Journal of Petroleum Science and Engineering xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Petroleum Science and Engineering

journal homepage: http://www.elsevier.com/locate/petrol

Training-while-drilling approach to inclination prediction in directional drilling utilizing recurrent neural networks

Andrzej T. Tunkiel[a,*], Dan Sui[a], Tomasz Wiktorski[b]

[a] Department of Energy and Petroleum Engineering, Faculty of Science and Technology, University of Stavanger, 4036 Stavanger, Postboks, 8600, Forus, Norway
[b] Department of Electrical Engineering and Computer Science, Faculty of Science and Technology, University of Stavanger, 4036 Stavanger, Postboks, 8600, Forus, Norway

ARTICLE INFO

Keywords:
Directional drilling
Machine learning
Continuous learning
Inclination prediction

ABSTRACT

Machine Learning adoption within drilling is often impaired by the necessity to train the model on data collected from wells analogous in lithology and equipment used to the well where the model is meant to be deployed. Lithology information is not always well documented and fast-paced development of drilling equipment complicates the challenge even further, as a model would likely become obsolete and inaccurate when new technologies are deployed. To bypass this problem a training-while-drilling method utilizing neural networks that are capable of modelling dynamic behaviour is proposed. It is a continuous learning approach where a data-driven model is developed while the well is being drilled, on data that is received as a continuous stream of information coming from various sensors. The novelty in presented approach is the use of Recurrent Neural Network elements to capture the dynamic behaviour present in data. Such model takes into account not only values of the adjacent data, but also patterns existing in the data series. Moreover, results are presented with a focus on the continuous learning aspect of the method, which was sparsely researched to date. A case study is presented where inclination data is predicted ahead of the inclination sensor in a directional drilling scenario. Our model architecture starts to provide accurate results after only 180 m of training data. Method, architecture, results, and benchmarking against classical approach are discussed; full dataset with complete source code is shared on GitHub.

Credit statement

Andrzej Tunkiel, Conceptualization, Methodology, Software, Data curation. Dan Sui, Formal analysis, Writing - review & editing, Supervision. Tomasz Wiktorski, Writing - review & editing, Supervision

1. Introduction

Lack of adequate training data is one of the major issues preventing machine learning model deployment within petroleum. While in general it is relatively easy to develop data-driven models for problems like rate of penetration (ROP) prediction, such models will be valid only for wells where geology, equipment, and general design matches closely the training dataset. This is further corroborated by the lack of published general-purpose data-driven ROP prediction models. All machine learning models face such challenge; if an algorithm is trained to detect cats, but the dataset contains only cats indoors, it will struggle to classify pictures taken outdoors. Such problem was explored in practice when a neural network was trained to discern dogs from wolves. Training dataset was made flawed on purpose, where pictures of dogs were taken on grass, and pictures of wolves in the snow. This lead to the classifier using snow as the key feature, and subsequently poor model performance (Ribeiro et al., 2016).

To solve this underlying issue, continuous learning (Liu, 2017) methods could be used, where a model is continuously retrained while the well is being drilled. Data collected from a drilled section are used to train a model that can be applied to the further section of the same well. When additional section of a well is drilled, the process is repeated to create an updated model. Advances in computational power make data-driven model training time negligible in comparison to time required to drill a well making the training-while-drilling approach feasible. Model training is often fast enough to be completed in the short breaks in the drilling process, such as adding a stand to the drillstring. Additional benefit of a dynamically trained model is that any

* Corresponding author.
E-mail address: andrzej.t.tunkiel@uis.no (A.T. Tunkiel).
URL: http://www.ux.uis.no/%7Eatunkiel/ (A.T. Tunkiel).

https://doi.org/10.1016/j.petrol.2020.108128
Received 24 June 2020; Received in revised form 24 September 2020; Accepted 9 November 2020
Available online 14 November 2020

discrepancies between predictions and incoming data are used as a feedback to improve the subsequent iteration of the model.

There is limited previous research related to such approach in drilling. Data-driven rate of penetration (ROP) prediction models are abundant in the latest literature: (Ahmed et al., 2019a, 2019b; Hegde and Gray, 2017, 2018; Hegde et al., 2015; Soares and Gray, 2019a; Sabah et al., 2019; Han et al., 2019; Shi et al., 2016; Mantha and Samuel, 2016; Eren and Ozbayoglu, 2010; Soares et al., 2016; Amar and Ibrahim, 2012; Eren and Ozbayoglu, 2010, 2010; Yi et al., 2014; Jiang and Samuel, 2016). There is however limited work related to continuously expanding dataset. Only few papers were identified where the training to testing ratio was explored showing improvement over analytical methods even at smallest training datasets (Hegde et al., 2017). Application of continuous expanding of the training dataset was researched as well in other papers, such as (Hegde and Gray, 2017), applying random forest algorithm to again predict the ROP, and (Soares and Gray, 2019b), where expanding dataset was used for ROP prediction implemented as changing train/test ratio, evaluating random forest, support vector machines and neural networks, and comparing it to analytical models such as Bingham, and Bourgoyne and Young. No other analysis of continuous learning in drilling environment was identified.

To expand on this existing work, a novel model was developed that uses not only the real-time attributes as inputs from a specific time and space, but also utilizes previous values; this is what this paper refers to as dynamic behaviour. It means that the model is aware of not only the current state, but also of the previous values and how they change along the data series, be it space or time, identifying the dynamics of the local environment. This is achieved through the use of Recurrent Neural Network (RNN) (Rumelhart et al., 1986), where attribute values are fed to the network from multiple steps along the data series.

To the best of our knowledge, no drilling related continuous learning research was done that utilized the recurrent neural networks the way this paper proposes. This paper also performs a thorough analysis of how the models' performance change as the data is continuously acquired; we were unable to identify any drilling-related paper that would discuss this aspect in a comparable detail.

While our novel approach does not produce results from the first meter drilled, it requires relatively small dataset to start working reliably. A case study is presented where lagging inclination data is predicted in a directional drilling scenario using a bent sub. It was selected because the problem is sparsely explored in the existing research, and the way that data behaves makes it a good candidate for a neural network model with recurrent elements. The applied model is based on our earlier work (Tunkiel et al., 2020a) where the problem of predicting lagging inclination data was first explored. In this paper, accuracy along the depth of the well is explored to evaluate method's usefulness and applicability in real-life situations.

1.1. Motivation

In the recent years, directional drilling became one the common drilling methods, especially in relation to shale developments (Wang et al., 2018). Precise well placement is an important factor when it comes to the future well performance. Directional driller depends on the values from downhole sensors to know where the well is being placed. One of the challenges is, that due to space constraints, those sensors are at a significant distance from the bit, often tens of meters. This in turn creates a blind zone, a section of a well that is drilled, but the driller does not know where it exactly is, potentially leading to a delayed corrective actions.

As the sensor data is delayed, decisions taken based on these sensors' readings are delayed as well, leading to suboptimal well placement. With pay zones only 5–15 m thick, as in case of the Bakken field (Zou, 2017), minimizing that delay distance in the directional readings is critical. The goal of this case study is to predict such continuous inclination readings that are yet to be made, predicting the well direction between the sensor and the bit.

1.2. Innovation

There are a number of innovative elements in the presented paper. Only one prior published study was identified discussing the recreation of sensor data using machine learning methods, apart from parts of the proposed method presented by the authors on the 39th *International Conference on Ocean, Offshore & Arctic Engineering* in August 2020 (Tunkiel et al., 2020a). Presentation was given (Koryabkin et al., 2019) on similar topic applying basic regression algorithms, lasso, ridge, random forest and gradient boosting, to predict a number of sensor values lagging behind the bit. Achieved results showed relative error less than 16% for 80% of the tested data. Our research uses more advanced network architecture as well as is considered within continuous learning environment. Applying machine learning allows for method deployment when prior specific knowledge of the bit steering mechanism is not necessary. Such exact information is on the other hand needed to follow recently published analytical approach, such as performed by (Wang, 2017; Wang et al., 2020), where beam bending model is developed based on exact bottom hole assembly geometry and function.

Another key innovative element presented in the case study is the application of *continuous learning*. This concept is related to lifelong machine learning (Liu, 2017), where continuously expanding training dataset is used to evaluate samples from the immediate future. While there is significant research related to data stratification, i.e. the split ratio between training and testing datasets, such as (Anifowose et al, 2011, 2017), it must be noted that this is a similar, yet different topic. Continuous learning mimics the real life learning, where immediate future is predicted using all the past experiences, while stratification studies consider a fixed dataset and the best way to split it. Presented case study focuses of the models' performance in the continuous learning scenario in detail, which we were unable to identify in literature.

