DEVELOPMENT OF A RECURSIVE DEEP LEARNING USING NATURAL LANGUAGE PROCESSING AND COMPUTER VISION FOR OIL DRILLING OPERATION

***D. U. Ashishie, **Dr. Chinogolum Ituma, ***Dr. Igwe Joseph.**
Computer Science Department, Ebonyi State University Abakalika, Nigeria

ABSTRACT

This paper presents the development of a recursive deep learning framework utilizing natural language processing and computer vision techniques for optimizing oil drilling operations. Following the CRISP-DM methodology, the objectives include extracting both structured and unstructured data from various oilfield drilling report documents for sentiment analysis, training models capable of autonomously understanding hidden information in text/images to determine performance accuracy, leveraging daily drilling report analysis to enhance model accuracy and mitigate non-productive time risks, thus improving drilling operation efficiency, and creating a user-friendly interface for seamless interaction with the model. The results and discussion encompass predicted values, comparison between actual and predicted values, as well as scattered and line plots visualizing the relationship between actual and predicted values.

Keywords: Recursive Deep Learning, Natural Language Processing, Computer Vision, Oil Drilling, Sentiment Analysis, CRISP-DM, Model Training, Efficiency Optimization.

INTRODUCTION

Human life basically consists of emotions and opinions. Emotions and opinions manage how humans communicate with each other and how they motivate their actions. Emotions and opinions play a role in nearly all human actions and can influence the way humans think, what they do, and how they act. In the past few years, a great attention has been received by web documents as a new source of individual opinions and experience. This situation is producing increasing interest in methods for automatically extracting and analyzing individual opinion from web documents such as customer reviews, weblogs and comments on news. This increase was due to the easy accessibility of documents on the web, as well as the fact that all these were already machine-readable on gaining (Cambria, 2017).

At the same time, Machine Learning methods in Natural Language Processing (NLP) and Information Retrieval have considerably increased development of practical methods, making it an interesting area of research. Recently, many researchers have focused on this area by trying to fetch opinion information and analyze it automatically with computers (Sohangir, 2018). This is because there are large amounts of information created by users on the Internet, including product reviews, movie reviews, forum entries, blog and so on. How to analyze and summarize the opinions expressed in these documents is a very interesting domain for researchers. Natural language processing (NLP) is a theory motivated range of computational techniques for the automatic analysis and representation of human language. NLP research has evolved from the era of punch cards and batch processing, in which the analysis of a sentence could take up to 7 minutes, to the era of Google, in which millions of WebPages can be processed in less than a second (Cambria, 2014). NLP enables computers to perform a wide range of natural language related tasks at all levels, ranging from parsing and part-of-speech (POS) tagging, to machine translation and dialogue systems. Deep learning architectures and algorithms have already made impressive advances in fields such as computer vision and pattern recognition. Going by this trend, NLP research in recent time is now increasingly focusing on the use of new deep learning methods. For decades, machine learning approaches targeting NLP problems have been based on shallow models like support vector machines (SVM) and logistic regression (LR) trained on very high dimensional and sparse features. In the last few years, neural networks based on dense vector representations have been producing superior results on various NLP tasks.

This trend is sparked by the success of word embedding (Mikolov, Karafi, Burget, Cernock, and Khudanpur, 2010) and deep learning methods (Socher, Perelygin, Wu, and Chuang, 2013). Deep learning enables multi-level automatic feature representation learning. In contrast, traditional machine learning based NLP systems liaise heavily on hand-crafted features which is time consuming and often incomplete.

Integrating computer vision and natural language processing is a novel interdisciplinary field that has received a lot of attention recently. In human perception, visual information is the dominant modality for acquiring knowledge of the world; about 30% of the human brain is dedicated to visual processing. The extent to which language is directly involved in the visual process is still a matter of debate. However, in the attempt to achieve artificial intelligence, making use of certain aspects of language provides interpretability and enables productive human-machine interaction.

Computer vision (CV) is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on derived information. Computer vision leverages on Artificial Intelligence (AI) to enables computers to think, to see, observe and understand. Computer vision works much the same as human vision, human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision on the other hand trains machines to perform these functions, but in much less time with cameras, data and algorithms rather than retinas, optic nerves and a visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

Computer Vision (CV) tasks can be summarized by the concept of 3Rs (Jonathan and Jitendra, 2015), which are *reconstruction*, *recognition*, and *reorganization* (3R). Reconstruction involves estimating the three-dimensional (3D) scene that gave rise to a particular visual image. It can be accomplished using a variety of processes incorporating information from multiple views, shading, texture, or direct depth sensors. Reconstruction process results in a 3D model, such as point clouds or depth images. Some examples for reconstruction tasks are Structure from Motion, scene reconstruction, and shape from shading. Recognition involves both 2D problems (like handwritten recognition, face recognition, scene recognition, or object recognition), and 3D problems (like 3D object recognition from point clouds which assists in robotics manipulation). Recognition results in assigning labels to objects in the image.

Reorganization involves bottom-up vision segmentation of the raw pixels into groups that represent the structure of the image. Reorganization tasks range from lowlevel vision like edge, contour, and corner detection, intrinsic images, and texture segmentation to high-level tasks like semantic segmentation (Joao and Cristian, 2010), which has an overlapping contribution to recognition tasks. A scene can be segmented based on low-level vision (David, Charless, and Jitendra, 2004) or high-level information like shadow segmentation (Aleksandrs, Cornelia, and Yiannis, 2014) that utilizes class information.

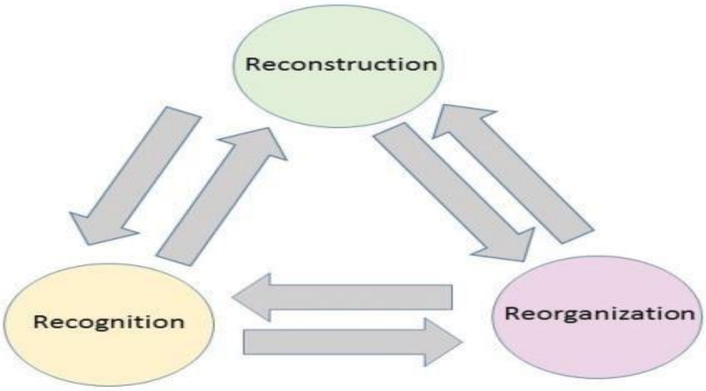

Figure 1: The 3R in computer vision (Jonathan and Jitendra, 2015)

There is always meaning lost when translating between one language and another, hence there is need to bridge the gap. When "translating" between the low level pixels or contours of an image and a high level description in word labels or sentences, there is a wide chasm to be crossed. Bridging the Semantic Gap (Zhao and William, 2002) means building a bridge from visual data to language data like words or phrases. An example could include labeling an image patch that contains an object with a word is object recognition. Labeling a background in an image is scene recognition. Assigning words for pixel grouping is semantic segmentation (Jeffrey, Socher, and Christopher, 2014)(Carreira and Sminchisescu, 2010). If we know how the words are related to each other, then it can give a clue for visual processing to better disambiguate different visual constructs.

Bernard Vauquois examined the machine translation approaches to represent it in the form of a triangle. The Vauquois triangle visualizes and describes the classical approaches to machine translation, showing the evolution of those approaches.

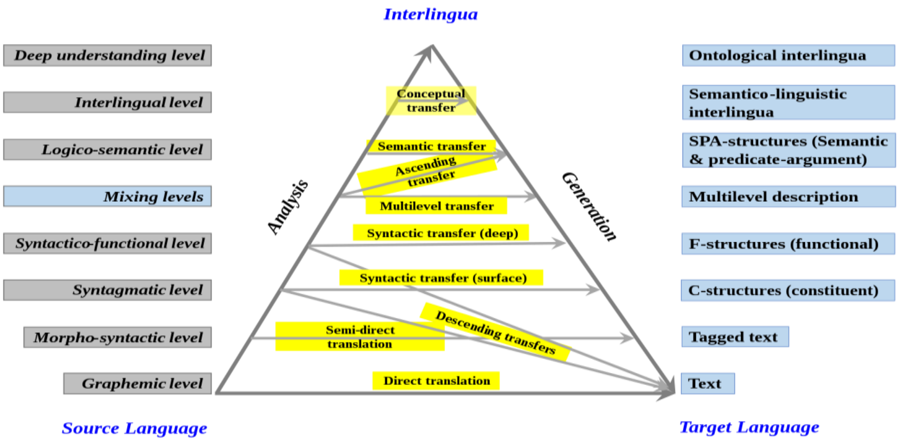

Figure 2: A rendition of the Vauquois triangle, illustrating the various approaches to the design of machine translation systems (Boitet, 2022)

OBJECTIVES

1. To extract both structured and unstructured data from free text in different oilfield drilling report documents suitable for sentiment analysis.
2. To train models that is capable of independently understanding hidden information in text/images and systematically determines the performance accuracy of the trained models.
3. To use analysis from oil daily drilling report to increase the accuracy of the model and mitigate the risks of non-productive time in drilling events thereby increasing the efficiency of the drilling operation.
4. To develop an interface that will enable end users interact easily with the model.

LITERATUIRE REVIEW

There has been great progress in delivering technologies in natural language processing (NLP) such as extracting information from big unstructured data on the web, sentiment analysis in social networks or grammatical analysis for essay grading. One of the goals of NLP is the development of general and scalable algorithms that can jointly solve these tasks and learn the necessary intermediate representations of the linguistic units involved. However, standard approaches towards this goal have two main limitations which includes.

