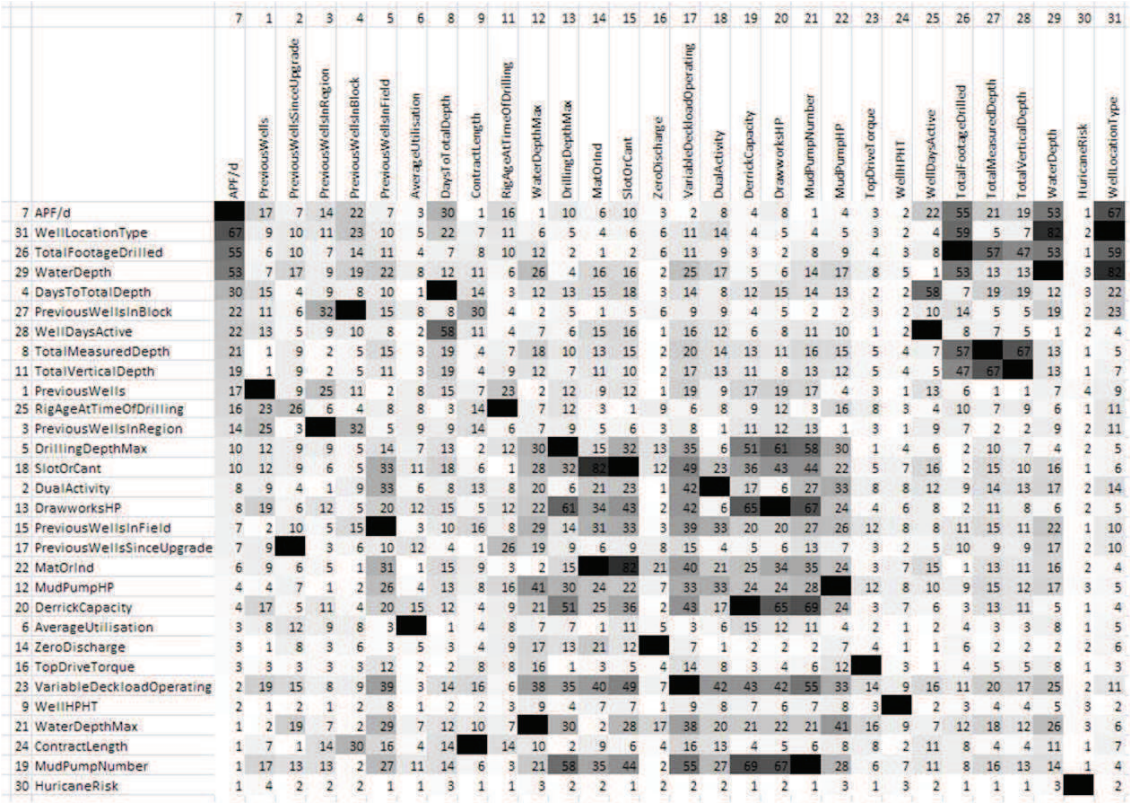
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
		PreviousWells	PreviousWellsSinceUpgrade	PreviousWellsInRegion	PreviousWellsInBlock	PreviousWellsInField	AverageUtilisation	APF/d	DaysToTotalDepth	ContractLength	RigAverageePerformance	RigAgeAtTimeOfDrilling	WaterDepthMax	DrillingDepthMax	MatOrInd	SlotOrCant	ZeroDischarge	VariableDeckloadOperating	DualActivity	DerrickCapacity	DrawworksHP	MudPumpNumber	MudPumpHP	TopDriveTorque	WellHPHT	WellDaysActive	TotalFootageDrilled	TotalMeasuredDepth	TotalVerticalDepth	WaterDepth	HuricaneRisk	WellLocationType
7	APF/d	15	9	13	24	11	4		20	3	7	16	5	8	4	7	4	6	10	2	5	3	6	4	3	15	61	23	21	61	2	73
31	WellLocationType	10	10	11	23	8	5	73	23	7	6	11	5	6	5	6	6	10	13	5	6	4	5	3	2	4	59	5	7	82	2	
26	TotalFootageDrilled	7	10	7	15	11	4	61	7	8	4	9	12	2	1	2	6	11	9	3	2	8	9	4	3	7		58	47	53	1	59
29	WaterDepth	8	16	10	19	22	7	61	12	11	6	6	26	4	16	16	1	25	17	5	7	14	16	8	5	1	53	13	13		3	82
4	PreviousWellsInBlock	12	6	32		17	7	24	7	30	7	4	3	5	1	5	6	9	9	4	4	2	2	3	2	11	15	7	6	19	2	23
27	TotalMeasuredDepth	1	9	3	7	18	2	23	20	5	8	6	19	11	13	15	2	20	14	13	12	16	15	5	4	6	58		68	13	1	5
28	TotalVerticalDepth	1	9	2	6	13	2	21	20	4	10	8	13	7	11	11	2	17	13	12	9	14	12	5	4	4	47	68		13	1	7
8	DaysToTotalDepth	15	3	9	7	13	1	20		14	4	3	13	13	16	19	2	15	8	13	15	15	14	2	2	57	7	20	20	12	2	23
11	RigAgeAtTimeOfDrilling	23	26	6	4	8	10	16	3	14	3		7	12	3	1	9	7	8	10	12	3	16	8	3	3	9	6	8	6	1	11
1	PreviousWells		9	25	12	1	9	15	15	6	7	23	3	12	9	12	1	19	9	17	19	17	4	3	1	13	7	1	1	8	4	10
25	WellDaysActive	13	4	9	11	5	2	15	57	11	4	3	7	6	15	16	1	15	12	6	7	11	10	1	3		7	6	4	1	2	4
3	PreviousWellsInRegion	25	3		32	6	9	13	9	15	9	6	7	9	5	6	3	7	1	11	12	13	1	3	1	9	7	3	2	10	2	11