Lastly, inclusion of past values as inputs via use of recurrent neural networks is also a topic sparsly explored in research related to drilling. Publications related to flow rate estimation (Chhantyal et al., 2018) utilized generic recurrent neural networks, as well as newer work on kick detection (Osarogiagbon et al., 2020) utilized newer architecture of Long-Short Term Memory. Our work expands on this by utilizing Gated Recurrent Units, RNN cell first discussed in 2014 (Chung et al., 2014) in a continuous learning scenario, a combination that we were unable to identifiy in literature related to drilling.

The proposed solution is fast to deploy, requires no proprietary software and can be run using any modern consumer-grade Graphics Processing Unit (GPU), making the necessary investment very low. Properly set up system automatically adapts to available data through dimensionality reduction techniques discussed in the further chapters. A single well data is required to validate the method for a given use case. Given the auxiliary nature of the generated results, there is little to no risk in deploying the presented method to the field. The accuracy of the method can be continuously monitored, since true values are measured with 23 m lag relative to the prediction.

1.3. Machine learning methods used

Machine learning can be applied in various ways. Generally speaking, an algorithm learns the correlations between inputs and outputs that can later be exploited for prediction purposes. One of the methods of implementing this is to use data from a given moment in time to predict a different, unknown parameter. For example, weight on bit (WOB) and drill bit's rpm can be correlated with rate of penetration, so that optimization can be done on the developed model to maximize the ROP. That correlation can be captured using various algorithms, such as linear regression, decision trees, neural networks, gradient boosting and others. This approach will however not capture any dynamic behaviour

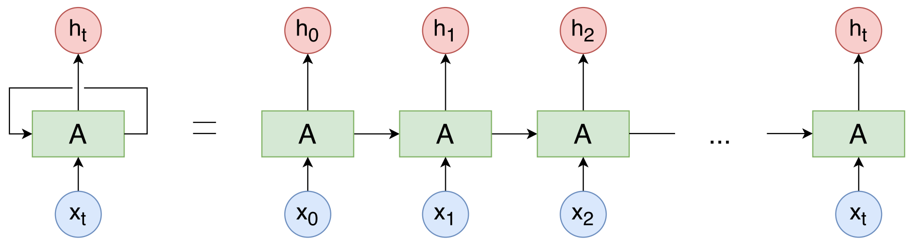


Fig. 1. Basic Recurrent Neural Network schematic.

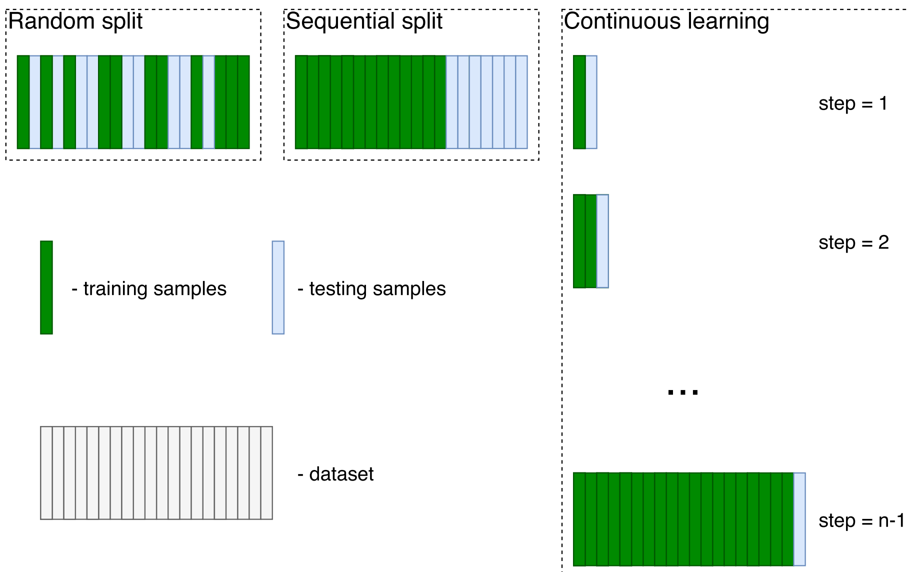


Fig. 2. Train/test split strategies.

of the model. This can be rectified partially by calculating derivatives of inputs, but it will have a very limited impact. To fully capture dynamic behaviour of a given model Recurrent Neural Networks (Rumelhart et al., 1986) are used. This is an architecture suited for data-series, such as speech, language processing, or drilling logs. Its internal structure is well suited to take inputs both from the current state as well as a number of previous states. It contains a connection that feeds the output from step *t-1* to step *t*. The basic principle is shown in Fig. 1 on the left hand side. Practically this type of network is implemented in an unfolded form, seen on the right hand side. Input $x_0$generates output h_0. At the next step, the network is fed both input $x_1$as well as the output h_0, generating new output h_1. The actual model of the case study uses Gated Recurrent Units (Cho et al., 2014) as its RNN component. This architecture was found to perform well on relatively small datasets (Chung et al., 2014), which is a key requirement for the training while drilling approach, where dataset gradually grows from empty while the well is drilled.

Another important aspect of developing machine learning model is how the training and testing datasets are created. This is especially important in work related to drilling, where logs are data-series. Most common way of creating a train/test split is random sampling, where a percentage of a dataset is randomly selected to be a part of a training or testing. This is a method that cannot be used for predictive models in drilling, since spurious correlations will inflate the testing result. Correct approach is to split the data into continuous sections, where first *n*% of a well is used as training, and remainder is used as testing. This is the most common way of performing a data split in research related to drilling.

A relatively new approach is continuous learning (Liu, 2017), where training dataset is continuously growing, and predictions are done based on training on all previous data. This approach fits field deployment particularly well, because it is equivalent to how data is collected while drilling. In this approach initial results are poor due to small size of the dataset, but the assumption is, that while the dataset expands the model will outperform models created on data from offset wells, as it better represents drilling currently at hand. Fig. 2 is meant to visually explain the data split strategies discussed above.

2. Case study and model design

The case study data from the open Volve dataset (Equinor, 2018; Tunkiel et al., 2020b) was used, specifically the well F9A. It was chosen as it contained a relatively long section of the well without any data issues in its depth-based log. It contains a curved section drilled with a bent sub motor, where inclination rises and falls in waves, as is characteristic of this method, see Fig. 3 for reference. The sensor lag is introduced artificially in the data and is equal to 23 m, a value that is in range of a typical BHA configration. This was necessary as the log in question contained already depth-corrected data, an operation that is performed after the well is drilled, hence a reverse operation was needed for a case study. What the model predicts is the continuous inclination data between the sensor and the bit location of each sample. Real-time attributes are the input to the model, including Rate of Penetration, and Weight on Bit from all the locations behind the bit, hence overlapping with the continuous inclination prediction. Inclination from the locations behind the sensor is used as in input to the RNN portion of the network. This is explained in detail in further sections.

2.1. Data preparation

Raw data from the real-time drilling logs are rarely useable as-is. A

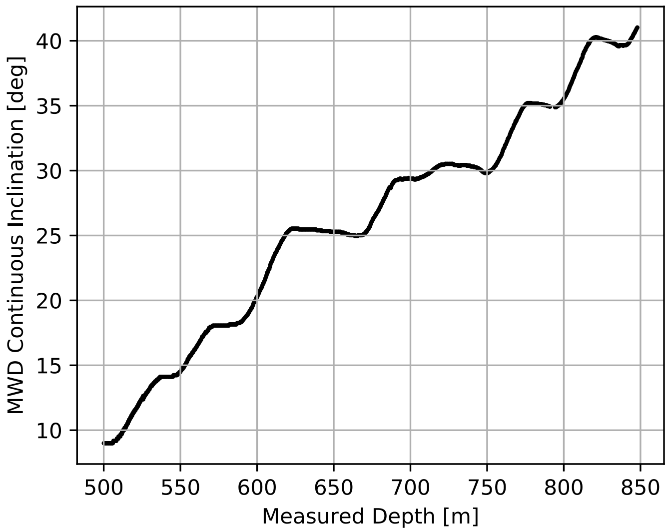


Fig. 3. Case study well inclination profile.