1. **Simplifying Language Assumptions:** In NLP and machine learning, we often develop an algorithm and then force the data into a format that is compatible with this algorithm. For instance, a common first step in text classification or clustering is to ignore word order and grammatical structure and represent texts in terms of unordered lists of words that is called bag of words; which leads to obvious problems when trying to understand a sentence.
2. **Feature Representations:** While a lot of time is spent on models and inference, a well-known secret is that the performance of most learning systems depends crucially on the feature representations of the input. For instance, instead of relying only on word counts to classify a text, state of the art systems uses part-of speech tags, special labels for each location, person or organization (so called named entities); parse tree features or the relationship of words in a large taxonomy such as WordNet. Each of these features has taken a long time to develop and integrating them for each new task slows down both the development and runtime of the final algorithm (Bowman, Potts and manning, 2014).

These two main issues can be overcome by providing effective and general representations for sentences without assuming word order independence as well as providing a most unusual performance with no or little manually designed features. These were achieved by combining ideas from the fields of natural language processing and deep learning. Deep learning however, is a sub field of machine learning which comfortably handles the problem of feature representation by automatically learning feature representations from raw input which can then be readily used for prediction tasks.

There has been great success using deep learning techniques in image classification (Krizhevsky, Sutskever, and Hinton, 2012) (Krizhevsky *et al.*, 2012) and speech recognition (Hinton *et al.*, 2012). However, an important aspect of natural language processing and computer vision that has not been accounted for in deep learning is the pervasiveness of recursive or hierarchical structure. Therefore, this thesis describes new deep models that extend the ideas of deep learning to structured inputs and outputs, thereby providing a solution to the problem of simplifying language assumptions. In other words, while the methods implemented here are based on deep learning they extend general deep learning ideas beyond classifying fixed sized inputs and introduce recursion and computing representations for grammatical language structures.

The model used in this thesis is Recursive Deep Learning which is a variation and extension of unsupervised and supervised recursive neural networks. Recursive neural networks parse natural language which enables them to find the grammatical structure of a sentence and align the neural network architecture accordingly. Also, the recursion comes applying the same neural network at every node of the grammatical structure. Recursive deep models also address the fundamental issue of learning feature vector representations for variable sized inputs without ignoring structure or word order. These structures when discovered, helps to characterize the units of a sentence or image and how they compose to form a meaningful whole. The models can also learn compositional semantics often purely from training data without a manual description of features that are important for a prediction task (Frome, Corrado, and Shlens, 2013).

In recent time, most natural language processing activities were carried out using shallow machine learning approach; which is slow and most often yield very low performance.

Hence, the needs to introduce a deep learning approach. Majority of machine learning methods work well because of human-designed representations and inputs features. When machine learning is applied only to the input features, it only becomes merely about optimizing weights to make the best final prediction. Deep learning is about putting back together representation learning with machine learning. It attempts to jointly learn good features, across multiple levels of increasing complexity and abstraction, and the final prediction (Goldberg, 2016).

Natural language processing (NLP) has long been viewed as one aspect of artificial intelligence (AI), since understanding and generating natural language are high-level indications of intelligence; deep learning is an effective AI tool as well as a bridge between the massive amounts of data and AI (Goldberg, 2016). Deep learning refers to applying deep neural networks to massive amounts of data to learn a procedure aimed at handling a task. The task can range from simple classification to complex reasoning. In other words, deep learning is a set of mechanisms ideally capable of deriving an optimum solution to any problem given a sufficiently extensive and relevant input dataset. Summarily, deep learning is detecting and analyzing important structures/features in the data aimed at formulating a solution to a given problem.

Deep Learning Architectures

Numerous deep learning architectures have been developed in different research areas, for example, in NLP applications, employing recurrent neural networks (RNNs) (Lipton, Berkowitz, and Elkan, 2015), convolutional neural networks (CNNs) (Kim, 2014), and recursive neural networks (Socher, Lin, and Manning, 2011.)

Multi-Layer Perceptron: A multilayer perceptron (MLP) has at least three layers (input, hidden, and output layers). A layer is simply a collection of neurons operating to transform information from the previous layer to the next layer. In the MLP architecture, the neurons in a layer do not communicate with each other. MLP employs nonlinear activation functions. Every node in a layer connects to all nodes in the next layer, creating a fully connected network. Figure 2. below shows the MLPs, which is the simplest type of Feed-Forward Neural Networks (FNNs). FNNs represent a general category of neural networks in which the connections between the nodes do not create any cycle, i.e. in a FNN there is no cycle of information flow (Li, Sun, and Han, 2020).

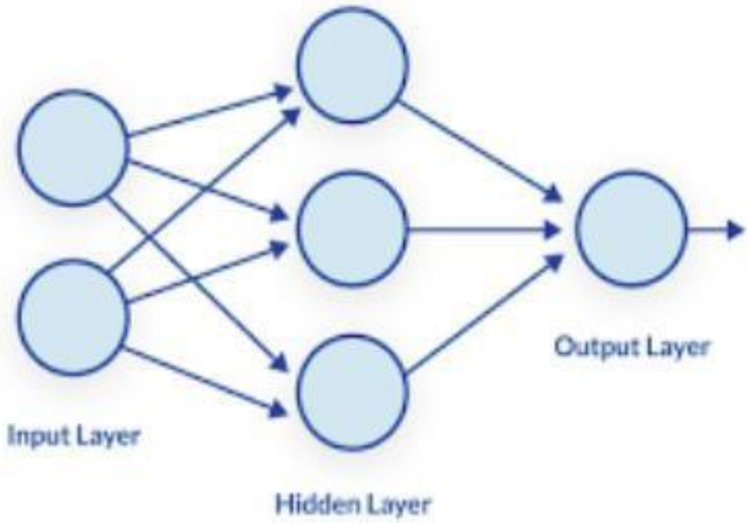

Figure 3: Multi-layer perceptron architecture (Abdi, Hasan, Shamsuddin, Idris, and Piran, 2021)

Convolutional Neural Networks

Convolutional neural networks (CNNs), whose architecture is inspired by the human visual cortex, are a subclass of feed-forward neural networks. CNNs are named after the underlying mathematical operation, convolution, which yields a measure of the interoperability of its input functions. Convolutional neural networks are usually employed in situations where data is or needs to be represented with a 2D or 3D data map. In the data map representation, the proximity of data points usually corresponds to their information correlation.

In convolutional neural networks where the input is an image, the data map indicates that image pixels are highly correlated to their neighboring pixels. Consequently, the convolutional layers have 3 dimensions which are width, height, and depth. Hence, the reason why majority of researches dedicated to CNN are conducted in the Computer Vision field (Krizhevsky, Sutskever, and Hinton, 2012.)

A CNN takes an image represented as an array of numeric values. After performing specific mathematical operations, it represents the image in a new output space. This operation is also called feature extraction and helps to capture and represent key image content. The extracted features can be used for further analysis, for different tasks. One example is image classification, which aims to categorize images according to some predefined classes. Other examples include determining which objects are present in an image and where they are located.

In the case of utilizing CNNs for NLP, the inputs are sentences or documents represented as matrices. Each row of the matrix is associated with a language element such as a word or a character. The majorities of CNN architectures learn word or sentence representations in their training phase. A variety of CNN architectures were used in various classification tasks such as Sentiment Analysis and Topic Categorization (Kim, 2014). CNNs were employed for Relation Extraction and Relation Classification as well (Zeng, Liu, Lai, Zhou, and Zhao, 2014.).

Recurrent Neural Network

A recurrent neural network (RNN) is constructed by lining up a sequence of FNNs and feeding the output of each FNN as an input to the next one. Just as in FNNs, layers in an RNN can be categorized into input, hidden, and output layers. In discrete time frames, sequences of input vectors are fed as the input, one vector at a time. For example, after inputting each batch of vectors, conducting some operations and updating the network weights, the next input batch will be fed to the network. Figure 2. below shows a recurrent neural network where at each time step predictions are made and parameters of the current hidden layer are used as input to the next time step (Tang, Qin, and Liu, 2015).

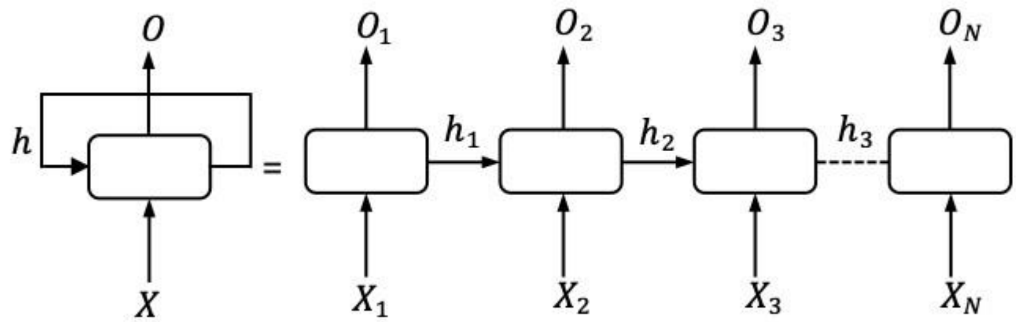

Figure 4: Recurrent Neural Networks (RNN) (Tang, Qin, & Liu, 2015)

Hidden layers in recurrent neural networks can carry information from the past; this feature makes it possible for hidden layers in recurrent neural networks to be classified as a memory. This characteristic makes them specifically useful for applications that deal with a sequence of inputs such as language modeling (Mikolov, Karafia, Burget, ernocky, and Khudanpur, 2010), i.e. representing language in a way that the machine understands. Long Short-Term Memory Network (LSTM (Rao, Huang, Feng, and Cong, 2018) is one of the most widely used classes of RNNs. LSTMs try to capture even long-time dependencies between inputs from different time steps, modern Machine Translation and Speech Recognition often rely on long short-term memory (LSTMs).