5	PreviousWellsInField	1	9	6	17		3	11	13	17	9	8	31	14	32	34	4	38	31	20	21	27	26	12	7	5	11	18	13	22	2	8
18	DualActivity	9	4	1	9	31	6	10	8	13	8	8	20	6	22	23	1	42		17	6	27	33	8	8	12	9	14	13	17	2	13
2	PreviousWellsSinceUpgrade	9		3	6	9	9	9	3	1	1	26	19	9	6	9	8	15	4	5	6	13	7	3	2	4	10	9	9	16	2	10
13	DrillingDepthMax	12	9	9	5	14	6	8	13	2	6	12	30		15	33	13	36	6	52	61	58	30	1	4	6	2	11	7	4	2	6
10	RigAveragePerformance	7	1	9	7	9	35	7	4	5		3	7	6	9	7	8	4	8	3	2	4	2	4	2	4	4	8	10	6	1	6
15	SlotOrCant	12	9	6	5	34	11	7	19	6	7	1	28	33	82		12	49	23	36	43	44	23	5	7	16	2	15	11	16	2	6
17	VariableDeckloadOperating	19	15	7	9	38	5	6	15	16	4	7	38	36	40	49	7		42	43	42	55	33	14	8	15	11	20	17	25	2	10
22	MudPumpHP	4	7	1	2	26	5	6	14	8	2	16	41	30	24	23	8	33	33	24	25	28		13	8	10	9	15	12	16	3	5
12	WaterDepthMax	3	19	7	3	31	7	5	13	10	7	7		30	2	28	17	38	20	22	23	22	41	16	9	7	12	19	13	26	3	5
20	DrawworksHP	19	6	12	4	21	15	5	15	6	2	12	23	61	34	43	2	42	6	66		67	25	4	6	7	2	12	9	7	2	6
6	AverageUtilisation	9	9	9	7	3		4	1	4	35	10	7	6	3	11	2	5	6	17	15	13	5	1	1	2	4	2	2	7	1	5
14	MatOrInd	9	6	5	1	32	3	4	16	9	9	3	2	15		82	21	40	22	26	34	35	24	3	7	15	1	13	11	16	2	5
16	ZeroDischarge	1	8	3	6	4	2	4	2	4	8	9	17	13	21	12		7	1	2	2	2	8	4	1	1	6	2	2	1	1	6
23	TopDriveTorque	3	3	3	3	12	1	4	2	8	4	8	16	1	3	5	4	14	8	3	4	6	13		3	1	4	5	5	8	1	3
9	ContractLength	6	1	15	30	17	4	3	14		5	14	10	2	9	6	4	16	13	4	6	7	8	8	2	11	8	5	4	11	1	7
21	MudPumpNumber	17	13	13	2	27	13	3	15	7	4	3	22	58	35	44	2	55	27	69	67		28	6	7	11	8	16	14	14	1	4
24	WellHPHT	1	2	1	2	7	1	3	2	2	2	3	9	4	7	7	1	8	8	7	6	7	8	3		3	3	4	4	5	3	2
19	DerrickCapacity	17	5	11	4	20	17	2	13	4	3	10	22	52	26	36	2	43	17		66	69	24	3	7	6	3	13	12	5	1	5
30	HuricaneRisk	4	2	2	2	2	1	2	2	1	1	1	3	2	2	2	1	2	2	1	2	1	3	1	3	2	1	1	1	3		2

