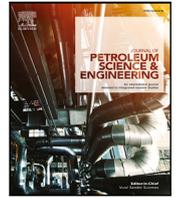




Contents lists available at ScienceDirect

Journal of Petroleum Science and Engineering

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

Data-driven sensitivity analysis of complex machine learning models: A case study of directional drilling

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:

Sensitivity analysis
Partial derivative
Directional drilling
Volve dataset

ABSTRACT

Classical sensitivity analysis of machine learning regression models is a topic sparse in literature. Most of data-driven models are complex black boxes with limited potential of extracting mathematical understanding of underlying model self-arranged through the training algorithm. Sensitivity analysis can uncover erratic behavior stemming from overfitting or insufficient size of the training dataset. It can also guide model evaluation and application. In this paper, our work on data-driven sensitivity analysis of complex machine learning models is presented. Rooted in one-at-a-time method it utilizes training, validation and testing datasets to cover the hyperspace of potential inputs. The method is highly scalable, it allows for sensitivity analysis of individual as well as groups of inputs. The method is not computationally expensive, scaling linearly both with the available data samples, and in relation to the quantity of inputs and outputs. Coupled with the fact that calculations are considered embarrassingly parallel, it makes the method attractive for big models. In the case study, a regression model to predict inclinations using recurrent neural network was employed to illustrate our proposed sensitivity analysis method and results.

1. Introduction

Sensitivity of a model describes the severity of change of the model's output related to the change of a given input value. It can give an insight in the influence of input variables on outputs. Such analysis is necessary for understanding models' behavior in terms of the change of input values, noise tolerance, data quality, internal structure, etc.

There are a number of statistical methods to evaluate sensitivity, for instance Sobol' indices (Sobol, 1993), cobweb plots (Kurowicka and Cooke, 2006), various metamodels (Simpson et al., 2001) and more (Iooss and Lemaître, 2015). These methods might however be not suitable for high-dimensional models of hundreds of input variables. In cases where inputs are highly dependent on each other, as in case of recurrent neural networks, sensitivity analysis remains an area of active research (Iooss and Lemaître, 2015). More recent work have introduced Shapley effect (Radaideh et al., 2019) to tackle this problem, however the high computational cost remains a problem.

Sensitivity information can be, to an extent, extracted by examining weights of neurons in the network. Such approach has been published in the past (Jamal et al., 2014), however it becomes impractical for

networks when neurons are counted in hundreds. Most commonly however, a simple sensitivity index (Loucks et al., 2005), a one-at-a-time (OAT) approach is utilized. In light of the rise in machine learning approaches applied to drilling, such as Ulker and Sorgun (2016) and Sorgun and Ulker (2016), a rise in focus on models' sensitivity analysis is expected in the near future.

This paper presents sensitivity analysis using partial derivatives (PaD) with the dataset used for development of the machine learning model as a basis of a quasi-Monte Carlo analysis (Caflich, 1998). To our knowledge this is the first comprehensive exploration of PaD method of sensitivity analysis for models with number of inputs over 100; case study of directional drilling model this paper presents consists of 643 inputs and 100 outputs.

In this paper, presented case study contains hundreds of inputs and outputs and is based on the model prepared for prediction and recovery of inclination data in directional drilling (Tunkiel et al., 2020). The aim of this study is to explore the sensitivity analysis for complex machine learning models, such as those exploiting recurrent neural networks, where it is impossible or impractical to follow the classical

* Corresponding author at: Department of Energy and Petroleum Engineering, Faculty of Science and Technology, University of Stavanger, 4036 Stavanger, Postboks, 8600 Forus, Norway.

E-mail address: andrzej.t.tunkiel@uis.no (A.T. Tunkiel).

URL: <https://www.ux.uis.no/~atunkiel/> (A.T. Tunkiel).

<https://doi.org/10.1016/j.petrol.2020.107630>

Received 29 May 2020; Received in revised form 6 July 2020; Accepted 8 July 2020

Available online 10 July 2020

0920-4105/© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

methods — this is done in detail in Section 3.2. Employing the dataset that was originally used for training, validation and testing of the model is the basis for our analysis. Basic method where an input is altered within carefully selected range and response of the output is evaluated was applied; this was in turn applied through all the samples of existing dataset as individual starting points, generating individual results. Using this method, sensitivity can be presented as a statistical distribution based on starting points representative of realistic potential inputs.

Similar work on a much smaller scale was performed by Lu et al. (2001) for neural networks with 19 inputs for spool fabrication productivity studies, which was later picked up for business research (Detienne et al., 2003), climatology (Nourani and Sayyah Fard, 2012), nanotechnology (Babakhani et al., 2017), petroleum (Dutta and Gupta, 2010), metallurgy (Shojaeefard et al., 2013) and ecology (Franceschini et al., 2019). These studies were however restricted to non-recurrent models and no more than 26 inputs in total. Existing analyses lack specific focus on the base dataset, which becomes necessary with the rising number of inputs, as the dataset necessarily will not populate the complete possible input hyperspace. This paper contributes to filling these gaps, applying and analyzing the method for big machine learning models. To our knowledge we are first in presenting, and formally defining a data-driven sensitivity analysis that is suitable for recurrent models with inputs and outputs defined as spatio-temporal sequences.