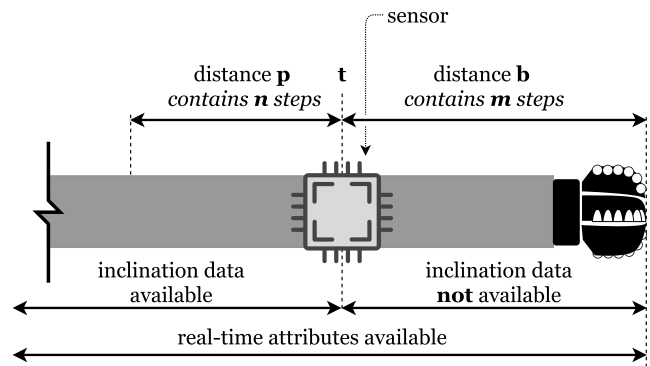


Fig. 4. Senor lag.

number of processes were applied to increase their quality. First, a section of the well without any missing data was identified, in this case, between 500 and 848 m measured depth. Since our approach relies on neighbouring data in the model as an input, depth-steps in the data series had to be made even. RadiusNeighbourRegressorm part of scikit-learn (Pedregosa et al., 2011) was used to re-sample the data at even depth intervals of 0.230876 m - median distance between datapoints in the original dataset. Attributes that have missing data after resampling process are considered not complete enough and disregarded. If a section of the well is missing some attributes, it will get discarded from future predictions. Alternatively, one can develop a system where such section of the well may be ignored completely in the process to retain certain attributes in the model when they come back on-line.

To include the past values information a windowing process was applied. Referring to Fig. 4, a single input sample contains inclination data from behind the sensor (already measured inclination values), as well as real-time attributes from behind and ahead of the sensor. In the presented case study, the distance between the sensor and the bit is 23 m, divided into 100 discrete measurements. Distance behind the sensor taken as an input the model is also 23 m, divided into 100 discrete measurements. The output of the model is inclination values between the sensor and the bit, also 23 m and 100 discrete values. Referring back to Fig. 4, distances p and b are equal to 23 m; number of discrete steps, both n and m is equal to 100. The distance between steps is even and approximately 0.23 m. This setup creates a model with a high number of inputs and outputs. Each included real-time attribute adds 200 inputs, since there are 100 values before and 100 values after the sensor. There are also 100 inputs related to inclination values. Presented case study has 51 useable real-time attributes. These are however reduced to 3 attributes through principle component analysis (PCA), described in further subsection, resulting in practice in $3x(100 + 100) + \ 100 = 700$inputs to the machine learning algorithm itself.

2.2. PCA transformation

In relation to input attributes, to simplify selection process, and easy field deployment, all instantaneously available attributes are used. Inclination data is stored separately, while all other data is compressed using Principle Component Analysis (Pearson, 1901), a dimensionality reduction method. Note that this reduces dimensions that the machine learning algorithm is exposed to only, as the input to the complete setup still takes all attributes. Resampled data is first normalized to a range (0, 1), fed through a PCA algorithm that reduces it to the prescribed amount of components, and normalized again to a range of (0,1). The number of output attributes and how it affects the prediction was evaluated and it was found that a reduction to 3 components from initial 51 attributes[1] generates best results (Mean Square Error (MSE) = 0.035) in terms of prediction error. The study for determining optimal number of PCA components was performed through complete training-while-drilling exercise, from 15% to 80% of available data, with 1% increments - process explained in detail further in the paper. Results were on average better than selecting all the attributes without PCA dimensionality reduction (MSE = 0.041). PCA-based results were also better than manual selection of attributes based on engineering judgement - approach applied to a related case study before (Tunkiel et al., 2020a) (MSE = 0.048), where average surface torque, average rotary speed, and rate of penetration were selected as inputs. Data from PCA dimension evaluation results are shown in Fig. 5, where mean square training error is plotted against the number of PCA components used, plus the reference values. The best solution, with 3 components, explains 88% of the total variance. It is worth noting that standardization of data was not performed. This process of substracting mean from sample values was tried through using RobustScaler, a solution from the Sklearn package (Pedregosa et al., 2011), and it produced overall inferior results.

The reason why dimensionality reduction decreases the error of a model is most likely tied to overfitting and spurious correlations. As explained earlier, inclusion of each real-time attribute in our case study increases number of inputs by 200. This results in 10 000 inputs if 50 attributes were to be used. Such high number of inputs in a dataset as (relatively) small as ours is bound to cause overfitting to some extent.

It must be noted that no prior attribute selection was performed. No correlation matrices were calculated nor any other approach was applied. This is connected to the expected deployment of the method, were decision related to which attributes will be available during drilling operation is not always known much in advance. Attribute selection is not trivial, and methods, such as mentioned correlation analysis are difficult to implement to work automatically; furthermore, the basic correlation methods will uncover only linear relationships. Therefore using all the available parameters through the PCA transformation is proposed as a solution that can be done fully automatically without manual intervention.

2.2.1. Nominal and incremental inclination data

Preparation of inclination data was different than for other parameters. It is not immediately obvious if best results will be achieved while predicting inclination data itself, or change in inclination (incremental value, first derivative), therefore both approaches were evaluated in parallel. Use of inclination change is simpler, as it can be used directly with (0,1) normalization. Use of actual inclination data is more difficult,

[1] These are attributes such as Weight on Bit kkgf, Average Standpipe Pressure kPa, Average Surface Torque kN.m, Rate of Penetration m/h, etc..

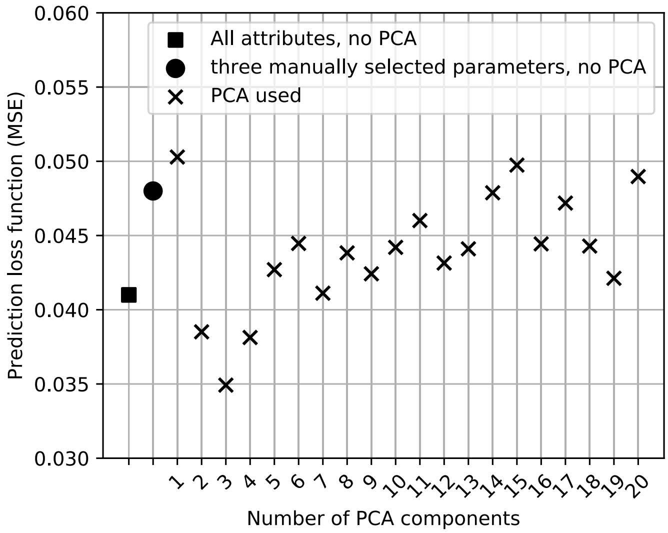


Fig. 5. PCA dimension evaluation.

as it requires normalization through introduction of a local coordinate system. Proposed neural network uses RNN layer to process previous values of the predicted attribute; in our case n=*100* input steps were selected, a value selected through hyperparameter tuning, which is explained in detail later in this paper. Nominal inclination data have to be scaled such, that first and oldest inclination input value is zero in the local coordinate system, and the highest value is no bigger than one. The length of the dataset after complete preparation is 1486 samples, a value that is a function of well depth data at hand, resampling rate, and the n and m values of the model described in the previous section.

2.3. Model design

2.3.1. Overall architecture

The model consists of two branches, RNN branch and Multi-Layer Perceptron (MLP) (Jain et al., 1996) branch; these branches respectively contain additional Gaussian noise and dropout layers. They are later concatenated and connected into a single Dense layer. See Fig. 6 for reference, as well as the publication first discussing this general model (Tunkiel et al., 2020a). All the layers used are from Keras library (Chollet and others, 2015) and therefore specific details can be found in the project's documentation. Inclination data is fed into the RNN branch, and all real-time attributes are fed into the MLP branch. The model was implemented in TensorFlow 2.1.0 with Keras library. Full source code used for this paper's case study is available on Github.[2]

As indicated in the data preparation section of this paper, it is not immediately clear if in the presented case study one should predict nominal inclination or the change in inclination, later referred as *incremental* method, as opposed to *nominal* method. Neither of the methods predict the actual inclination, since individual samples are encapsulated in a local coordinate system and scaled in range (0,1). The difference is that the incremental method works on the first derivative along the depth of the inclination. In practice, the *nominal* method predicts the value of the inclination in the local coordinate system, while the *incremental* method predicts the change in inclination value in the same local coordinate system. There are pro's and con's to each of the methods, which are highlighted in the results section of this paper.