Figure 32: Pearson Correlations of all continuous variables for WRD2.5



	1 PreviousWells	2 PreviousWellsSinceUpgrade	3 PreviousWellsInRegion	4 PreviousWellsInBlock	5 PreviousWellsInField	6 AverageUtilisation	7 APF/d	8 DaysToTotalDepth	9 ContractLength	11 RigAgeAtTimeOfDrilling	12 WaterDepthMax	13 DrillingDepthMax	14 MatOrInd	15 SlotOrCant	16 ZeroDischarge	17 VariableDeckloadOperating	18 DualActivity	19 DerrickCapacity	20 DrawworksHP	21 MudPumpNumber	22 MudPumpHP	23 TopDriveTorque	24 WellHPHT	25 WellDaysActive	26 TotalFootageDrilled	27 TotalMeasuredDepth	28 TotalVerticalDepth	29 WaterDepth	30 HuricaneRisk	31 WellLocationType
1 PreviousWells		9	25	11	2	8	17	15	7	23	2	12	9	12	1	19	9	17	19	17	4	3	1	13	6	1	1	7	4	9
2 PreviousWellsSinceUpgrade	9		3	6	10	12	7	4	1	26	19	9	6	9	8	15	4	5	6	13	7	3	2	5	10	9	9	17	2	10
3 PreviousWellsInRegion	25	3		32	5	9	14	9	14	6	7	9	5	6	3	8	1	11	12	13	1	3	1	9	7	2	2	9	2	11
4 PreviousWellsInBlock	11	6	32		15	8	22	8	30	4	2	5	1	5	6	9	9	4	5	2	2	3	2	10	14	5	5	19	2	23
5 PreviousWellsInField	2	10	5	15		3	7	10	16	8	29	14	31	33	3	39	33	20	20	27	26	12	8	8	11	15	11	22	1	10
6 AverageUtilisation	8	12	9	8	3		3	1	4	8	7	7	1	11	5	3	6	15	12	11	4	2	1	2	4	3	3	8	1	5
7 APF/d	17	7	14	22	7	3		30	1	16	1	10	6	10	3	2	8	4	8	1	4	3	2	22	55	21	19	53	1	67
8 DaysToTotalDepth	15	4	9	8	10	1	30		14	3	12	13	15	18	3	14	8	12	15	14	13	2	2	58	7	19	19	12	3	22
9 ContractLength	7	1	14	30	16	4	1	14		14	10	2	9	6	4	16	13	4	5	6	8	8	2	11	8	4	4	11	1	7
11 RigAgeAtTimeOfDrilling	23	26	6	4	8	8	16	3	14		7	12	3	1	9	6	8	9	12	3	16	8	3	4	10	7	9	6	1	11
12 WaterDepthMax	2	19	7	2	29	7	1	12	10	7		30	2	28	17	38	20	21	22	21	41	16	9	7	12	18	12	26	3	6
13 DrillingDepthMax	12	9	9	5	14	7	10	13	2	12	30		15	32	13	35	6	51	61	58	30	1	4	6	2	10	7	4	2	5
14 MatOrInd	9	6	5	1	31	1	6	15	9	3	2	15		82	21	40	21	25	34	35	24	3	7	15	1	13	11	16	2	4
15 SlotOrCant	12	9	6	5	33	11	10	18	6	1	28	32	82		12	49	23	36	43	44	22	5	7	16	2	15	10	16	1	6
16 ZeroDischarge	1	8	3	6	3	5	3	3	4	9	17	13	21	12		7	1	2	2	2	7	4	1	1	6	2	2	2	2	6
17 VariableDeckloadOperating	19	15	8	9	39	3	2	14	16	6	38	35	40	49	7		42	43	42	55	33	14	9	16	11	20	17	25	2	11
18 DualActivity	9	4	1	9	33	6	8	8	13	8	20	6	21	23	1	42		17	6	27	33	8	8	12	9	14	13	17	2	14
19 DerrickCapacity	17	5	11	4	20	15	4	12	4	9	21	51	25	36	2	43	17		65	69	24	3	7	6	3	13	11	5	1	4
20 DrawworksHP	19	6	12	5	20	12	8	15	5	12	22	61	34	43	2	42	6	65		67	24	4	6	8	2	11	8	6	2	5
21 MudPumpNumber	17	13	13	2	27	11	1	14	6	3	21	58	35	44	2	55	27	69	67		28	6	7	11	8	16	13	14	1	4
22 MudPumpHP	4	7	1	2	26	4	4	13	8	16	41	30	24	22	7	33	33	24	24	28		12	8	10	9	15	12	17	3	5
23 TopDriveTorque	3	3	3	3	12	2	3	2	8	8	16	1	3	5	4	14	8	3	4	6	12		3	1	4	5	5	8	1	3
24 WellHPHT	1	2	1	2	8	1	2	2	2	3	9	4	7	7	1	9	8	7	6	7	8	3		2	3	4	4	5	3	2
25 WellDaysActive	13	5	9	10	8	2	22	58	11	4	7	6	15	16	1	16	12	6	8	11	10	1	2		8	7	5	1	2	4
26 TotalFootageDrilled	6	10	7	14	11	4	55	7	8	10	12	2	1	2	6	11	9	3	2	8	9	4	3	8		57	47	53	1	59
27 TotalMeasuredDepth	1	9	2	5	15	3	21	19	4	7	18	10	13	15	2	20	14	13	11	16	15	5	4	7	57		67	13	1	5
28 TotalVerticalDepth	1	9	2	5	11	3	19	19	4	9	12	7	11	10	2	17	13	11	8	13	12	5	4	5	47	67		13	1	7
29 WaterDepth	7	17	9	19	22	8	53	12	11	6	26	4	16	16	2	25	17	5	6	14	17	8	5	1	53	13	13		3	82
30 HuricaneRisk	4	2	2	2	1	1	1	3	1	1	3	2	2	1	2	2	2	1	2	1	3	1	3	2	1	1	1	3		2
31 WellLocationType	9	10	11	23	10	5	67	22	7	11	6	5	4	6	6	11	14	4	5	4	5	3	2	4	59	5	7	82	2	

Figure 33: Pearson Correlations of all continuous variables for WRD2.5, ordered by correlation with AveragePerformanceFootagePerDay