Auto-encoders

Auto-encoders implement unsupervised methods in deep learning and they are widely used in dimensionality reduction (Bengio, Ducharme, Vincent and Jauvin, 2003) or NLP applications, which consist of sequence to sequence modeling (Mikolov, Karafi, Burget, Cernock, and Khudanpur, 2010). Since auto-encoders are unsupervised, there is no label corresponding to each input. Rather, they aim to learn a code representation for each input. The encoder is like a feed-forward neural network in which the input gets encoded into a vector (code). The decoder operates similarly to the encoder, but in reverse, i.e. constructing an output based on the encoded input. In data compression applications, there is need for the created output to be as close as possible to the original input. Auto-encoders are loss, this is to say, the output is an approximate reconstruction of the input. Fig. 2.3 illustrates the schematic of an Auto-encoder.

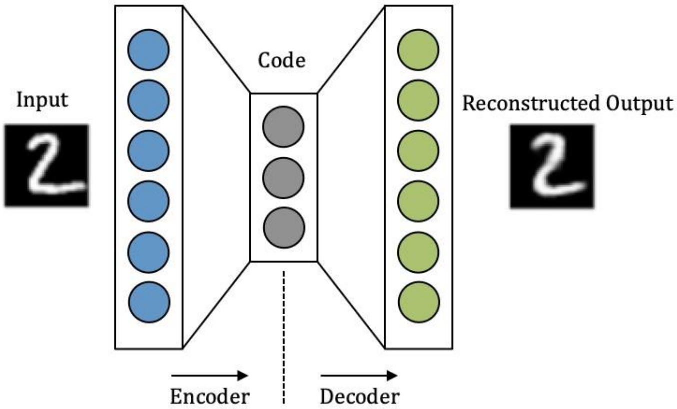

Figure 5: Schematic of an Auto-encoder (Mikolov, Karafia, Burget, and Ernocky, 2010.)

Generative Adversarial Networks

Goodfellow (Goodfellow, Pouget-Abadie, Mirza, Xu, and Warde-Farley, 2014) introduced Generative Adversarial Networks (GANs). Fig. 6 below shows a GAN as a combination of two neural networks, a discriminator and a generator. The whole network is trained in an iterative process. The generator network generates a fake sample, then the discriminator network tries to determine whether this sample (ex. an input image) is real or fake, i.e. whether it came from the real training data (data used for building the model) or not. The goal of the generator is to fool the discriminator in a way that the discriminator believes the artificial (i.e. generated) samples synthesized by the generator are real. This iterative process continues until the generator produces samples that are indistinguishable by the discriminator. In other words, the probability of classifying a sample as fake or real becomes like flipping a fair coin for the discriminator.

The goal of the generative model is to capture the distribution of real data while the discriminator tries to identify the fake data. One of the interesting features of GANs (regarding being generative) is that once the training phase is finished, there is no need for the discrimination network, hence, the generation network can be solely worked with; this is to say, having access to the trained generative model is sufficient.

Different forms of GANs has been introduced, e.g. Sim GAN (Shrivastava, Pfister, Tuzel, Susskind, and Wang, 2017). In one of the most elegant GAN implementations (Karras, Aila, Laine, and Lehtinen, 2017), entirely artificial, yet almost perfect, celebrity faces are generated; the pictures are not real, but fake photos produced by the network. GAN‟s has since received significant attention in various applications and have generated astonishing result

(Tavaf, Torfi, Ugurbil, and Van de Moortele, 2021). In the NLP domain, GANs often are used for text generation (Yu, Zhang, Wang, and Yu, 2017).

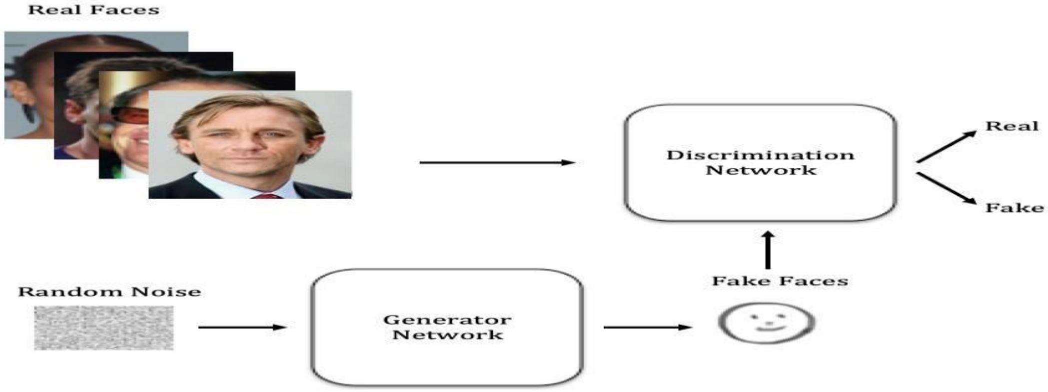

Figure 6: A Generative Adversarial Network (Radford, Metz, and Chintala, 2015)

Recursive Neural Networks

While recurrent neural networks represent a natural way to model sequences, languages however, exhibits a natural recursive structure, where words and subphrases combine into phrases in a hierarchical manner. This structure can be represented by a constituency parsing tree. Hence, tree-structured models have been used to better make use of such syntactic interpretations of sentence structure (Cheng, Yu, Feris, Kumar, and Choudhary, 2015.). Specifically, in a recursive neural network, the representation of each non-terminal node in a parsing tree is determined by the representations of all its children.

Oil drilling operation domain knowledge

Drilling process

Basically, the drilling process can be described as follows (Nybo, Ph.D. thesis, NTNU, 2009). A rotating pipe (or string), extends from the rig to the bottom of the well, ending in a bottom hole assembly (BHA) that includes the drill bit. The drill bit is the head of the whole assembly, where resides the actual drilling mechanism. The pipe is composed by several elements assembled one by one from the top, as the hole gets deeper. A stand, is composed by two or three segments of drill pipe joint together and constitutes a typical reference unit for the drilling process (three-joint stands are also called trebles or triples). As the drilling process proceeds, new stands are mounted one on top of the other to augment the drill pipe length. Conversely, in tripping the stands are disconnected and stocked as they are extracted from the well. A stand can be made of drill collars as well (drill collars are heavier pipes used to provide weight on bit for drilling).

The gravity acting on the drill pipe and in particular on the collars provides the downward force necessary to the bit to break the rock. The driller lowers the drill string on the bottom of the wellbore, and then controls the weight applied to the bit. The weight on bit, technical name of the weight that the operator let to be applied to the bit, is always smaller than the actual weight of the string. We could say that an operator never lays down the string freely. The drill bit crushes the rock into cuttings, which are removed from the bottom and mechanically transported up towards the rig by way of a special drilling mud. The latter is pumped down the pipe and returns to the rig passing through the annulus, i.e. the space between the pipe and the wall of the borehole. The wall of the well is periodically coated with a protective casing and cemented. For this and other tasks (e.g., substitution of the drill bit), the whole pipe must be extracted from the hole and reinserted afterwards, an operation called tripping.

Oil rig description and nomenclature

The rig includes various subsystems: the hoist and rotation system, the power generation system, the mud circulation system, and the well control system. The hoist and rotation system have two main functions. First of all, it is used to hoist or lower the drilling equipment from the well. In addition, it applies rotation to the drill string and the bit, as required in the drilling process. Its main elements, are the derrick (or mast) with its substructure, the crown block at the top of the rig, the traveling block, the top drive, the rotary table, the draw works, the drill line, and the dead line anchor.

The traveling block is the set of sheaves that moves up and down in the derrick. The wire rope (drill line) threaded through them is \reeved" back to the stationary crown blocks located on the top of the derrick. This pulley system enables heavy loads (drill string, casing and liners) to be lifted out of or lowered into the wellbore. The top drive is a motor suspended from the derrick that is used to rotate the drill string during the drilling process.

It replaces the traditional Kelly or rotary table, which is also used to apply rotation and to support the drilling assembly. The mud circulation system is required to prepare, store, and pump the drilling mud. The mud has various functions: carry the drilled cuttings from the hole bottom up to the rig level, cool down and lubricate the bit, hold the bore hole from falling down. The hole is stable when the hydrostatic pressure of the mud is greater than the pore pressure. Otherwise gas blows upwards (well owing/kick), and a blowout can occur. If the mud hydro- static pressure is too large it may cause fractures in the formation, with consequent mud losses. The mud can be water or oil base.

The main function of the well control system is to prevent a blow-out of the well. The Blow Out Preventer (BOP) is a large valve that may provide a seal for the well bore. The BOP stack is an assembly of blow out preventers. The choke & kill manifold and lines are arrangements of pipes and special valves used to circulate the mud when the BOPs are closed to control the pressures encountered during a kick.

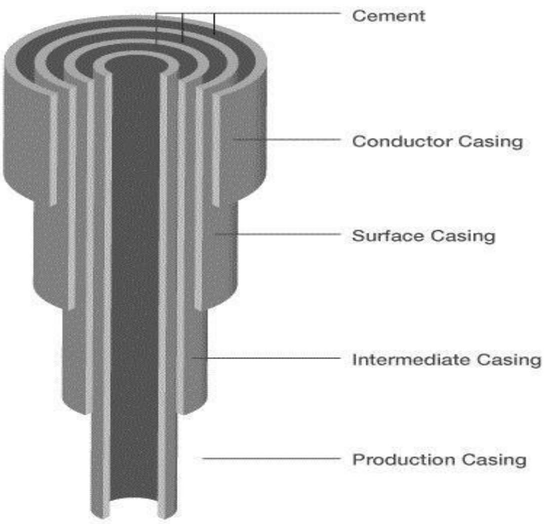

Figure 7. Casing.