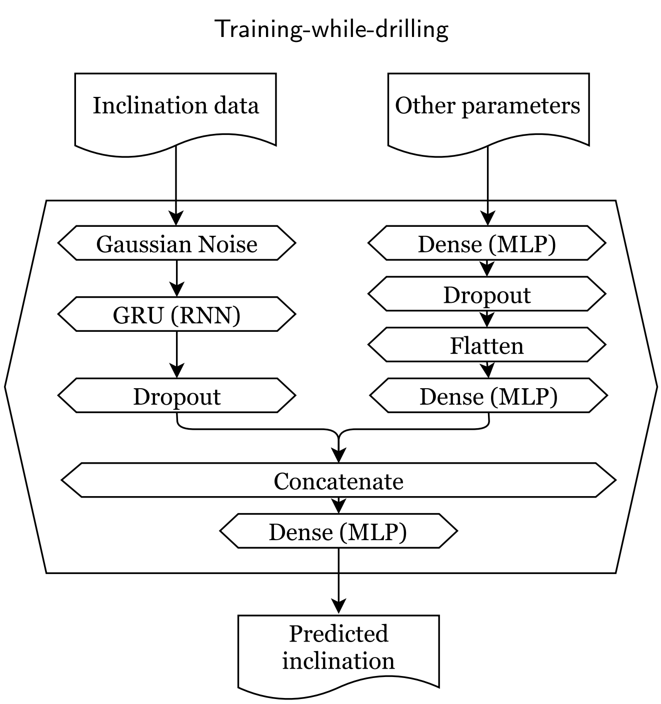


Fig. 6. Neural network architecture.

2.4. Continuous learning implementation

Continuous learning implementation flowchart is displayed in Fig. 7. Evaluation process starts at 15 percent of available dataset, or 52 m of drilling data. This minimum value was selected as the data from the drilled section has to be split further into training and validation. 15 percent of our dataset contains only 195 samples which are further split into *195*80%* samples for training and *195*20%* for validation. This is already a small dataset and it was decided not to train data on even

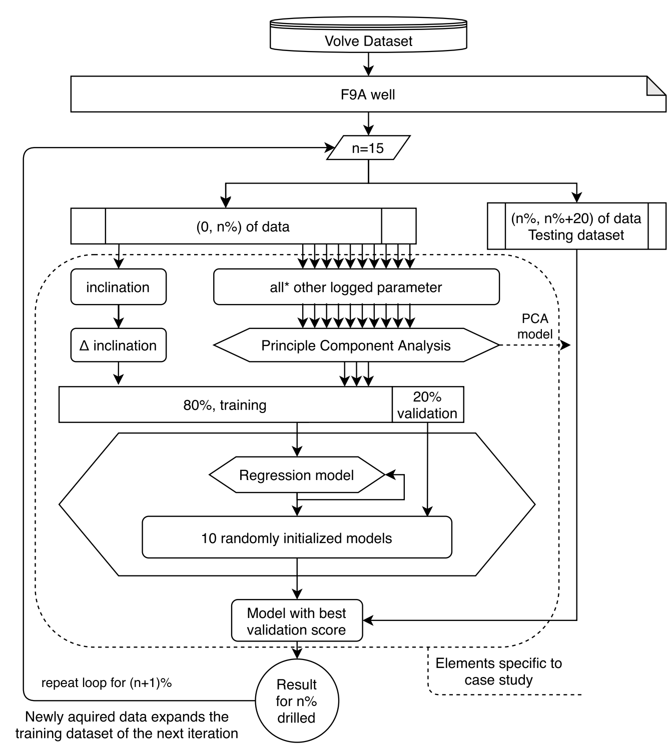


Fig. 7. Training while drilling process workflow.

[2] https://github.com/AndrzejTunkiel.

smaller sample size, hence starting point of 15% was selected. Training and validation subsets are continuous and the validation data boarders with the recent end of the data; alternative strategies were tested for locating the validation data, and the best results were achieved when it was placed at the end. This split is necessary to implement early stopping, another method crucial for avoiding overfitting. The validation data are not used in the backpropagation part of the training process itself, but they are continuously evaluated while the model is trained. Typically validation error drops together with training error along the training epochs, but at the point where overfitting begins, it starts to increase. This is the point where training is stopped and the model with best validation score is retained. Data consisting of future 20 percent of the dataset is set aside for testing of the model from current iteration. 20% is relatively big, and it was chosen to be indicative of a wider model performance. It is also important to mention that the PCA dimensionality reduction model is fit only on the available data within an iteration, and not on the testing data, as it is considered not available at the time of training. In other words, the PCA transformation rules (calculating the data covariance) are established only on the part of the dataset that is considered known. Subsequent transformation is done on the dataset that contains the testing data. The inclination values are not a part of PCA transformation. The PCA model is later used for model evaluation, as the input data have to be processed with the same PCA model that was used for training.

Training process is repeated ten times to increase accuracy with two competing strategies evaluated: a lottery ticket approach (Frankle and Carbin, 2018), where the model with best validation score is later used for testing, and an average of all ten models - results from both approaches are elaborated on in the results section. Next, the percentage of the well assumed to be drilled is increased by one percentage point and the complete training process is repeated. Increments can in practice be either shorter or longer. New models can be trained continuously and there is no underlying reason to artificially increase the intervals.

Our implementation uses TensorFlow 2.1.0 with integrated Keras library and Python 3.7. Model training was performed on Intel Core i7-8850H CPU, 32 GB of RAM and NVIDIA Quadro P2000 GPU with 5 GB of GDDR5 memory providing Peak Single Precision FP32 Performance at 3 TFLOPS. Model training required 2–15 min (2–15 m of drilled well at ROP of 60 m/h), depending on the simulated percentage size of the well drilled. Predictions based on the trained model are for all intents and purposes calculated instantaneously.

2.5. Hyperparameter tuning

Hyperparameter tuning is a process of adjusting various settings in the machine learning algorithm to increase its performance and is done utilizing training and validation dataset. This poses a problem as our proposed method assumes no prior access to data. Performing hyperparameter tuning on similar dataset and with the same goals can be done to overcome such issue. Such approach is utilized in other areas of machine learning, for instance a neural network detecting cats and dogs will not call for new hyperparameter tuning when detection classes are expanded to birds and rabbits since the problem at hand is technically identical from the perspective of the neural network. This is not to be confused with requirement of training a model on a similar well. This process is much more generic and likely not sensitive to geology or equipment used. Hyperparameters found to be working well for our case study are likely to provide good results when reused in model application to any bent sub directional drilling around the world. We were regrettably unable to evaluate and confirm this assumption due to lack of access to suitable dataset.

In our case study due to available data being limited to one well, hyperparameter tuning was performed on the same well that was later used for method evaluation. This was limited to layer size, dropout size, learning rate, kernel initialization variants, Gaussian noise levels and batch size, hence should not artificially increase the performance of evaluated case study. Hyperparameters stay constant throughout the complete drilling operation and all the iterations of the model generated as new data becomes available.

Tuning of these parameters was done using Bayesian optimization algorithm (Nogueira, 0000). Best parameters vary between nominal inclination and change in inclination approaches, with dropout layer at 50 percent, Gaussian noise layer at standard deviation equivalent to 0.2 percent of full scale, and approximately 350 neurons in the RNN layer and 10 and 100 neurons in final dense layer, depending on the prediction approach. Specific values can be found in the source code provided. All tuning was done with early stopping, with patience at 25 epochs and saving only the best model.

Three datapoints were selected, with 30, 55 and 80 percent of dataset used in the case study for training and validation as a basis for hyperparameter tuning exercise. Average loss of these three points was used when evaluating changing performance. Alternative methods are possible, such as evaluation based on the worst score, or evaluation based only on most difficult sections, i.e. those with little data. Method selection should be driven by specific objectives of the network under development. In our case study average overall performance was chosen as the key factor and method selected accordingly. Only three percentage points were selected to limit the time required for hyperparameter tuning, which is notorious for being time consuming. Note that PCA dimensionality reduction was not a part of final hyperparameter tuning. It was decided that this is a critical aspect of the model and therefore analysis of component quantity from 1 to 20 was performed separately, as shown before in Fig. 5.

In the future, as computational power increases, hyperparameter tuning prior to model deployment may not be necessary. As it is required to evaluate hundreds of alternative hyperparameter configurations in the tuning process, even models that are trained in mere minutes take hours to become optimized. This time has to be significantly reduced, by two orders of magnitude, to perform it during the drilling operation itself. Considering current progress in the discipline this is unlikely to happen in the next 10 years, unless new, more efficient algorithms are discovered.

2.6. Overfitting

Proposed method was optimized for small datasets to provide useful results as fast as possible. Small datasets are often prone to overfitting, where a machine learning algorithm *memorizes* specific datapoints instead of creating a method capable of generalizing. A number of methods were applied to tackle this problem. Typical approaches to overfitting are a dropout layer, where neurons are randomly dropped while training, Gaussian noise layer, where artificial noise is added to the signal and an architecture minimizing the number of neurons. Another approach to overfitting reduction is an ensemble of models, which is explained in detail in the results section of this paper.