	7 APF/d	1 PreviousWells	2 PreviousWellsSinceUpgrade	3 PreviousWellsInRegion	4 PreviousWellsInBlock	5 PreviousWellsInField	6 AverageUtilisation	8 DaysToTotalDepth	9 ContractLength	11 RigAgeAtTimeOfDrilling	12 WaterDepthMax	13 DrillingDepthMax	14 MatOrInd	15 SlotOrCant	16 ZeroDischarge	17 VariableDeckloadOperating	18 DualActivity	19 DerrickCapacity	20 DrawworksHP	21 MudPumpNumber	22 MudPumpHP	23 TopDriveTorque	24 WellHPHT	25 WellDaysActive	26 TotalFootageDrilled	27 TotalMeasuredDepth	28 TotalVerticalDepth	29 WaterDepth	30 HuricaneRisk	31 WellLocationType
7 APF/d		17	7	14	22	7	3	30	1	16	1	10	6	10	3	2	8	4	8	1	4	3	2	22	55	21	19	53	1	67
31 WellLocationType	67	9	10	11	23	10	5	22	7	11	6	5	4	6	6	11	14	4	5	4	5	3	2	4	59	5	7	82	2	
26 TotalFootageDrilled	55	6	10	7	14	11	4	7	8	10	12	2	1	2	6	11	9	3	2	8	9	4	3	8		57	47	53	1	59
29 WaterDepth	53	7	17	9	19	22	8	12	11	6	26	4	16	16	2	25	17	5	6	14	17	8	5	1	53	13	13		3	82
4 DaysToTotalDepth	30	15	4	9	8	10	1		14	3	12	13	15	18	3	14	8	12	15	14	13	2	2	58	7	19	19	12	3	22
27 PreviousWellsInBlock	22	11	6	32		15	8	8	30	4	2	5	1	5	6	9	9	4	5	2	2	3	2	10	14	5	5	19	2	23
28 WellDaysActive	22	13	5	9	10	8	2	58	11	4	7	6	15	16	1	16	12	6	8	11	10	1	2		8	7	5	1	2	4
8 TotalMeasuredDepth	21	1	9	2	5	15	3	19	4	7	18	10	13	15	2	20	14	13	11	16	15	5	4	7	57		67	13	1	5
11 TotalVerticalDepth	19	1	9	2	5	11	3	19	4	9	12	7	11	10	2	17	13	11	8	13	12	5	4	5	47	67		13	1	7
1 PreviousWells	17		9	25	11	2	8	15	7	23	2	12	9	12	1	19	9	17	19	17	4	3	1	13	6	1	1	7	4	9
25 RigAgeAtTimeOfDrilling	16	23	26	6	4	8	8	3	14		7	12	3	1	9	6	8	9	12	3	16	8	3	4	10	7	9	6	1	11
3 PreviousWellsInRegion	14	25	3		32	5	9	9	14	6	7	9	5	6	3	8	1	11	12	13	1	3	1	9	7	2	2	9	2	11
5 DrillingDepthMax	10	12	9	9	5	14	7	13	2	12	30		15	32	13	35	6	51	61	58	30	1	4	6	2	10	7	4	2	5
18 SlotOrCant	10	12	9	6	5	33	11	18	6	1	28	32	82		12	49	23	36	43	44	22	5	7	16	2	15	10	16	1	6
2 DualActivity	8	9	4	1	9	33	6	8	13	8	20	6	21	23	1	42		17	6	27	33	8	8	12	9	14	13	17	2	14
13 DrawworksHP	8	19	6	12	5	20	12	15	5	12	22	61	34	43	2	42	6	65		67	24	4	6	8	2	11	8	6	2	5
15 PreviousWellsInField	7	2	10	5	15		3	10	16	8	29	14	31	33	3	39	33	20	20	27	26	12	8	8	11	15	11	22	1	10
17 PreviousWellsSinceUpgrade	7	9		3	6	10	12	4	1	26	19	9	6	9	8	15	4	5	6	13	7	3	2	5	10	9	9	17	2	10
22 MatOrInd	6	9	6	5	1	31	1	15	9	3	2	15		82	21	40	21	25	34	35	24	3	7	15	1	13	11	16	2	4
12 MudPumpHP	4	4	7	1	2	26	4	13	8	16	41	30	24	22	7	33	33	24	24	28		12	8	10	9	15	12	17	3	5
20 DerrickCapacity	4	17	5	11	4	20	15	12	4	9	21	51	25	36	2	43	17		65	69	24	3	7	6	3	13	11	5	1	4
6 AverageUtilisation	3	8	12	9	8	3		1	4	8	7	7	1	11	5	3	6	15	12	11	4	2	1	2	4	3	3	8	1	5
14 ZeroDischarge	3	1	8	3	6	3	5	3	4	9	17	13	21	12		7	1	2	2	2	7	4	1	1	6	2	2	2	2	6
16 TopDriveTorque	3	3	3	3	3	12	2	2	8	8	16	1	3	5	4	14	8	3	4	6	12		3	1	4	5	5	8	1	3
23 VariableDeckloadOperating	2	19	15	8	9	39	3	14	16	6	38	35	40	49	7		42	43	42	55	33	14	9	16	11	20	17	25	2	11
9 WellHPHT	2	1	2	1	2	8	1	2	2	3	9	4	7	7	1	9	8	7	6	7	8	3		2	3	4	4	5	3	2
21 WaterDepthMax	1	2	19	7	2	29	7	12	10	7		30	2	28	17	38	20	21	22	21	41	16	9	7	12	18	12	26	3	6
24 ContractLength	1	7	1	14	30	16	4	14		14	10	2	9	6	4	16	13	4	5	6	8	8	2	11	8	4	4	11	1	7
19 MudPumpNumber	1	17	13	13	2	27	11	14	6	3	21	58	35	44	2	55	27	69	67		28	6	7	11	8	16	13	14	1	4
30 HuricaneRisk	1	4	2	2	2	1	1	3	1	1	3	2	2	1	2	2	2	1	2	1	3	1	3	2	1	1	1	3		2

APF/d is AveragePerformanceFootagePerDay.