The casing provides mechanical support and isolation to the well. Various casings with progressively smaller diameters may be used, starting from an initial hole with a diameter of 70 cm down to 10 cm. The surface casing is run to install the well head and BOP system. It also has the function of isolating the shallow water. The intermediate casing is run to stabilize the bore hole and to isolate the levels with different gradient. The production casing is run to isolate the productive levels and to protect the completion devices. When the mud weight value becomes closer to the fracture gradient it is necessary to set a new casing. Each time a new casing is set up it is necessary to change the BHA. This is due to the smaller diameter of the hole that clearly needs a smaller bit.

The casing can break from the inside (burst) or collapse for large pressure differences between the inside and the outside. A burst can happen e.g. during kicks or blow outs, cementing jobs, a LOT (Leak out test) or a FIT (Formation Integrity test). It can collapse because of mud losses or during the cementing job (because of the cement slurry density). During the casing running it may also be subject to axial loads (tensile or compression stress). Tensile stresses may also occur during cementing (at bump plug) or because of thermal expansion.

The casing operations may last from one day to several days, depending on the length of the casing. Cementing is mainly used to seal casings, but it may also be used to isolate the well or to perform a squeeze (cement is pumped in a fractured formation to prevent mud losses). The cementing job consists in pumping the cement slurry in the annulus between open hole and casing. The slurry is separated from the mud by two plugs. The mud pushes the cement downwards. At some point the lower plug exits the casing and the slur outside the casing recirculating upwards. The mud is stopped when the upper plug reaches the bottom. Then, the cement is left to consolidate (Wait on Cement, WOC). When the drilling resumes, the cement at the bottom is drilled and removed.

Well completion consists in running down hole the devices required to produce the formation fluid (hydrocarbon) to the surface in a controlled way. The completion design includes also safety devices to shut the well in case the well head is damaged and the well integrity is compromised. In our study we consider just data coming from the drilling part of the Well creation, from the initial phase to the set-up of the last casing or in some cases until the total break, i.e. the moment when the drilling assembly is left in the bottom of the hole due to the impossibility to unlock it. In this situation the loss is dramatic and it could be necessary to drill a completely new hole.

Characterization of typical anomalies

Types of anomalies

1. Well problems: all borehole and cased hole problems generated by the well with exclusion of the equipment failures or malfunctions. These include: circulation losses, sticking, fluid (kick, blowout), other geological problems (caving, gumbo, tight hole, hole geometry, salt formations, hole), cased hole problems, etc.
2. Rig failures: all drilling rig contractor equipment failures or malfunctions with exclusion of the down hole equipment. The following components are involved: draw-works1, mud pumps, power generation system, top drive system, surface b.o.p., subsea equipment, mooring etc.
3. Down hole equipment failures: failures or malfunctions of every equipment run in hole. The following components are involved: BHA, drill string, casing/liner, completion/test string, wire line equipment, coiled tubing, etc.
4. Surface equipment failures: failures or malfunctions of every equipment used in surface or at sea bed with exclusion of the drilling contractor equipment. The following components are involved: wellhead, company surface equipment, contractor surface equipment, etc.
5. Waiting: time spent with no well operation in progress. This can be ascribed to weather, contractors, company, etc.

Stuck pipe

Sticking occurs when the drill string cannot be moved (rotated or reciprocated) along the axis of the wellbore. During drilling operations, a pipe is considered stuck if it cannot be freed from the hole without damaging the pipe, and without exceeding the drilling rig's maximum allowed hook load. Pipe sticking can be classified under two categories: differential pressure pipe sticking and mechanical pipe sticking. There is a higher risk of sticking in high-angle and horizontal wells.

Drilling through depleted zones, where the pressure in the annulus exceeds that in the formation, might cause the drill string to be pulled against the wall and embedded in the alter cake deposited there. The internal cake pressure decreases at the point where the drill pipe contacts the alter cake, causing the pipe to be held against the wall by differential pressure. Notice that a relatively low differential pressure applied over a large working area can succeed to stick the pipe. In high-angle and horizontal wells, gravitational force contributes to extended contact between the drill string and the formation. Properly managing the lubricity of the drilling fluid and the quality of the alter cake across the permeable formation can help reduce occurrences of stuck pipe.

Methods used to get the pipe free, in addition to pulling and torqueing the pipe, include (Andrew and Sridhar, 2003) lowering hydrostatic pressure in the wellbore (Chenevert, Bourgoyne, and Millhelm, 2020) placing a spotting fluid next to the stuck zone and applying shock force just above the stuck point by mechanical jarring, or all the above. The most common approach, however, to getting free is to place a spot3 of oil, oil base mud, or special spotting fluid.

The limiting or prevention of motion of the drill string for other reasons is generally denoted mechanical sticking. Mechanical sticking can be caused by junk in the hole, wellbore geometry anomalies, cement, keyseats4 or a buildup of cuttings in the annulus.

Mechanical causes for stuck pipe include key seating from poor hole-cleaning (the cuttings settle and eventually pack around the drill string), shale swelling, wellbore collapse, plastic-owing formation (i.e. salt), bridging.

Early signals of a poor hole-cleaning conditions can be found in an erratic torque (the string is repeatedly getting stuck in the cuttings, wound up and spun free), an unexplained increase in the bottom hole pressure (which may be associated to a tight spot with packings causing restrictions further up the annulus), or an unexpected hook load (if the drill string rests on a tight packing the hook load is lower than anticipated) (Nybo, Ph.D. thesis, NTNU,, 2009).

Preventing a stuck pipe can require close monitoring of early warning signs, such as increases in torque and drag, excessive cuttings loading, tight spots while tripping, loss of circulation while drilling. Depending on what the suspected cause of sticking is, it might be necessary to increase the drilling fluid density (to stabilize a swelling shale) or to decrease it (to protect the depleted zone and avoid differential sticking).

Kick and blowout

A kick is a flow of formation fluids into the wellbore during drilling, caused by the pressure in the wellbore being less than that of the formation fluids. This can happen because the mud weight is too low with respect to the drilled formation (underbalanced kick), so that the hydrostatic pressure exerted on the formation by the fluid column is insufficient to hold the formation fluid in the formation). An induced kick occurs if dynamic and transient fluid pressure effects, usually due to motion of the drill string or casing, effectively lower the pressure in the wellbore below that of the formation. A blowout is an uncontrolled flow of reservoir fluids into the wellbore, and sometimes to the surface. A blowout may consist of salt water, oil, gas or a mixture of these. Kicks occur both in drilling and in tripping out, and if not handled properly can develop into blowouts.

Well control is the practice of preventing well flows and kicks and to maintain control of the well even in the event of such occurrences. The total pressure (e.g., mud hydrostatic pressure and casing pressure) at the hole bottom is maintained at a value slightly greater than the formation pressures to prevent further influxes of formation fluids into the wellbore (constant-bottom hole-pressure concept). Since the pressure is only slightly greater than the formation pressure, the possibility of inducing a fracture and an underground blowout is minimized. In the event of a kick, first the well must be shut in, and then the kick fluid is pumped out of the well (kick-killing), possibly using mud with increased weight (kill mud). One concern regarding this operation is the initial amount of time required to increase the mud density, during which a sticking may occur. For this reason, some well control methods start pumping out fluid immediately after the shut-in. One or two complete fluid displacements may be required to complete the procedure, depending on the method adopted. Another issue that may occur is the raising of the surface pressures to alarming heights, e.g. because of gas-volume expansion near the surface. If the kick-imposed stresses are greater than the formation can withstand, an induced fracture occurs, creating the possibility of an underground blowout. The procedure that imposes the least down hole stress while maintaining constant pressures on the kicking zone is considered the most conducive to safe kick killing.

Circulation losses

A circulation loss is the uncontrolled flow of mud into a formation, sometimes referred to as a thief zone. This usually occurs when the hydrostatic head pressure of the column of drilling fluid exceeds the formation pressure. This loss of fluid may be loosely classified as seepage losses, partial or total losses, each of which is handled differently depending on the risk to the rig and personnel and the economics of the drilling uid and each possible solution. Circulation losses may be caused by formations that are inherently fractured, cavernous, or highly permeable, or by fracturing induced by the drilling process itself (e.g., because of excessive downhole pressures, improper annular hole cleaning, excessive mud weight, shutting in a well in high-pressure shallow gas, etc.). Losing mud into the oil or gas reservoir can drastically reduce (or eliminate) the operator's ability to produce the zone. If lost circulation zones are anticipated, preventive measures should be taken by treating the mud with loss of circulation materials (LCMs), and preventive tests such as the leaked test (LOT6) and formation integrity test (FIT7) should be performed to limit the possibility of loss of circulation. If a LOT/FIT fails, a cement squeeze8 should be carried out before drilling resumes to ensure that the wellbore is competent.

In the case of severe lost circulations, the use of various plugs to seal the zone becomes mandatory. It is important to know the location of the lost circulation zone before setting a plug. Various types of plugs used throughout the industry include bentonite/diesel-oil squeeze, cement/bentonite/diesel oil squeeze, cement, barite. When a loss zone is encountered, the top priority is keeping the hole full so the hydrostatic pressure does not fall below formation pressure and allow a kick to occur. The hydrostatic pressure may be purposely reduced to stop the loss, as long as sufficient density is maintained to prevent well-control problems. Loss zones also pose a high risk of differential sticking. Rotating and reciprocating the drill string helps reduce this risk while an LCM treatment is prepared. If the location of the loss zone is known, it might be advisable to pull the drill string to a location above the affected area.