3. Results

Results from a single sample can be visualized by plotting the past inclination data, predicted inclination data and ground-truth target values. The same method is used regardless of using nominal inclination data or incremental inclination data. This gives a good representation of the task at hand in terms of practical results that can be achieved. One sample of such chart is shown in Fig. 8. Note how the inclination prediction follows the same pattern and values relatively close to the actual data. The rotating potion of the bent sub drilling, where inclination is temporarily constant is also well represented. Note that the complete cycle of build-hold-build takes approximately 20 m in our case study, and prediction window used is 23 m. The y-axis refers to a local coordinate system of a sample, where first, oldest inclination datapoint is moved to zero.

There are multiple ways of describing the error between the

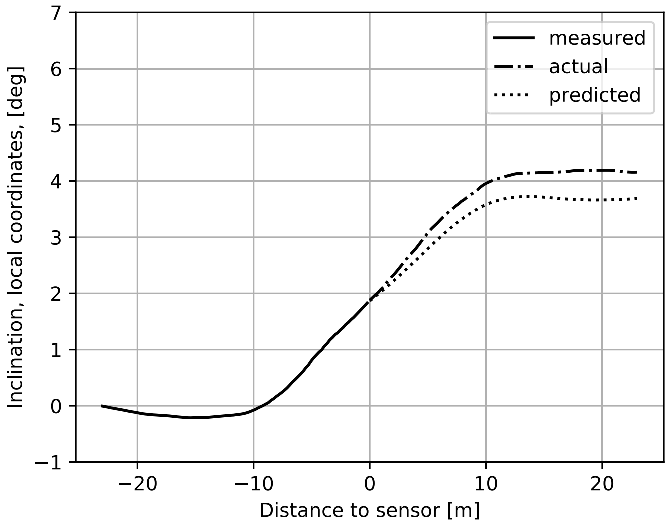
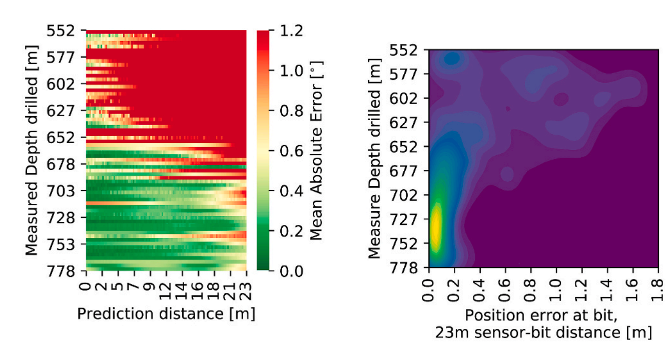


Fig. 8. Sample result, 60% of the dataset for training and validation, Measured Depth ca. 700 m, incremental inclination model.

prediction and the ground truth. It must be first established what is actually the value of interest that is being predicted, as this may change from application to application. Fig. 9a shows the mean absolute error that this paper uses to evaluate the presented model in its variations. This specific figure is a heatmap showing error for the prediction with the prediction distance on the horizontal axis and a given distance drilled on vertical axis. This figure immediately shows that the further from sensor one predicts, the worse the results; it also shows that predictions made early in the well are worse than the later predictions. This is in line with expectations, as at measured depth of 552 m (after 52 m of available data, which starts at MD = 500 m) the training dataset is too small to reliably predict future data yet.

Another potential way of understanding data is in terms of the positional error. Fig. 9b shows how different is the predicted bit position from the true position, considering a local coordinate system. This is calculated in the inclination plane using predicted inclination value and the fixed step of the prediction of 0.23 m. Here, the horizontal axis shows position error, and the vertical axis measured depth drilled, making a two dimensional histogram. The position error drops all the way to and below 0.1 m in the latter section of the dataset. Alternatively, positional confidence intervals can be plotted, as shown in Fig. 10, displaying roughly the same information, but in a more quantifiable manner. It shows both the median line as well as the confidence intervals, between 5th and 50th, and between 50th and 95th percentile.

Error can also be quantified in terms of R^2 value. This was calculated both based on predicted angle as well as predicted local x and y



(a) Mean Absolute Error, degrees, as a function of prediction distance.

(b) Positional error at bit, distribution heatmap.

Fig. 9. Different prediction error metrics in a function of well depth drilled, nominal method.

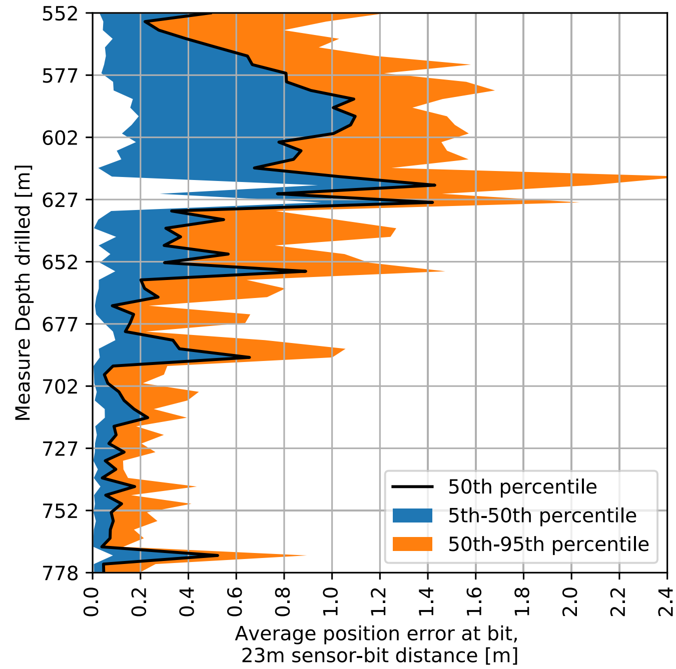


Fig. 10. Confidence intervals for bit position prediction, nominal method.

coordinates. While results for all those 3 parameters follow similar trend, they are slightly different as seen in Fig. 11. The beginning of the well shows lack of meaningful prediction, i.e. no correlation with R^2 values below zero, which significantly improves later in the well. Note that R^2 values below zero were changed to zero for readability. The calculations were done for angle prediction at all prediction steps, while the error for coordinates were done only for the bit position, since all predictions are used to calculate the final position anyway. The prediction based on coordinates shows better results, which is likely due to averaging out of the noise in predicted data.

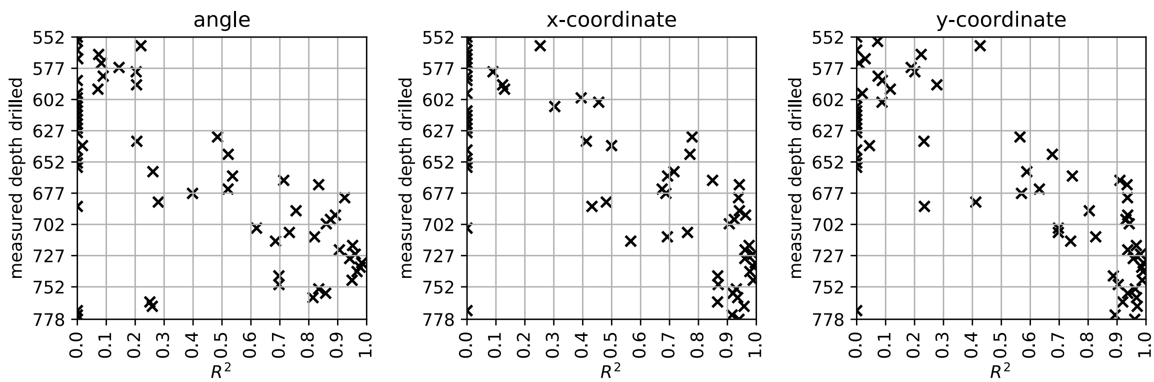
From multiple evaluation metrics we have chosen to use mean absolute error to discuss and elaborate on alternative prediction methods, as the most intuitively understood value. As the data used for case study is shared, as well as complete source code, it is possible to calculate additional metrics on demand with relative ease. It must also be stressed that presented results, in terms of accuracy, have no equivalent in current state of the art. Presented method generates values that otherwise would simply not be there; normally the inclination value between the sensor and the bit is considered simply as unknown.

3.1. Alternative architectures benchmark

To highlight the benefits of including past values' information into the network two more traditional architectures were tested - MLP and extreme gradient boosting (XGBoost) (Chen and Guestrin, 2016),[3] a method that won multiple machine learning competitions. These algorithms were applied using a single datapoint per sample, without taking into account the past values. That is, the only inputs are from data strictly co-located in time with the output. Results are shown in Fig. 12. Note that in this approach the inclination change was predicted, and actual inclination was reconstructed through cumulative sum to calculate the mean absolute error relative to the actual inclination value.