Figure 34: Pearson Correlations of all continuous variables for WRD2.0-preD

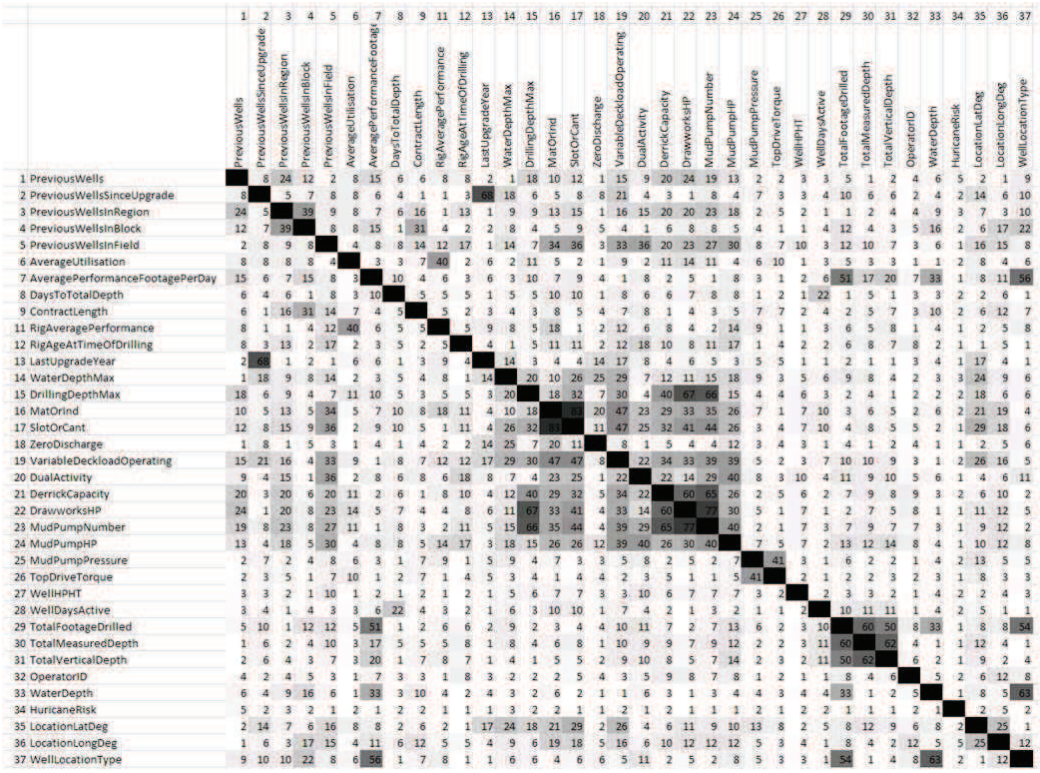


Figure 35: Pearson Correlations of all continuous variables for WRD2.0-preD, ordered by correlation with AveragePerformanceFootagePerDay