Oilfield Domain Knowledge

Daily Drilling Report

A daily drilling report is an industry-standard report of drilling activities on the drilling rig. It is a 24-hour summary report of the prior day's operations to keep the interested parties aware of the operations and issues on the rig (Bob, 2020). It is used for a variety of needs such as logging drilling data or tracking drilling performance dependent on the operator (IADC, 2018). The format of the daily drilling report has been paper form for decades resulting in historical daily drilling reports being scanned in PDF format. Daily drilling reports in recent decades have evolved to be more digital as the industry tries to move towards the goal of smart reporting (WITSML, 2017). The current generation of the daily drilling report is now a combination of automatic inputs from sensor data and manually inputted comments about the drilling activities. Daily drilling reports come in many different formats dependent on operation area, drilling type and company culture.

Some common aspects of daily drilling report relevant to this thesis are listed below (Repository, 2019).

1. **Report Date:** The daily drilling report is generated in a 24-hour period and hence this is useful identifying metadata.
2. **Well Name:** The format of the well name is dependent on the company and can be either an internal name or a legal well name but is nonetheless useful identifying metadata.
3. **Header Summaries:** The header of the report contains other useful summary information such as measured depth and true vertical depth which can be used as quality control parameters for the extracted measurements in the drilling report summary.
4. **Blowout Preventer:** Some reports have a section to record tests on this safety device which is vital to monitor lead-ups to a catastrophic failure event called a blowout.
5. **Mud:** Mud tables listing various properties of the mud in the drilling operation are useful as mud weight is an important calibration point and can be used to quality control the automated measurement extraction. Furthermore, fluid loss recorded in the mud tables may also be a measurement which is useful to validate drilling event such as losses.
6. **Time Breakdown Section:** This is the key aspect of the daily drilling report for this project. This section constitutes around a quarter of the report but takes half the effort to generate as it is done manually (WITSML, 2017). It is usually in the form of a table with varying columns dependent on the operator. The target column is the operation comments which free text comments are written by the rig supervisor summarizing the drilling activities. The operation comments are the primary target to analyze using the implementation workflow to extract calibration points.

METHODOLOGY

A lot of work has been done in the field of natural language processing and computer vision, ranging from the use of shallow learning techniques to deep learning techniques. Deep learning techniques like recurrent neural network (RNN) and convolutional neural network (CNN) have been used in the past for natural language processing activities like sentiment analysis, part of speech tagging (POS), classification etc. and computer vision activities like image sensing, voice recognition, segmentation etc.

Recurrent neural network (RNN) does natural language processing by using the principle in feed forward neural networks (FNNs), where the layers in an RNN can be categorized into input, hidden, and output layers. Then, in discrete time frames, sequences of input vectors are fed as the input, one vector at a time. In convolutional neural network, however, is employed in situations where data is or needs to be represented with a 2D or 3D data map where the proximity of data points usually corresponds to their information correlation. In a situation where the input in convolutional neural network is an image, the data map indicates that image pixels are highly correlated to their neighboring pixels. Consequently, the convolutional layers have 3 dimensions which are width, height, and depth. A major issue with the learning techniques above is that they all have long term dependencies.

Deep learning-based methods learn low-dimensional, real valued vectors for word tokens, mostly from a large data corpus, successfully capturing syntactic and semantic aspects of text. For tasks where the inputs are larger text units, e.g. phrases, sentences or documents, a compositional model is first needed to aggregate tokens into a vector with fixed dimensionality that can be used for other NLP tasks. Both recurrent and recursive models suit this purpose, however, recursive model will be used in this thesis because it considers tokens sequentially, and it combine neighbors based on the recursive structure of parse trees, starting from the leaves and proceeding recursively in a bottom-up fashion until the root of the parse tree is reached;

these have the advantage of capturing long distance dependencies and also, powerful in learning hierarchical, tree-like structure.

CRISP-DM methodology

CRISP-DM (Cross Industry Standard Process for Data Mining (CRISP-DM) is the most popular framework for executing data science projects. It provides a natural description of a data science life cycle (the workflow in data-focused projects) (Huber and Seiger, 2016).

However, this task-focused approach for executing projects fails to address team and communication issues. Thus, CRISP-DM should be combined with other team coordination frameworks. This methodology is made up of six basic phases, which include:

1. Business understanding: The Business Understanding phase focuses on understanding the objectives and requirements of the project. While many teams hurry through this phase, establishing a strong business understanding is like building the foundation of a house – absolutely essential. It focuses mainly on determining business objectives, access situations, determine project goals and produce project plan
2. Data understanding: Adding to the foundation of Business Understanding, the Data Understanding phase focuses on identifying, collecting, and analyzing data sets that can help the project. This phase also has four tasks: *Collect initial data*, *describe data*, *explore data*, and *verify data quality.*
3. Data preparation: This phase, which is often referred to as "data munging", prepares the final data set(s) for modeling. A common rule of thumb is that 50% to 80% of the project effort is in the data preparation phase. This phase has activities like selection of data, cleaning of data, construction of data, integration and formatting of data
4. Modeling: Modeling is often regarded as data science's most exciting work. In this phase, the team builds and assesses various models, often using several different modeling techniques. Although the CRISP-DM guide suggests to "iterate model building and assessment until it is strongly believing that the best model(s)" is found, in practice teams might iterating until they have a "good enough" model.

This phase has four tasks:

a. *Select modeling techniques*: Determine which algorithms to try (e.g. regression, neural net).
b. *Generate test design*: Pending your modeling approach, you might need to split the data into training, test, and validation sets.
c. *Build model*: As glamorous as this might sound, this might just be executing a few lines of code like "reg Linear Regression (). fit (X, y)".
d. *Assess model*: Generally, multiple models are competing against each other, and the data scientist needs to interpret the model results based on domain knowledge, the pre-defined success criteria, and the test design.

5. Evaluation: Whereas the Assess Model task of the Modeling phase focuses on technical model assessment, the Evaluation phase looks more broadly at which model best meets the business and what to do next. In this phase, result is *evaluated*, there is *review of process* and the next step is *determined.*
6. Deployment: A model is not particularly useful unless the customer can access its results. So, deployment basically describes what it takes to actually use the results of the project. This can be as simple as sharing a report or as complex as implementing a live real-time predictive model. This final phase has four tasks, which include, p*lanning deployment*, *Planning monitoring and maintenance*, *produce final report and review project.*

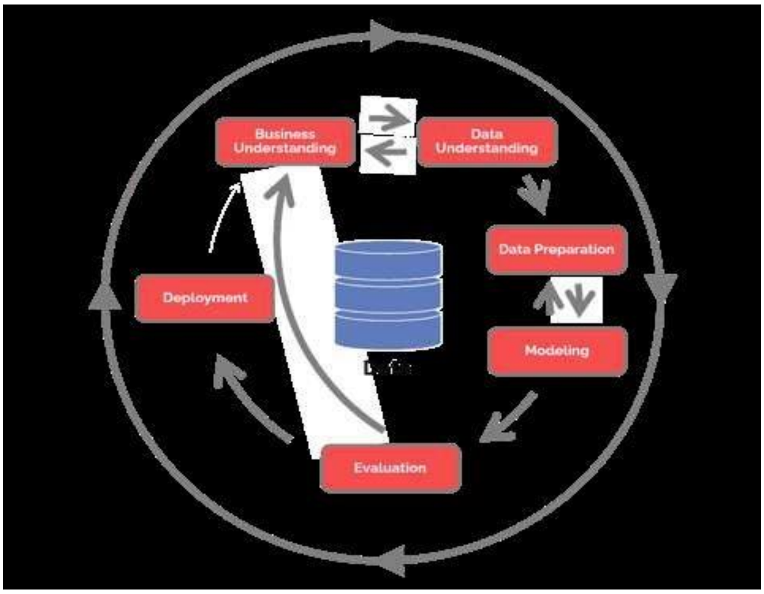

Figure 11. CRISP- DM methodology **(Wiemer & Schwarzenberger, 2017)**

Strengths and benefits of CRISP DM methodology

1. Cyclical: CRISP-DM can support the iterative nature of data science (but how to actually do iterations is not defined)
2. Adopt-able: CRISP-DM can be implemented without much training, organizational role changes, or controversy.
3. Right Start: The initial focus on Business Understanding, an often-overlooked step, is helpful to align technical work with business needs and to steer data scientists away from jumping into a problem without properly understanding business objectives.
4. Flexible: A loose CRISP-DM implementation can be flexible to provide many of the benefits of agile principles and practices. By accepting that a project starts with significant unknowns, the user can cycle through steps, each time gaining a deeper understanding of the data and the problem. The empirical knowledge learned from previous cycles can then feed into the following cycles.

This paper will adopt the CRISP DM methodology for the development of the model because, the CRISP DM provides a natural description of a data science life cycle while the agile methods attempt to minimize risk by developing software in short time boxes called iterations; where each iteration is like a miniature software project of its own, and includes all of the tasks necessary to release the mini-increment of new functionality. It also provides the platform for inheritance, encapsulation and polymorphism which makes it most fit for the research work.

RESULTS & DISCUSSION

Understanding the interplay between the number of epochs and model accuracy is crucial in avoiding overfitting and ensuring the model effectively learns patterns in the data. In the experiment carried out, a deep learning model with 100 epochs was trained. After training the model, notable trends in the loss and accuracy metrics across different epochs was observed.

Figure 20. below illustrates the trend in training and validation loss across epochs. Initially, both training and validation losses decreased consistently, indicating the model's learning process. However, beyond 50 epochs, the training loss continued to decrease, while the validation loss showed a slight increase, signifying potential overfitting.