While both the MLP method and XGB method show mean absolute error lower than the nominal method, this is mostly due to very high error rates in the early stages of drilling. Both of the simpler methods predict inclination well throughout the well only for short prediction

[3] Version xgboost-1.1.0 was used, Python implementation.



(a) R^2 values for angle predictions, all values. (b) R^2 values for bit position prediction, x component. (c) R^2 values for bit position prediction, y component.

Fig. 11. R^2scores for different predicted values.

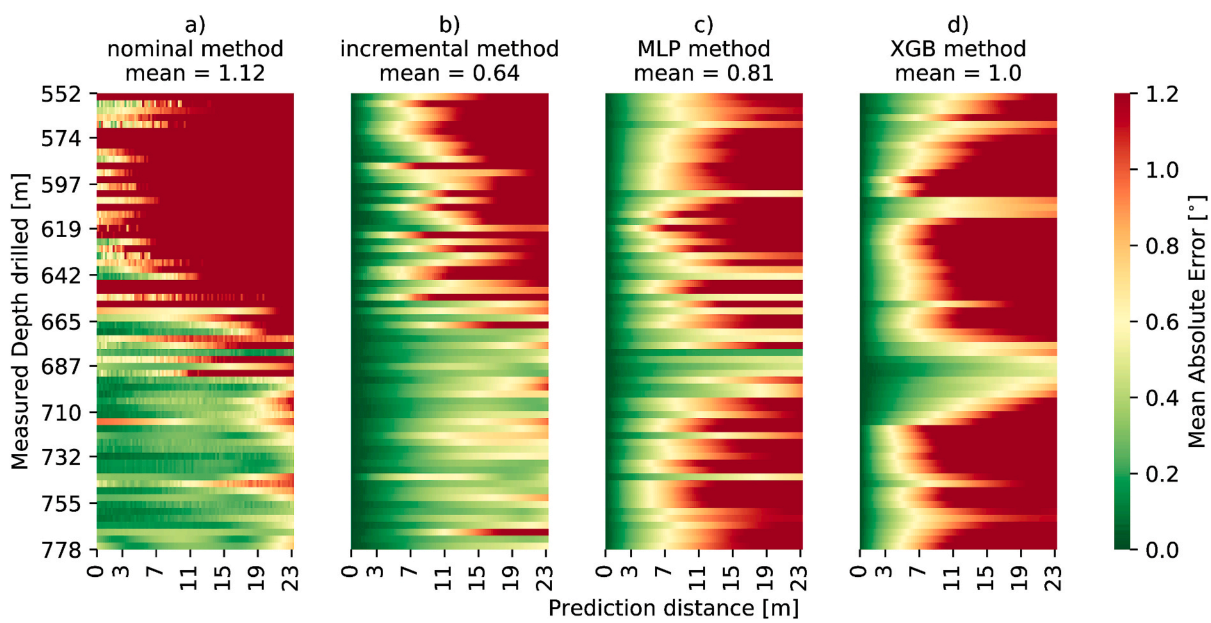


Fig. 12. Comparison between traditional and proposed approach.

distance. At approximately 600 m of measured depth drilled there is an area of low error visible, but this does not continue further into the well, suggesting area that is simply easier to predict.

Direct Comparison between XGBoost, the better performing of two simple methods, and our proposed method (incremental inclination, lottery ticket approach) was also performed, as seen in Fig. 13, where a difference between mean absolute error values are seen. What is interesting here is that the simpler model worked much better on a small dataset, although referring back to Fig. 12, error for predictions above few meters was nevertheless high. This behaviour most likely stems from the fact, that our proposed method uses a much more elaborate structure capable of finding more complex relationships. This in turn penalizes problems with small training datasets; only after collecting sufficiently large amount of data our proposed method outperforms simpler models. XGBoost, which is an ensemble of basic algorithms, perform better on a small dataset because it requires less data to train efficiently. Recurrent neural networks require bigger training datasets to perform well, which is evident in this comparison.

3.2. Incremental compared to nominal model

Although the incremental model shows significantly lower MAE

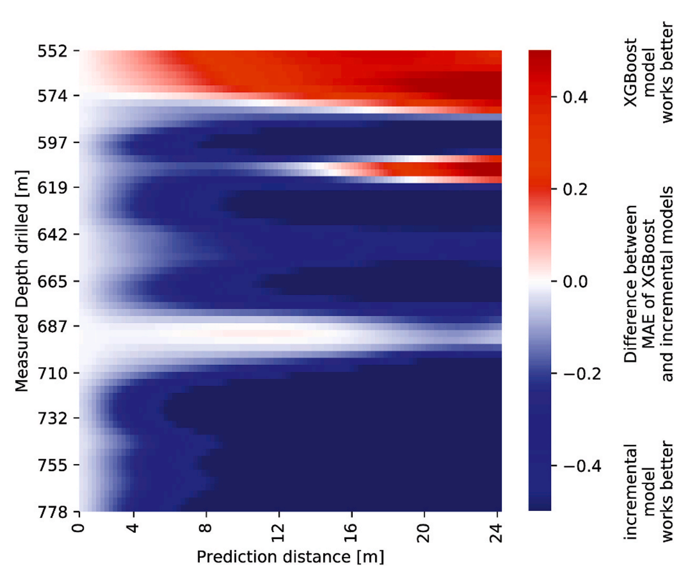


Fig. 13. Comparison between proposed approach and XGBoost.

scores it does not necessarily mean that it is better in all scenarios. As seen earlier in Fig. 13, nominal model performs better for further predictions. This is explored in detail in Fig. 14, where difference between the mean absolute error of two different models is calculated. Slight Gaussian blur was added to the results for easier analysis and outlier removal. With prediction distance on x-axis and drilled depth on y-axis, the color suggests which model is more accurate.

The results suggest, that if a further prediction is targeted, the nominal inclination approach is better. If however short prediction is needed, up to approximately 8 m, it is the incremental inclination method that works best. Referring back to Fig. 16, it is clear that inclination method delivers good prediction for the first few meters from the start and provides good 10 m + predictions earlier then the alternative. Depending on requirements, one or the other approach should be selected, or potentially an ensemble of those two methods can be implemented. The difference in results comes from the target predicted value - nominal inclination or inclination change. When predicting the nominal value, the algorithm may perform poorly in the short term prediction, however for further data-points the algorithm still *aims* at the actual inclination value, while the incremental model will accumulate the errors from each consecutive prediction, as all those values are needed to recreate further inclination values.

3.3. Lottery ticket and ensemble results

While developing the method it became clear that the results often vary between good and bad, even if calculated for identical data, only with different random seed. This is because training process has stochastic elements, such as weights and biases initialization. Distribution of MAE values for repeated runs is presented in Fig. 15 for both the method predicting nominal inclination as well as predicting inclination change. Standard deviation at approximately 5 percent of mean MAE suggests that it is possible to improve the accuracy. This is in line with recent research discussing *lottery ticket hypothesis*, that network initialization may be simply lucky and achieve better performance (Frankle and Carbin, 2018). Alternatively, average of models, otherwise known as ensemble, is often used to increase the prediction performance, which is especially common in climate research (Goerss, 2000; Najafi and Moradkhani, 2015). Both approaches were evaluated by training the model 10 separate times, and in one scenario selecting the model with best validation score, and in the other taking the average prediction of all the models. Repeating model training ten times was chosen as a balance between increased performance and increased model training

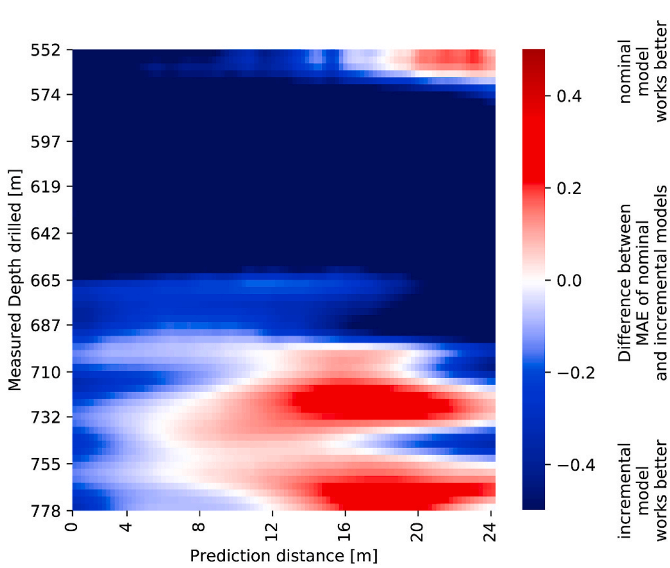


Fig. 14. Comparison between approaches.