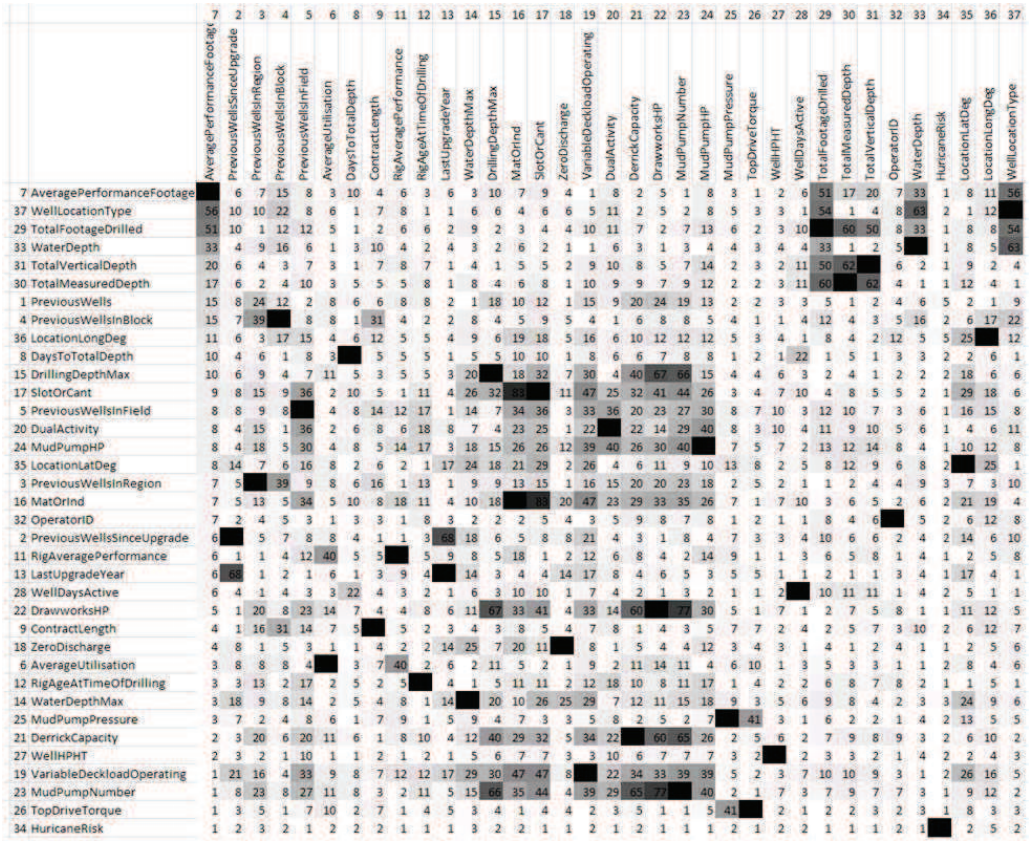


Table 36: Parameter search with WRD2.5

measure	init	markovB	parents	measureBayesScore	pctCorrect
BAYES	FALSE	FALSE	6	-192327.7494	49.64089565
BDeu	FALSE	FALSE	6	-194415.7853	49.64089565
BAYES	FALSE	FALSE	4	-196013.8979	49.64089565
BDeu	FALSE	FALSE	4	-197179.9876	49.64089565
BAYES	FALSE	FALSE	2	-203746.936	49.64089565
BDeu	FALSE	FALSE	2	-204099.4416	49.64089565
MDL	FALSE	FALSE	4	-204203.5	49.64089565
MDL	FALSE	FALSE	6	-204203.5	49.64089565
MDL	FALSE	FALSE	2	-205129.4829	49.64089565
AIC	FALSE	FALSE	6	-198210.3985	49.58808619
AIC	FALSE	FALSE	4	-198412.1354	49.58808619
AIC	FALSE	FALSE	2	-204151.5986	49.58808619
ENTROPY	FALSE	FALSE	2	-204596.153	48.97549641
ENTROPY	FALSE	TRUE	1	-225299.3704	47.69750739
AIC	FALSE	TRUE	1	-225236.0427	47.67638361
BAYES	TRUE	TRUE	2	-217287.1189	47.539079
BAYES	TRUE	FALSE	2	-217287.1189	47.539079
BDeu	TRUE	TRUE	2	-218079.2275	47.52851711
BDeu	TRUE	FALSE	2	-218079.2275	47.52851711
AIC	TRUE	TRUE	2	-217549.5577	47.47570765
AIC	TRUE	FALSE	2	-217549.5577	47.47570765
BAYES	FALSE	TRUE	1	-225201.3871	47.46514575
BDeu	FALSE	TRUE	1	-225031.6635	47.44402197
MDL	FALSE	TRUE	1	-225051.0142	47.42289818
MDL	TRUE	TRUE	2	-218689.7139	47.33840304
MDL	TRUE	FALSE	2	-218689.7139	47.33840304
ENTROPY	TRUE	TRUE	2	-217747.09	47.32784115
ENTROPY	TRUE	FALSE	2	-217747.09	47.32784115
BDeu	TRUE	TRUE	6	-212357.6852	47.29615547
BDeu	TRUE	FALSE	6	-212357.6852	47.29615547
BDeu	TRUE	TRUE	4	-212947.5254	47.29615547
BDeu	TRUE	FALSE	4	-212947.5254	47.29615547
MDL	FALSE	TRUE	4	-212654.9179	47.24334601
MDL	FALSE	TRUE	6	-212654.9179	47.24334601
MDL	FALSE	TRUE	2	-213333.9878	47.13772708
MDL	TRUE	TRUE	4	-217197.4225	47.08491762
MDL	TRUE	TRUE	6	-217197.4225	47.08491762
MDL	TRUE	FALSE	4	-217197.4225	47.08491762
MDL	TRUE	FALSE	6	-217197.4225	47.08491762
BDeu	FALSE	TRUE	2	-213835.4574	47.02154626
AIC	FALSE	TRUE	2	-214357.0081	47.02154626
BAYES	FALSE	TRUE	2	-213442.8301	46.9687368
AIC	TRUE	TRUE	6	-212485.5379	46.87367976
AIC	TRUE	FALSE	6	-212485.5379	46.87367976
AIC	TRUE	TRUE	4	-212763.8625	46.87367976
AIC	TRUE	FALSE	4	-212763.8625	46.87367976
BAYES	TRUE	TRUE	4	-210057.6912	46.85255598
BAYES	TRUE	FALSE	4	-210057.6912	46.85255598
ENTROPY	FALSE	TRUE	2	-214269.6016	46.73637516
ENTROPY	FALSE	FALSE	4	-217774.8697	46.63075623
BAYES	TRUE	TRUE	6	-208758.6153	46.47232784
BAYES	TRUE	FALSE	6	-208758.6153	46.47232784
BDeu	FALSE	TRUE	4	-214209.3131	46.16603295
AIC	FALSE	TRUE	6	-213162.5507	45.96535699
AIC	FALSE	TRUE	4	-213226.0655	45.96535699
ENTROPY	FALSE	TRUE	4	-223496.999	45.6590621
BAYES	FALSE	TRUE	4	-213371.1479	45.21546261
BDeu	FALSE	TRUE	6	-215360.0178	44.78242501
BAYES	FALSE	TRUE	6	-215022.3872	44.46556823
ENTROPY	TRUE	TRUE	4	-232577.8949	43.87410224
ENTROPY	TRUE	FALSE	4	-232577.8949	43.87410224
ENTROPY	FALSE	FALSE	1	-221176.4407	39.63878327
AIC	FALSE	FALSE	1	-221165.1337	39.61765948
BAYES	FALSE	FALSE	1	-221080.2708	39.56485002
MDL	FALSE	FALSE	1	-221105.8043	39.56485002
BDeu	FALSE	FALSE	1	-221126.3488	39.56485002
ENTROPY	FALSE	FALSE	6	-278192.6579	39.46979299
ENTROPY	FALSE	TRUE	6	-282679.4357	39.06844106
ENTROPY	TRUE	TRUE	6	-301915.3489	38.27629911
ENTROPY	TRUE	FALSE	6	-301915.3489	38.27629911
BAYES	TRUE	TRUE	1	-260225.713	33.65019011
BAYES	TRUE	FALSE	1	-260225.713	33.65019011
BDeu	TRUE	TRUE	1	-260225.713	33.65019011
BDeu	TRUE	FALSE	1	-260225.713	33.65019011
MDL	TRUE	TRUE	1	-260225.713	33.65019011
MDL	TRUE	FALSE	1	-260225.713	33.65019011
ENTROPY	TRUE	TRUE	1	-260225.713	33.65019011
ENTROPY	TRUE	FALSE	1	-260225.713	33.65019011
AIC	TRUE	TRUE	1	-260225.713	33.65019011