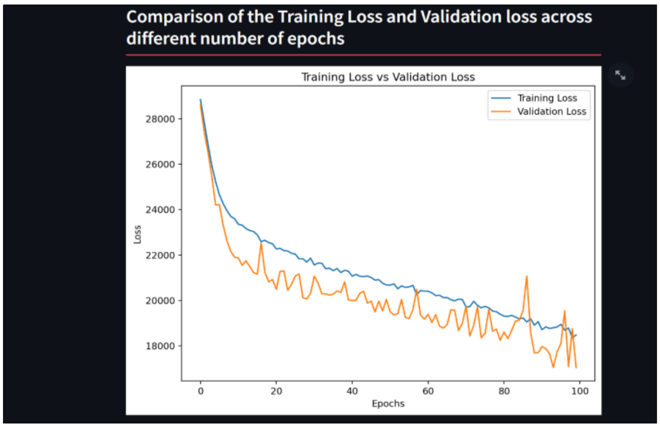

Figure 20: Comparison of Training and Validation Loss

The comparison between training loss and validation loss (Figure 20) is crucial in assessing the model's performance. When the training loss continues to decrease while the validation loss increases, it shows the model is becoming too specific to the training data, potentially leading to good generalization on new data.

The scatter plot in Figure 21 demonstrates the relationship between predicted and actual values. A closer clustering of points around the diagonal line represents more accurate predictions.

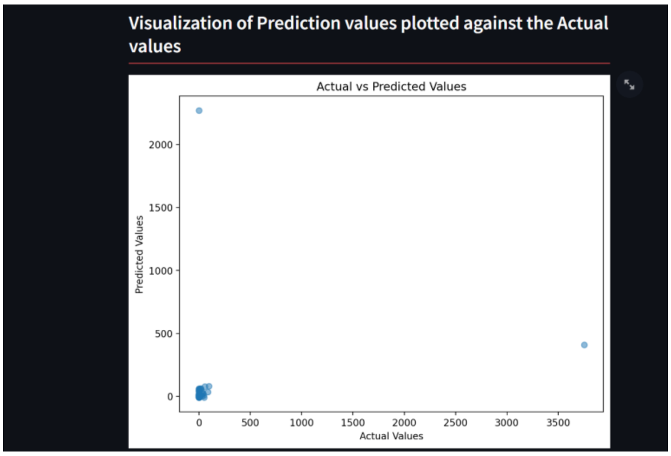

Figure 21. Scatter plot of Predicted vs Actual Values

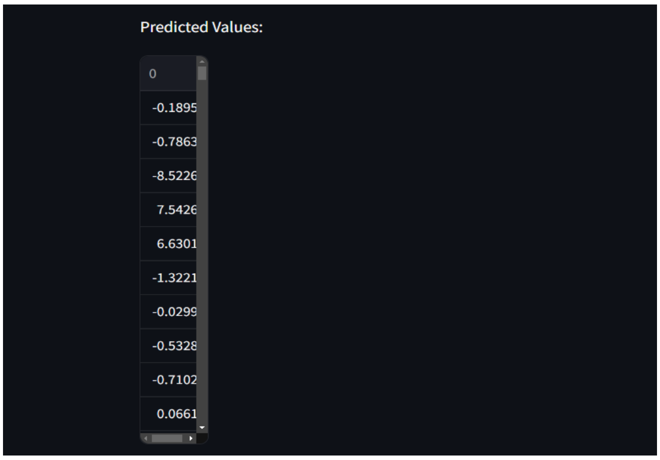

Figure 22. Prediction Values gotten from the Model

Sample of Predicted vs Actual Values:

	Actual	Predicted
132	0	-0.4202
112	0	7.3113
62	0	2.7758
57	0	1.8002
93	0	2.2347
218	0	-3.929
273	0	1.8141
47	0	5.191
258	0	-0.7629
1	0	-0.7863

Figure 23. Predicted values and the Actual values

The predicted values (Figure 23.) obtained from the model hold significant relevance in drilling operations. These predictions can guide decision-making in adjusting drilling parameters like pressure, torque, or mud volume to optimize the Rate of Penetration (ROP) during drilling. Moreover, by analyzing the predicted values against actual parameters (Figure 23.), engineers can pinpoint optimal calibration points for efficient drilling activities. *In this AI-driven drilling operation focusing on regression (predicting ROP), the AUC-ROC plot, which is mainly/commonly used in classification tasks, is not applicable in this prediction task. AUC-ROC is more relevant in classification problems, helping to evaluate a model's performance in distinguishing between classes, which doesn't directly apply in our regression context.*

These observations emphasize the need for careful model tuning, monitoring loss trends, and interpreting predicted values in optimizing drilling operations for efficiency and accuracy. *The predicted values derived from our model hold paramount significance in optimizing drilling operations. These predicted values serve as pivotal indicators guiding decision-making and operational adjustments in real-time drilling.*

In optimizing drilling operations, the predicted values, particularly those related to Rate of Penetration (ROP), plays a critical role in optimizing drilling parameters. By leveraging these predictions, drilling engineers can fine-tune various operational factors like:

Pressure and Torque: Here, the predicted ROP values assist in optimizing pressure and torque on the drilling equipment. Adjusting these parameters based on predicted values ensures the machinery operates within optimal ranges, enhancing drilling efficiency while preventing equipment damage.

Mud Volume and Composition: Here, the model's predictions enable precise adjustments in mud volume and composition. Engineers can now tailor the mud's properties, including density and viscosity, to match predicted ROP values, thereby facilitating smoother drilling processes.

Rotational Speed and Weight on Bit: Here, the predicted ROP values inform decisions on the rotational speed and weight applied on the drill bit. Aligning these parameters with predicted values aids in maintaining optimal drilling rates without compromising tool integrity.

Calibration Point Identification: utilizing predicted values against actual drilling parameters allows for the identification of calibration points. By comparing predicted ROP values with observed values during drilling operations, engineers can discern optimal calibration settings for the drilling equipment. This process aids in establishing precise operational configurations for enhanced drilling efficiency.

In Conclusion we can say that, with the integration of predicted values into drilling operations offers a strategic advantage by enabling proactive adjustments and optimizations.

These values serve as actionable insights, empowering drilling engineers to fine-tune operational parameters, identify optimal calibration settings, and ensure efficient, speedy and productive drilling operations.

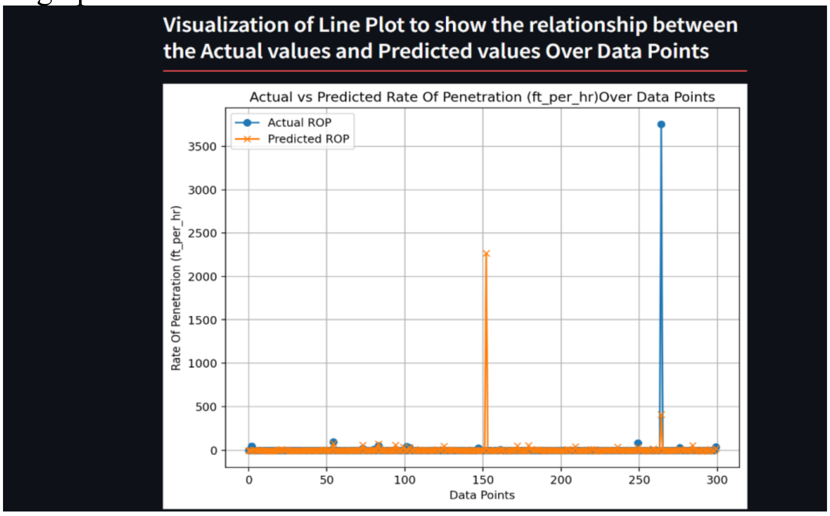

Figure 24. Model predictive performance

The Line Plot (Figure 4.10) visualizes the model's predictive performance by demonstrating how closely the predicted ROP aligns with the actual ROP values. The consistent alignment between the two (Actual values and the Predicted values) lines indicates a strong model that accurately predicts ROP across the dataset. This line plot provides an intuitive way for users and experts to observe and interpret the relationship between predicted and actual values, thereby gaining insights into the model's performance in predicting ROP for drilling operations.

CONCLUSION

Conclusively, the research in trying to answer the questions outlined in chapter one, took an in-depth view of all the key components of the system architecture "the development of recursive deep learning based natural language processor for crude oil drilling operations". The components are all inter-related and worked together to achieve the aim of extracting calibration points from daily oil drilling document to make informed predictions on drilling processes; this is geared towards increasing the accuracy of the model and mitigating the risks of non-productive time in drilling events thereby increasing the efficiency of the drilling operation. The algorithm is recursive, ensuring accurate result which is visually represented through an appropriate visualization for ease of decision making.

REFERENCES

Abdi, A., Hasan, S., Shamsuddin, S., Idris, N., & Piran, J. (2021). A hybrid deep learning architecture for opinion-oriented multi-document summarization based on multi-feature fusion. *Knowledge-Based Systems*, 106658.

Afsari, M. (2009). An effective tool for borehole stability analysis and managed pressure drilling. *SPE Middle East Oil and Gas Show and Conference.Society of Petroleum Engineers*, (pp. pp. 15–18). Manama.

Akgun, F. (2012). How to estimate the maximum achievable drilling rate without jeopardizing safety. *SPE 78567, Abu Dhabi Inter. Pet. Exh. And Conf.*, (pp. 32-40). Abu Dhabi.

Aleksandrs, E., Cornelia, F. & Yiannis, A. (2014). Shadow free segmentation in still images using local density measure. *In Proceedings of the 2014 IEEE International Conference on Computational Photography (ICCP).*, (pp. 1–8.).