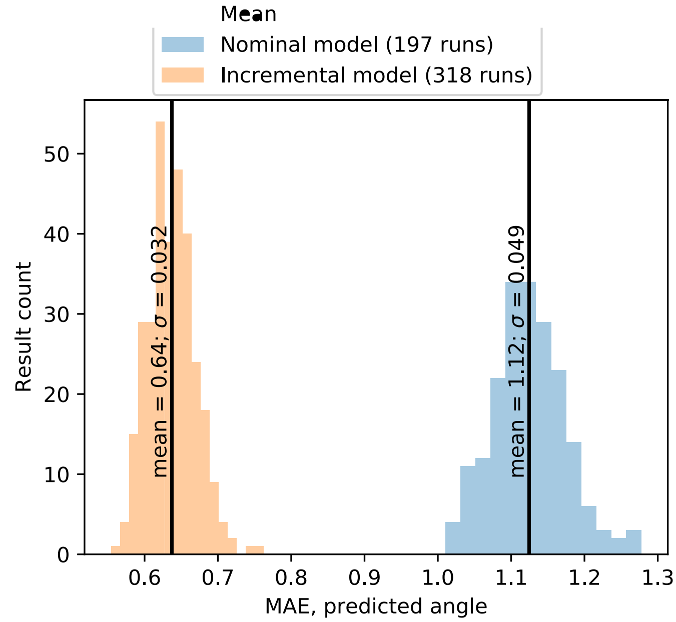


Fig. 15. Histograms of incremental and nominal method MAE, multiple runs compared.

time; increasing this value would continue to yield continuously smaller improvements. Additionally, predicting inclination variant and predicting of inclination change was tested, resulting in total of four different models. Results are shown in Fig. 16. When using the nominal inclination model, the mean MAE dropped from 1.12 to 1.07, approximately 5% improvement to the lottery ticket method. The inclination change method also showed 5% improvement for the lottery ticket method, with the ensemble method actually increasing MAE, although the standard distribution was significantly reduced. This suggests that the lottery ticket brings tangible, modest improvements, and should be used. There were approximately 40 simulations run for each ensemble and lottery ticket variant to find the distribution of the results.

The color refers to Mean Absolute Error; the scale was set such that yellow color is equivalent to 0.6 degree of average absolute error, a value tentatively deemed acceptable. Note that over the course of predicted 23 m inclination can change value between minus 0.4° and plus 5°. What is worth highlighting is that some red areas are off the scale, above the 1.2° error visible as the deep red color. All predictions with error above that threshold were considered useless.

4. Discussion, usability threshold

In relation to our case study, it was possible to achieve results with mean absolute error under 0.5° for prediction horizon of 23 m after approximately 180 m of drilled well. This is a relatively short section of a well and an acceptable *start-up* period necessary to build a sufficient training dataset after which reasonable predictions can be made. Considering the practical use of the case study this can be considered acceptable when the target inclination is further ahead, which it typically is. In relation to other applications, such as ROP prediction, this method should be applicable as well. While ROP lacks clear sequences as in bent sub drilling, information from data directly adjacent to the prediction area undoubtedly carries useful information.

There are some caveats when this technology is deployed in the field. Considerations have to be made when changes are introduced to the bottom hole assembly. These can be minor, such as replacing a blunt bit, or major, such as change in bent sub angle. A decision has to be made whether the data from before the change should continue to be used, or if a new training while drilling model should be trained from scratch. The best solution is likely to execute both approaches simultaneously

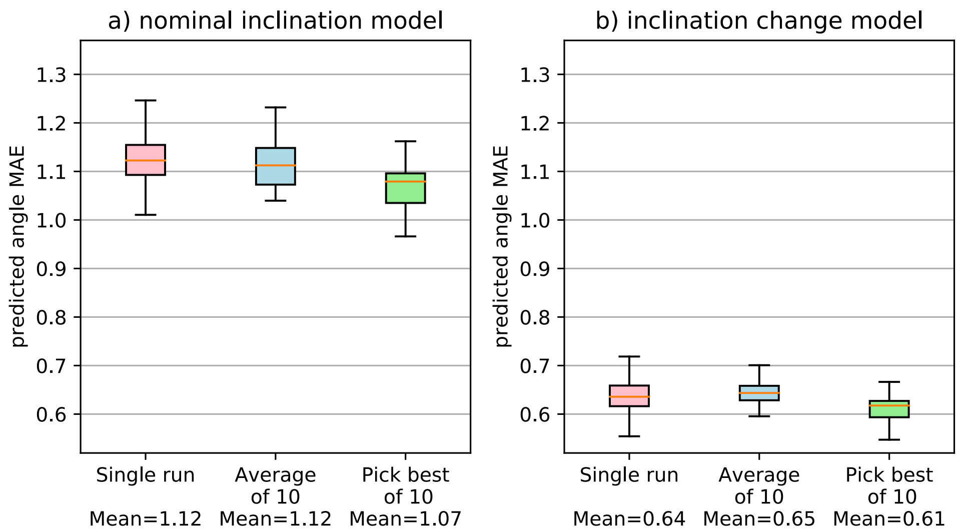


Fig. 16. Lottery ticket and ensemble performance evaluation and b) showing nominal inclination method, and subplot c) and d) the incremental inclination method. Lottery ticket approach is shown in subplots a) and c), and ensemble in subplots b) and d). The x-axis is the prediction distance, here 23 m, and y-axis refers to measured depth drilled.

and monitor performance.

Referring to all the results presented in this paper, it is worth highlighting that the case study predicts the data gap that is 23 m long. While various results suggest that certain models work better for a shorter prediction horizon, it still has to be considered in relation to a 23 m long prediction model. When tasked with shorter prediction, for example in a case where a sensor is 10 m behind the bit, results may be different.

Improvements can be achieved by utilizing logging while drilling (LWD) data, as it contains formation related information. While that information is not immediately available, and cannot be considered a real-time attribute, it certainly has potential for improved performance, as well as more efficient model training which can lead to acceptable results earlier in the drilling process.

4.1. Selecting the best model

Multiple alternative models were discussed in this paper, namely two including dynamic behaviour, nominal and incremental, and two standard regression models, MLP and XGBoost. All of these approaches seem to have strong and weak sides related to how much training is necessary and how far the prediction could be done with acceptable results. To indicate which one performs best in which area, a figure was created identifying the best out of four models.

For each point relating to specific distance drilled and prediction distance the best performing model was selected and plotted with an individual color. Additionally, areas with Mean Absolute Error above 0.6° were truncated indicating that none of the explored methods worked sufficiently well. Results are shown in Fig. 17. Note that the marker size decreases with the rising error, giving an additional visual clue about the performance. The area of the chart is overwhelmingly occupied by both proposed models with dynamic behaviour, with simpler alternatives occupying very small portions of it, especially early in the well. This again confirms previously stated conclusions, that the simpler models learn faster, but as the training set expands, the more complex ones prevail.

5. Conclusion

Presented method tailored for continuous learning shows good performance in the case study of predicting sensor data during directional drilling with bent motor. With existing methods being able to predict only nearest 7 m while keeping the mean absolute error under 0.6°, our proposed method achieve that goal for 23 m of prediction most of the time. With multiple inputs decomposed to only three via PCA method, the model can be applied with little analysis in terms of available attributes, significantly reducing the workflow related to hyperparameter tuning.

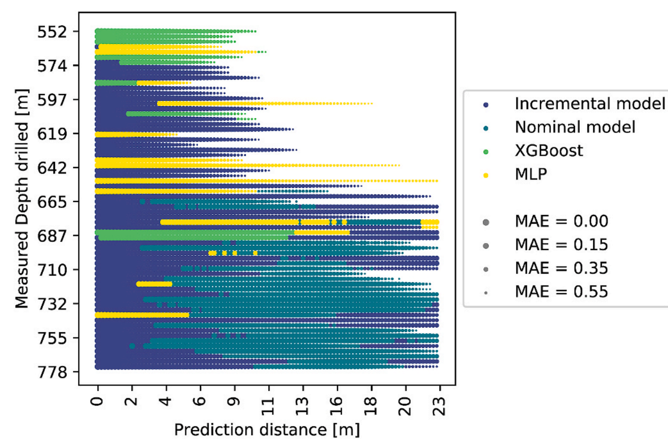


Fig. 17. Mapping best models.

Further work is needed to verify the method's applicability to predicting sensor readings of other attributes, such as gamma ray, neutron measurement, and others; and to fully quantify its potential in drilling. Presented method may also find applications in non-petroleum areas such as weather forecasting and motion capture technologies, creating models through continuous learning filling in data for failed sensors, obscured markers, and data delayed for other reasons.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to express gratitude to Equinor for providing funding for this research through Equinor Akademia Program.