Review of Extracts from Kordon's Work in View of This Research

One of the main issues encountered while working on this research was the barriers raised in the course of interaction between different disciplines: abstract computing research and the real world of engineering and stakeholders. In his book "Applying computational intelligence" [27], Kordon explores the problems and issues associated with technology transfer between the world of advanced theory of *computational intelligence* and the world of practical applications of engineering methodologies to the industry. He explores the specificities of various *computational intelligence* techniques, their typical applications and the creation of value and reviews application strategies. Throughout the book, Kordon's focus on 'competitive advantage' helps expanding the industrial side of this research.

The adoption of the technology developed in this research relies on the credibility associated with it at the end of the project. Kordon highlights that "one of the differences between *computational intelligence* and the other high-tech alternatives is that it has already demonstrated its potential for value creation in many application areas" [27] (page 221). This confirms observations in the energy sector regarding the use of *computational intelligence* for wind power systems [167], electric power systems [168], thermal plants [169], and the oil industry (including reservoir characterization, gas storage, seismic inversion, engine oil development, oil field development, production scheduling) [170] as well as for biology [171]. Kordon explains that a variety of technology such as "search engines, word-processor, spell checkers and [...] rice cooker" are also everyday examples of the application of the technology [27]. One highly publicised event, demonstrating further avenue of value creation of *computational intelligence*, was the "chess battle between Kasparov and Big Blue" [27].

The key elements to demonstrate a competitive advantage from a research approach are expressed in [27] (page 233). The competitive advantage is clearly demonstrated by the clarification of its technical superiority, the indication of a low cost of ownership and the evidence of its ability to be applied in "areas of high impact". This project is able to deliver the competitive advantage sought by Kordon because, first, it is inherent to the scientific and methodical approach; secondly, the cost of computation is constantly lowering while the cost of field experts is raising or at least remains stable; lastly, the potential impact of an objective decision support system to rig performance can easily be considered a "high impact" in view of the oil drilling rig operating costs only.

In the course of his discussion, Kordon [27] (page 248) analyses the main competitive advantage of *computational intelligence*. The first advantage is the objectivity of the intelligence provided by those algorithms. They have indeed less possibility to be contaminated by human biases an expert could have. The second advantage is the ability of the algorithms to deal with uncertainty. The *computational intelligence* approach to data modelling inherently includes the real world uncertainties within its models. Evolutionary Computation adopts the strategy of "reducing uncertainty through simulated evolution". "This technology is one of the rare cases when modelling can begin with no a priori assumptions at all" [27]. Kordon also states that "uncertainty is gradually reduced by the evolving population and the fittest winners in this process are the final results of this fight with the unknown." The third advantage is the ability of *computational intelligence* to deal with complexity. Evolutionary Computation amounts to "reducing complexity through simulated evolution" [27]. One side-effect during simulated evolution is that "the unimportant variables are gradually removed from the final solutions, which leads to automatic variable selection and dimensionality reduction" [27]. The fourth advantage is the "unique capability to automatically create innovative solutions" [27]. Using small building blocks, an evolutionary algorithm can generate almost any type of new structure and find new relationships between the different variables. Finally, the relative low-cost of modelling is a major advantage. *Computational*

variables. Finally, the relative low-cost of modelling is a major advantage. *Computational intelligence* can create high quality empirical models that would have taken decades, if at all possible, for human experts to imagine.