Andor, D., Alberti, C., & Weiss, D. A. (2016). Globally normalized transition-based neural networks. *preprint arXiv:1603.06042*.

Andrew, G. & Sridhar, M. (2003). Recent advances in hierarchical reinforcement learning: Discrete event dynamic systems. *Journal of Information System*, 2330.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., & Batra, D. (2015). VQA: Visual question answering. In: *Proceedings of the IEEE International Conference on Computer Vision*, (p. 2433).

Athithan D, S. (2019.). Information extraction from daily drilling reports using machine learning. *IEEE*, 15-19.

Barragan, R. & Santos, O. (2018). Optimization of multiple bit runs. *SPE/IADC Drilling Conference*, (pp. 18-26). Amsterdam.

Bengio, Y. & Ducharme, R. (2003). A neural probabilistic language model. *JMLR*, 3, 1137-1155.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3(2), 1137–1155.

Ben-Younes, H., Cadene, R., Thome, N., & Cord, M. (2019). Bilinear superdiagonal fusion for visual question answering and visual relationship detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 8102–8109.

Bilal, E. & Al, E. (2010). Automated operations classification using text mining. *The 3rd International Conference on Computational Intelligence and Industrial Application.* Huazhong: Research Gate.

Bilal, E., & *et al.* (2012). A hybrid multiple classifier system for recognizing usual and unusual drilling events. *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, (pp. 1754–1758).

Blick, F., & Bizanti, M. (1999). Fluid dynamics of wellbore bottom hole cleaning. *SPE,* 12888, 12-24.

Bob, M. (2020). Daily drilling report. *Malone Petroleum Consulting*.

Bourgoyne, A., & Young, F. (1974). A multiple regression approach to optimal drilling and abnormal pressure detection. *SPE 4238*, 20-25.

Bourgoyne, T. & Young, F. (2018). A multiple regression approach to optimal drilling and abnormal pressure detection. *SPE 4238*, 4238.

Bowman, S., Potts, S., & Manning, C. (2014). Recursive neural networks forlearning logical semantics. *CoRR*, 45-60.

Brulé, M. I. (2015). How big data technologies advance data management and analytics in E&amp. *SPE Digital Energy Conference and Exhibition.* (pp. 3– 5.). Woodlands: doi: https://doi.org/10.2118/173445-MS. url: https://www.onepetro.org/conference-paper/SPE-173445-MS.

Cambria, E. A. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2), 48–57.

Cambria, E. P. (2017). Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32(6), 74–80.

Carreira, J. & Sminchisescu, C. (2010). Constrained parametric min-cuts for automatic object segmentation. *In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 3241–3248).

Cayeux, E. & Dvergsnes, E. (2009). Real-time optimization of the drilling process – challenges in industrialization. *SPE/IADC , SPE/IADC Drilling Conference and Exhibition*, (p. 119650). Amsterdam, The Netherlands.

Cayeux, E. F. (2016). Monitoring and Control of Drilling Utilizing Continuously Updated Process models. *SPE 99207, IADC/SPE Drilling Conference*, (pp. 51-65). Miami, Florida.

Cecilia, Q. (2003). Language ambiguity: A curse and a blessing. *Transl. Journal*, 7(1), 45-68.

Chenevert, M., Bourgoyne, A. & Millhelm, K. (2020). *Applied drilling engineering*. In R. SPE. Texas.

Cheng, Y., Yu, F., Feris, R., Kumar, S. & Choudhary, A. (2015). An exploration of parameter redundancy in deep networks with circulant projections. *In Proceedings of the IEEE International Conference on Computer Vision*, (pp. 2857–2865.).

Chia, C. & Monden, T. (2017). Operation support centers: Real time drilling optimization and risk mitigation. *SPE 110950, SPE Saudi Arabia Technical Symposium*, (pp. 4-19). Symposium, Dhahran.

Christine N, J. S. (2019.). A brief survey of text mining applications for the oil and gas industry. *International Petroleum Technology Conference.*, (p. 2523). Beijing.

Ciresan, Dan, Ueli, M., Jonathan, M., & Luca, G. (2011). High performance convolutional neural networks for image classification. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, (pp. 1237–1242).

Collobert, R. & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. *ICML*, 160-167.

Dan, S., Nybo, R. & Giulio, G. (2011). Ensemble methods for process monitoring in oil and gas industry operations. *Journal of Natural Gas Science and Engineering* , 748-753.

Dan, S., Nybo, R. & Vahid, A. (2013). Real-time optimization of rate of penetration during drilling operation. *10th IEEE International Conference on Control and Automation (ICCA) (Hangzhou, China)*, (pp. 357-362.). China.

Das, A., Agrawal, H., Zitnick, L. P. & Batra, D. (2017). Human attention in visual question answering: Do humans and deep networks look at the same regions? *Journal of Computer Vision and Image Understanding*, 163, 90–100.

David, B., & *et al.* (2015). *Natural language processing for extracting conveyance.* Retrieved from https://patents.google.com/patent/US9251139B2/en.

David, M., Charless, F. & Jitendra, M. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5), 530–549.

Deshmukh, R. A. (2020). Deep learning techniques for part of speech tagging by natural language processing. *2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, (pp. 76–81).

Dubinsky, V. & Baecker, D. (2014). An interactive drilling dynamics simulator for drilling optimization and training. *SPE Annual Technical Conference*, (pp. 2230). New Orleans, September 2014.

Dupriest, F. & Koederitz, W. (2015). Maximizing drill rates with real-time surveillance of mechanical specific energy. *IADC/SPE 92194, Drilling Conference*, (pp. 45-60). Amsterdam.

Erhan, D., Courville, A., Bengio, Y. & Vincent, P. (2010). Why does unsupervised pre-training help deep learning. *JMLR*, 11.

Fear, M. (2018). How to improve rate of penetration in field operations. *SPE 55050, IADC/SPE Drilling Conference*, (pp. 30-36). New Orleans.

Frome, A., Corrado, G., & Shlens, J. (2013). A deep visual-semantic embedding model. *NIPS*, 34-75.

Garnier, J., & Van, L. (2022). Phenomena affecting drilling rates at depth. *Annual Fall Meeting of SPE, Houston, TX*, 1097.

Geng, Z., Chen, G., Han, Y., & Lu, G. &. (2020.). Semantic relation extraction using sequential and tree-structured lstm with attention. *Journal of Information Science*, 509, 183–192.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 345–420.

Goobie, R. & Tollefsen, E. (2016). Optimize drilling and reduce casing strings using remote real-time well hydraulic monitoring. *SPE 103936, First International Oil Conference and Exhibition*, (pp. 18-32). Cancun, Mexico.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B. & Warde-Farley, D. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, (pp. 2672–2680).

Graham, W., & Muench, L. (2020). Analytical determination of optimum bit weight and rotary speed combinations. *SPE 1349-G, Fall Meeting of the Society of Petroleum Engineers, Dallas, TX*, 1349.

Greenwald, T. (2018). https://www.wsj.com/articles/what-exactly-is-artificialintelligence-anyway. *Wall Street Journal Online Article*.

Gulsrud, T., Nyb, R. & Rkevoll, K. (2009.). Statistical method for detection of poor hole cleaning and stuck pipe, SPE Offshore Europe & Gas Conference & Exhibition. *Society of Petroleum Engineers*, (pp. 8-11). Aberdeen, U.K.

Hinton, G., Deng, L., Yu, D. & Dahl, A. (2012). Deep neural networks for acoustic modeling in speech recognition. *The Shared Views of Four Research Groups. IEEE Signal Process*, 29(6), 82-97.

Huang, Z., Xu, W. & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *preprint arXiv:1508.01991*.

Huber, S. & Seiger, R. (2016). Goal-based semantic queries for dynamic processes in the internet of things. *Int. Journal of Semantic Computing*, 10(2), 269.

IADC. (2018). Daily drilling report based on sensor data. Tech. rep. *International Association of Drilling Contractors*, 20-29.

Iqbal, F. (2008). Drilling optimization technique: Using real time parameters. *SPE 114543, SPE Russian Oil & Gas Technical Conference and Exhibition*, (pp. 28-30). Moscow, Russia.

Iversen, F., & *et al.* (2008). Offshore field test of a new integrated system for realtime optimization of the drilling process. *IADC/SPE 112744, IADC/SPE Drilling Conference held in Orlando*, (pp. 4-6). Florida, USA.

Jeffrey, P., Socher, R. & Christopher, M. (2014). Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Joao, C. & Cristian, S. (2010). Constrained parametric min-cuts for automatic object segmentation (CVPR). *In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, 3241–3248.

Jonathan, B. & Jitendra, M. (2015). Shape, illumination, and reflectance from shading. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8), 1670–1687.

Julio, H., & *et al.* (2018). Sequence mining and pattern analysis in drilling report with natural language processing. *Society of Petroleum*, 24–26.

Karras, T., Aila, T., Laine, S. & Lehtinen, J. (2017). Progressive growing of GANs for improved quality, stability and variation. *arXiv preprint arXiv:1710.10196*, 45-86.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *International Journal of Information Research*, 23-66.

Krizhevsky, A. & Sutskever, I. (2012). Imagenet classification with deep convolutional neural networks. *NIPS*, 55-66.

Krizhevsky, A., Sutskever, I. & Hinton, G. (2012.). Imagenet classification with deep convolutional neural networks. *In: Advances in Neural Information Processing Systems*, 1097–1105.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *NIPS*, 35-56.

Leonard, K. (2016). The process of building a mechanical earth model using well data. *Leoben*, 34-50.

Li, J., Sun, A. & Han, J. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 58-100.