References

Ahmed, A., Ali, A., Elkatatny, S., Abdulraheem, A., 2019a. New artificial neural networks model for predicting rate of penetration in deep shale formation. Sustainability 11, 6527. https://doi.org/10.3390/su11226527.

Ahmed, O.S., Adeniran, A.A., Samsuri, A., 2019b. Computational intelligence based prediction of drilling rate of penetration: a comparative study. J. Petrol. Sci. Eng. 172, 1–12. https://doi.org/10.1016/j.petrol.2018.09.027.

Amar, K., Ibrahim, A., 2012. Rate of penetration prediction and optimization using advances in artificial neural networks, a comparative study. In: IJCCI 2012 - Proceedings of the 4th International Joint Conference on Computational Intelligence, pp. 647–652. https://doi.org/10.5220/0004172506470652.

Anifowose, F., Khoukhi, A., Abdulraheem, A., 2011. Impact of training-testing stratification percentage on artificial intelligence techniques: a case study of porosity and permeability prediction. 5th Global Conference on Power Control and Optimization.

Anifowose, F., Khoukhi, A., Abdulraheem, A., 2017. Investigating the effect of training–testing data stratification on the performance of soft computing techniques: an experimental study. J. Exp. Theor. Artif. Intell. 29, 517–535.

Chen, T., Guestrin, C., 2016. XGBoost: a scalable tree boosting system. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, pp. 785–794. https://doi.org/10.1145/2939672.2939785.

Chhantyal, K., Hoang, M., Viumdal, H., Mylvaganam, S., 2018. Flow rate estimation using dynamic artificial neural networks with ultrasonic level measurements. In: Proceedings of the 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, the 57th SIMS Conference on Simulation and Modelling SIMS 2016. Linköping University Electronic Press, pp. 561–567.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing. Proceedings of the Conference 1724–1734. https://doi.org/10.3115/v1/d14-1179.

Chollet, F., others, 2015. Keras. \url{https://keras.io}.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. http://arxiv.org/abs/1412.3555.

Equinor, 2018. Volve field data (CC BY-NC-SA 4.0). https://www.equinor.com/en/news/14jun2018-disclosing-volve-data.html.

Eren, T., Ozbayoglu, M.E., 2010. Real time optimization of drilling parameters during drilling operations. In: SPE Oil and Gas India Conference and Exhibition. Society of Petroleum Engineers. https://doi.org/10.2118/129126-MS.

Frankle, J., Carbin, M., 2018. The lottery ticket hypothesis: finding sparse, trainable neural networks.

Goerss, J.S., 2000. Tropical cyclone track forecasts using an ensemble of dynamical models. Mon. Weather Rev. 128, 1187–1193.

Han, J., Sun, Y., Zhang, S., 2019. A data driven approach of ROP prediction and drilling performance estimation. International Petroleum Technology Conference. https://doi.org/10.2523/iptc-19430-ms.

Hegde, C., Daigle, H., Millwater, H., Gray, K., 2017. Analysis of rate of penetration (ROP) prediction in drilling using physics-based and data-driven models. J. Petrol. Sci. Eng. 159, 295–306. https://doi.org/10.1016/j.petrol.2017.09.020.

Hegde, C., Gray, K., 2018. Evaluation of coupled machine learning models for drilling optimization. J. Nat. Gas Sci. Eng. 56, 397–407. https://doi.org/10.1016/j.jngse.2018.06.006.

Hegde, C., Gray, K.E., 2017. Use of machine learning and data analytics to increase drilling efficiency for nearby wells. J. Nat. Gas Sci. Eng. 40, 327–335. https://doi.org/10.1016/j.jngse.2017.02.019.

Hegde, C., Wallace, S., Gray, K., 2015. Using trees, bagging, and random forests to predict rate of penetration during drilling. In: Society of Petroleum Engineers - SPE Middle East Intelligent Oil and Gas Conference and Exhibition. Society of Petroleum Engineers. https://doi.org/10.2118/176792-MS.

Jain, A.K., Mao, J., Mohiuddin, K.M., 1996. Artificial neural networks: a tutorial. Computer 29, 31–44.

Jiang, W., Samuel, R., 2016. Optimization of rate of penetration in a convoluted drilling framework using ant colony optimization. In: SPE/IADC Drilling Conference. Proceedings, Society of Petroleum Engineers (SPE). https://doi.org/10.2118/178847-ms.

Koryabkin, V., Semenikhin, A., Baybolov, T., Gruzdev, A., Simonov, Y., Chebuniaev, I., Karpenko, M., Vasilyev, V., 2019. Advanced data-driven model for drilling bit position and direction determination during well deepening. https://doi.org/10.2118/196458-MS.

Liu, B., 2017. Lifelong machine learning: a paradigm for continuous learning. Front. Comput. Sci. 11, 359–361. https://doi.org/10.1007/s11704-016-6903-6.

Mantha, B., Samuel, R., 2016. ROP optimization using artificial intelligence techniques with statistical regression coupling. In: Proceedings - SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers (SPE). https://doi.org/10.2118/181382-ms.

Najafi, M.R., Moradkhani, H., 2015. Multi-model ensemble analysis of runoff extremes for climate change impact assessments. J. Hydrol. 525, 352–361.

Nogueira, F.. Open source constrained global optimization tool for {Python}. https://github.com/fmfn/BayesianOptimization.

Osarogiagbon, A., Muojeke, S., Venkatesan, R., Khan, F., Gillard, P., 2020. A New Methodology for Kick Detection during Petroleum Drilling Using Long Short-Term Memory Recurrent Neural Network. Process Safety and Environmental Protection.

Pearson, K., 1901. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2, 559–572. https://doi.org/10.1080/14786440109462720.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. Why should i trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536. https://doi.org/10.1038/323533a0.

Sabah, M., Talebkeikhah, M., Wood, D.A., Khosravanian, R., Anemangely, M., Younesi, A., 2019. A machine learning approach to predict drilling rate using petrophysical and mud logging data. Earth Sci. India 12, 319–339. https://doi.org/10.1007/s12145-019-00381-4.

Shi, X., Liu, G., Gong, X., Zhang, J., Wang, J., Zhang, H., 2016. An efficient approach for real-time prediction of rate of penetration in Offshore drilling. Math. Probl Eng. 2016, 3575380. https://doi.org/10.1155/2016/3575380.

Soares, C., Daigle, H., Gray, K., 2016. Evaluation of PDC bit ROP models and the effect of rock strength on model coefficients. J. Nat. Gas Sci. Eng. 34, 1225–1236. https://doi.org/10.1016/j.jngse.2016.08.012.

Soares, C., Gray, K., 2019a. Real-time predictive capabilities of analytical and machine learning rate of penetration (ROP) models. J. Petrol. Sci. Eng. 172, 934–959. https://doi.org/10.1016/j.petrol.2018.08.083.

Soares, C., Gray, K., 2019b. Real-time predictive capabilities of analytical and machine learning rate of penetration (ROP) models. J. Petrol. Sci. Eng. 172, 934–959. https://doi.org/10.1016/j.petrol.2018.08.083.

Tunkiel, A.T., Wiktorski, T., Sui, D., 2020a. Continuous drilling sensor data reconstruction and prediction via recurrent neural networks. In: Submitted to Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE.

Tunkiel, A.T., Wiktorski, T., Sui, D., 2020b. Drilling dataset exploration, processing and interpretation using Volve field data. Submitted to Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE.

Wang, G., Long, S., Ju, Y., Huang, C., Peng, Y., 2018. Application of horizontal wells in three-dimensional shale reservoir modeling: a case study of Longmaxiâ€"Wufeng shale in fuling gas field, Sichuan basin. AAPG (Am. Assoc. Pet. Geol.) Bull. 102, 2333–2354. https://doi.org/10.1306/05111817144.

Wang, H., 2017. Drilling trajectory prediction model for push-the-bit rotary steerable bottom hole assembly. Int. J. Eng. 30, 1800–1806.

Wang, M., Li, X., Wang, G., Huang, W., Fan, Y., Luo, W., Zhang, J., Zhang, J., Shi, X., 2020. Prediction model of build rate of push-the-bit rotary steerable system. Math. Probl Eng. 2020, 4673759. https://doi.org/10.1155/2020/4673759.

Yi, P., Kumar, A., Samuel, R., 2014. Real-time rate of penetration optimization using the shuffled frog leaping algorithm (SFLA). In: Society of Petroleum Engineers - SPE Intelligent Energy International 2014. Society of Petroleum Engineers (SPE), pp. 116–125. https://doi.org/10.2118/167824-ms.

Zou, C., 2017. Unconventional Petroleum Geology. Elsevier.