Kordon also explores common issues encountered when applying *computational intelligence* and proposes ways to mitigate the risks [27] (page 257). The first issue to explore here is the *change function*. According to this idea, proposed by Coburn [210], "people are only willing to change and accept new technologies when the pain of their current situation outweighs the perceived pain of trying something new". The reaction level from a user to a new technology product can vary from indifference to crisis. "People are more willing to change the higher the level of crisis they have in their current situation." The suggested response to that issue [27] is that the application of *computational intelligence* should be made in such a way as to minimise the perceived pain of adoption, while demonstrating a clear competitive advantage such as when "a novel solution is needed in a dynamic environment of high complexity or uncertainty" [27]. The second issue to mention here is an issue created by the technologists themselves. Christensen mentions that three quarters of the money spent on product development investments result in products that do not succeed commercially [211]. This issue is produced by the technologists trying to create a need where nothing has ever been needed. "A typical result of the technocentric culture is pushing technology improvement at any cost by management. Often introducing technology is part of the process [...]" [27]. The "real customer needs" are neglected. Relating to that issue, Levitt notes: "When people buy quarter-inch drill bits, it's not because they want the bits themselves. People don't want quarter-inch drill bits – they want quarter-inch holes" [212]. Kordon goes further and says that: "Imposing the technology for purely technology's sake may lead to lost credibility, as we know from applied AI" [27]. The third important issue is the increased complexity of applied new solutions. This complexity is the "root cause for the high total perceived pain of adoption." An example given by Kordon [27] is the use of neural networks to control industrial nonlinear systems. The neural network will allow more flexibility and control over the dynamic environment but the user will have to deal with 10 to 12 parameters necessary to tune the neural network when a conventional PID controller (proportional–integral–derivative, generic control loop feedback mechanism widely used in industrial control systems [213]) commonly requires 3 parameters. Other issues such as "technology hype", "modelling fatigue" and the over-academic image of *computational intelligence* are examples of additional issues addressed in [27] as well.

Kordon describes his approach to integrating *computational intelligence* technology in [27] (page 309). One of his key messages on integration is that the "reality of industrial applications requires the joint solution of the technical, infrastructural, and people-related components of a given problem". Kordon [27] lists obstacles to the integration efforts. Some of them are usually related to data quality (availability, range and occurrence coverage, frequency of collection, noise level[30]), availability of expertise (to understand the domain knowledge), and limited infrastructure (integration capabilities with current systems).

[30] Kordon uses the acronym GIGO: Garbage-in-garbage-out to talk about issues with data quality as an input to data-trained models.

Glossary

- **Average Day Rate:** Average price of the rig on the market.
- **Average Feet drilled Per Non-(idle/WOW) Day:** Distance drilled adjusted for the days when the rig was not drilling.
- **Average Price Per Foot Drilled:** Relative price of drilling one foot with this rig. This is an alternative measure of performance.
- **Bulk Cement:** Bulk cement is a powder form of cement that is used for pumping down the well to set casing pipe, and to block the well if required. Only the amount needed to be used immediately is made up, as it sets very quickly.
- **Bulk Mud:** Bulk mud is powder storage of ingredients used in making up drilling fluid that is pumped through the drill pipe. Drilling mud has a very high density to hold gas and oil that is under pressure in the well. Bulk mud is stored in dry form so that different densities of liquid mud can be made up as and when required for the current drilling conditions.
- **Cantilever Capacity:** Cantilever capacity is the amount of weight that can be carried at the end of the cantilever on a deployed jack-up rig when on location. The cantilever on a jack-up rig is a mechanical arm able to extend the drilling package over the side of the rig hull. The drilling package includes the derrick and most of the machinery necessary for drilling an oil well.
- **Design Company:** The name of the company which designed the rig.
- **Drawworks:** This is a winch that is used to raise and lower the top drive which holds the drill pipe. The stronger the draw works, the longer the pipe that can be held and the faster the pipe can be raised and lowered.
- **Footage Drilled:** Linear length of the hole drilled under the ocean floor.
- **Last Upgrade Year:** The year the rig has been upgraded the last time.
- **Market Category:** The category in which the rig is marketed. Most often it is based on the rig type and the water depth rating.
- **Operator:** The name of the company operating the rig on behalf of the rig owner.
- **Overrun Rates:** Rates which have to be paid if the rig goes over contract when drilling a well.
- **Recent Management Change:** An indicator of recent management change. This can impact performance as the crew might have to adapt to new operating practices.
- **Rig Design:** The specific model of rig.
- **Spud Date:** Spud date is the date at which the rig starts drilling the well. Start date and Spud date are often different due to the preparation and deployment necessary before drilling.
- **Start Date:** Date the oil drilling rig arrives on site and starts preparing to drill.
- **Storage Mud Liquid:** Storage Mud Liquid is a liquid store of drilling fluid that is ready to be pumped down the drill pipe. The fluid in these containers is continuously circulated through the drill pipe and back up the well, cleaned and then pumped down again.
- **Termination date:** Date the oil drilling rig completes the drilling and testing operations following the drilling.
- **Time Since Last Well:** The number of days a rig has spent inactive since it last drilled a well.
- **Top Drive:** Tool used to drive the drilling process. It turns the drill pipe to perform drilling.
- **Total Depth Date:** Date the oil drilling rig has reached the target depth of drilling.
- **Utilisation of Fleet for Rig Type:** Ratio of the fleet in use for the specific rig type.
- **Utilisation of Fleet:** Ratio of the fleet in use or under contract.
- **Water Depth Rating:** Depth the oil drilling rig is certified to operate at.
- **Water Depth:** Depth from the ocean floor to the surface.
- **WOW:** Waiting on weather, the rig is idle and waiting for clearer conditions.