Lipton, Z., Berkowitz, J. & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *International Journal of Information Research*, 35-55.

Lummus, L. (2019). Drilling optimization. *SPE 2744, SPE 40th Annual California Fall Meeting, San Francisco*, 26-36.

Lummus, L. (2022). Acquisition and analysis of data for optimized drilling. *Journal of Petroleum Technology*, 16-28.

Maidla, E., & Ohara, S. (2022). Field verification of drilling models and computerized selection of drill bit, WOB, and drillstring rotation. *SPE Drilling Engineering*, 189-195.

Malinowski, M., Rohrbach, M. & Fritz, M. (2015). Ask your neurons: A neural-based approach to answering questions about images. *In Proceedings of the IEEE International Conference on Computer Vision*, 1–9.

Maurer, C. (2013). The "perfect-cleaning" theory of rotary drilling. *Journal of Pet. Tech.*, 23-38.

Mikolov, T., Karafi, M., Burget, L., Cernock, J. & Khudanpur, S. (2010). Recurrent neural network based language model in Interspeech. *Journal of Computer Science*, 2, 3.

Mikolov, T., Karafia, M., Burget, L. & Ernocky, J. (2010.). Recurrent neural network based language model. *In Eleventh Annual Conference of the International Speech Communication Association*, (pp. 120-185).

Mikolov, T., Karafia, M., Burget, L., Ernocky, J. & Khudanpur, S. (2010). Recurrent neural network based language model. *in Eleventh Annual Conference of the International Speech Communication Association*, (pp. 46-84).

Millheim, K., & Gaebler, T. (2018). Virtual experience simulation for drilling: The concept. *The Concept," SPE 52803, SPE/IADC Drilling Conference, Amsterdam, Holland*, (pp. 24-30). Amsterdam, Holland.

Milter, J. & Bergjord, O. (2016). Use of real-time data at the statfjord field anno. *SPE 99257, SPE Intelligent Energy Conference and Exhibition*, (pp. 28-32). Amsterdam, The Netherlands.

Mochizuki, S. & Saputelli, L. (2014). Real time optimization: Classification and assessment. *SPE 90213, SPE Annual Technical Conference and Exhibition*, (pp. 10-22). Houston, TX.

Moussa, M., & Al-Betairi, A. (2014). Multiple regression approach to optimize drilling operations in the Arabian gulf area. *SPE 13694, Middle East Oil Symposium, Bahrain*, 15-25.

Nekson, E. (2016). *Using graphical algorithm to identify risk and provide early arning* . Retrieved from Using graphical algorithm to identify risk and provide early arning: https://patents.google.com/patent/US552548B1/en Nybo, R. (2009). *Ph.D. thesis, NTNU.*

Nybo, R., & Sui, D. (2014). Closing the integration gap for the next generation of drilling decision support systems. *SPE Intelligent Energy Conference,Society of Petroleum Engineers*, (pp. 34-40). Netherland.

Osgouei, R. (2017). *Rate of penetration estimation model for directional andhorizontal wells.* Thesis, The Graduate School, Middle East Technical University.

Ozbayoglu, E. & Omurlu, C. (2005). Minimization of drilling cost by optimization of optimization of the drilling parameters. *15th International Petroleum Gas Congress and Exhibition of Turkey*, (pp. 23-34). Turkey.

Ozbayoglu, E., & Miska, Z. (2004). Analysis of the effect of major drilling parameters on cuttings transport efficiency for high-angle wells in coiled tubing drilling operations. *SPE, SPE/IcoTA CT Conf. and Exhb.,* , (p. 89334). Houston.

Pereira, J. (2015). *Comprehensive optimization of drilling parameters for horizontal wells*. Ph.D. Thesis, The Graduate School, University of Tulsa. UK.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*.

Rao, G., Huang, W., Feng, Z., & Cong, Q. (2018). LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*, 49–57.

Reed, L. (2015). A monte carlo approach to optimal drilling. *SPE 3513, SPE Annual Fall Meeting, New Orleans*, 12-20.

Reid, I. & John, Z. (2012). Optimized decision making through real time access to drilling and geological data from remote wellsites. *SPE Asia Pacific Oil and Gas Conference and Exhibition*, (pp. 11-20). Melbourne, Australia,.

Repository, N. D. (2019). *National data repository*. Retrieved 08 06, 2023, from https://en.wikipedia.org/wiki/National_Data_Repository.:

Rommetveit, R., & Bjorkevoll, S. (2014). Drilltronics: An integrated system for realtime optimization of the drilling process. *IADC/SPE 87124,IADC/SPE Drilling Conference*, (pp. 10-22). Dallas, Texas.

Rumelhart, E., Hinton, G., & Williams, J. (1986). Learning representations by backpropagating errors. *Journal of Information and Computer System*, 36-56.

Samuel, G., & Miska, S. (2000). Optimization of drilling parameters with the performance of multilobe positive displacement motor (PDM). *SPE IADC/SPE Asia Pacific Drilling Conference, Jakarta, Indonesia, September 1998*, (pp. 30-40).

Samuel, O. (2012). *Practical approach to solving wellbore instability problems*. Tech. Norman, Oklahoma, USA.

Schmidhuber, S. H. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Sethupathi, A. & Shebi, R. (2017). Augmented text mining for daily drilling reports using topic modeling and ontology. *Society of Petroleum Engineers*, 18-24.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., & Wang, W. (2017). Learning from simulated and unsupervised images through adversarial training. *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 2107–2116).

Silje, S. (2012). *Improved drilling process through the determination of hardness and lithology boundaries.* Norway.

Simmons, L. (2022). A technique for accurate bit programming and drilling performance optimization. *IADC/SPE 14784, Drilling Conference*, (pp. 2240). Dallas, TX.

Sivarajah, V., Kamal, M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70, 263–286.

Socher, R. & Huval, B. A. (2012). Semantic compositionality through recursive matrix-vector spaces. *In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, (pp. 1201–1211).

Socher, R., Lin, C. & Manning, C. (2011.). Parsing natural scenes and natural language with recursive neural networks. *in Proceedings of the 28th international conference on machine learning*, (pp. 129–136.).

Socher, R., Perelygin, A., Wu, J. & Chuang, J. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 16(31), 16-42.

Sohangir, S. W. (2018). Deep learning for financial sentiment analysis. *Journal of Big Data*, 1-5.

Speer, W. (2022). A method for determining optimum drilling techniques. *Drill and Prod. Prac. API*,.

Srivastava, N. & Salakhutdinov, R. (2012). Multimodal learning with deep boltzmann machines. *NIPS*, 45-63.

Stefan, M. (2019). Developments in petroleum engineering. *Collected Works of Arthur Lubinski*, 2, 266-275.

Strathman, M. & Elley, D. (2017). Time-based real-time-drilling-operations excellence delivered. *SPE 107303, SPE Digital Energy Conference and Exhibition*, (pp. 50-56). Houston, TX.

Surveys, A. C. (2016). Computer vision and natural language processing. *ACM Computing Surveys*, 4-10.

Tang, D., Qin, B. & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. *In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (pp. 1422–1432).

Tansev, E. (2016). A heuristic approach to drilling optimization. *SPE Annual Fall Meeting, Dallas, TX*, 15-21.

Tavaf, N., Torfi, A., Ugurbil, K., & Van de Moortele, F. (2021). GRAPPA-GANs for parallel MRI reconstruction. *arXiv preprint arXiv:2101.03135, Jan 2021*, 2565.

Ursem, L. & Williams, H. (2013). Real time operations centers: The people aspects of drilling decision making. *SPE/IADC 79893, SPE/IADC Drilling Conference*, (pp. 32-40). Amsterdam, Netherlands.

Valacich, G. (2020). Essentials of system analysis and design. *Journal of Information Science*, 26-45.

Vincent, P., Larochelle, H. & Bengio, Y. (2008). Extracting and composing robust features with denoising autoencoders. *ICML*, 23-40.

Wardlaw, H. (2000). Drilling performance optimization and identification of overpressure formations. *SPE 2388, The Uni. Of Texas Austin, TX*, 16-26.

Warren, M. (2016). Penetration-rate performance of roller–cone bits. *SPE Annual Technical Conference, Houston*, 13-25.

Wiemer, H., & Schwarzenberger, M. (2017). Holistic and DoE-based approach to developing and putting into operation: Complex manufacturing process chains of composite components. *Procedia CIRP*, 66, 147-152.

Wilson, D., & Bentsen, R. (2022). Optimization techniques for minimizing drilling costs. *SPE 3983, 47th SPE Annual Fall Meeting, San Antonio*, 23-30.

WITSML. (2017). *The solution lean automated reporting.* Energistics.

Witt, J., & Remmert, S. (2017). Implementation of ROP management process in Qatar North Field. *SPE 105521, SPE/IADC Drilling Conference*, (pp. 42-48). Amsterdam.

Wojtanowicz, A. & Kuru, E. (2000). Minimum-cost well drilling strategy using dynamic programming. *Journal of Energy Resources Technology*, 16-20.

Woods, A. & Galle, M. (2019). Best constant weight and rotary speed for rotary rock bits. *Drill. and Prod. Prac.*, 48-73.

Xue, X. A. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics*, 47, 101-235.

Yu, L., Zhang, W., Wang, J. & Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. *Thirty-First AAAI Conference on Artificial Intelligence*, 20-46.

Zeng, D., Liu, K., Lai, S., Zhou, G. & Zhao, J. (2014.). Relation classification via convolutional deep neural network. *in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, (pp. 2335–2344).

Zhao, R., & William, G. (2002). Bridging the semantic gap in image retrieval. Distributed multimedia databases: Techniques and applications. *Journal of Computer and Information System*, 14–36.