The paper is structured with focus of gradual understanding of the presented method. First, sensitivity calculations are performed using basic mathematical model as an example. Next, our novel method of data-driven approach to sensitivity is introduced. Using the same mathematical model as before, our method is applied and sample results and conclusions are presented. As a next stage, a mathematical model is replaced by a machine learning model mimicking the original one to evaluate how the results differ. This prepares the reader for analysis of our case study, a complex model for predicting continuous inclination data that utilizes 643 inputs and 100 outputs. In the case study, sensitivity analysis is performed for both individual inputs, as well as sets of inputs — a unique, new possibility in our novel method for sensitivity analysis. At the end, benchmarking against Sobol' indices is performed to evaluate how our proposed method compares with other established approaches.

To explore and better understand basic one-at-a-time method, assume that the system is described as:

$$R = f(S) \quad (1)$$

where R is the output and S is the input of a model described by a function f . Classical OAT sensitivity index is typically calculated at an arbitrarily selected point in the possible input domain, typically in the middle. Sensitivity index can be calculated through the following equation, that can be considered a partial derivative:

$$SI_{RS} = \frac{[R(S_0 + \Delta S) - R(S_0 - \Delta S)]}{2\Delta S} \quad (2)$$

where SI_{RS} denotes sensitivity index for an output variable R per unit change in the magnitude of an input parameter S from its base value S_0 . ΔS is the change applied to the input.

There are significant shortcomings to this basic approach. To illustrate we introduce a simple model of a flow rate through a nozzle:

$$Q = c \cdot A \cdot v = c \cdot \pi r^2 \cdot \sqrt{\frac{2\Delta p}{\rho}} \quad (3)$$

where:

- Q is Flow Rate through a nozzle, output in m^3/s .
- c is Discharge Coefficient (unitless), $c \in (0.5, 1)$
- A is Nozzle Area in mm^2
- v is Flow Velocity, in mm/s
- r is Nozzle Diameter, in mm , $r \in (1, 50)$

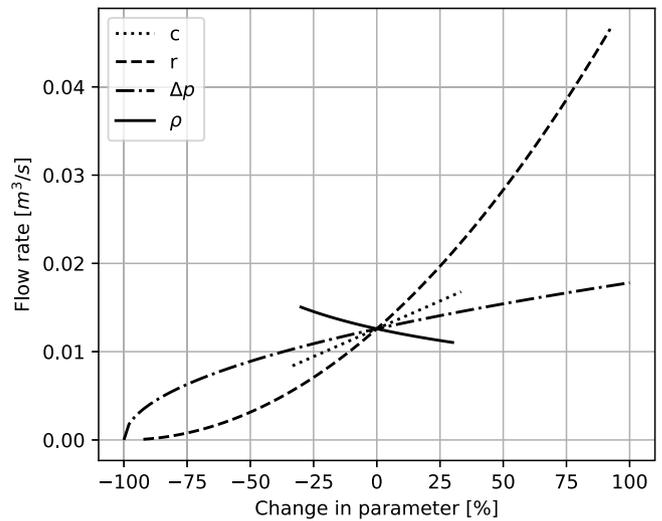


Fig. 1. Sample Spider plot of mathematical model's sensitivity.

- Δp is Pressure across the nozzle, in Pa, $\Delta p \in (0, 1000000)$
- ρ is Fluid Density, in kg/m^3 , $\rho \in (700, 1300)$

By using a spider plot, as seen in Fig. 1, it is visualized how the output value will change per percentage change in input. All inputs are set at their median value, considering their potential range, i.e. for discharge coefficient c the value is set at $(0.5 + 1)/2 = 0.75$. Keeping all other values at their median value, the output Q can be plotted as a function of c from -33.3 percent ($0.75 - 0.75 * 33.3\% = 0.5$) to $+33.3$ percent ($0.75 + 0.75 * 33.3\% = 1$). This is done in the same manner for all the inputs.

There is a lot of information that is hidden on this plot. For example, the change in *nozzle diameter* r suggests that the *flow rate* can be adjusted by this parameter between $0 \text{ m}^3/\text{s}$ and $0.05 \text{ m}^3/\text{s}$. This is however only true for the selected middle values for all the other parameters. More general statement can be made that doubling the *nozzle radius* will quadruple the *flow rate*, however the chart does not uncover that this is not valid when the pressure drop Δp is equal to zero. Most importantly however, OAT methods cannot cover a significant part of the high-dimensional Euclidean space (Saltelli and Annoni, 2010). The significance of this grows with the number of dimensions, which is especially high in case of recurrent neural networks.

2. Data-driven approach to sensitivity of data-driven models

In machine learning problems related to sequential data Recurrent Neural Networks (RNN) are often utilized (Yu et al., 2019). Abstracting from the specific internal structure of the network, the model takes multiple inputs, in the same order as they exist in a dataset. Model that will take into account last n values of a single input will technically have n inputs. This custom length will be driven by the performance of a given model as well as computational cost of training.

Neural Networks can have thousands of inputs without reaching a bottleneck in terms of computational power when training the model. Areas where machine learning often excels are commonly both highly dimensional with interdependent inputs, such as RNN and other complex models, which makes them unsuitable for classical sensitivity analysis methods. This paper proposes a number of approaches that were specifically designed for data-driven models with high number of inputs and outputs, addressing both of these issues.

A number of assumptions are made in the proposed method that stem from specific nature of data-driven models. This enables simple calculations and it leapfrogs shortcomings of the classical methods. The basis for proposed sensitivity analysis is the training, validation, and

testing dataset of the model. It is a requirement that this dataset represents the complete range of possible inputs and outputs relatively well. While this may be considered too strict or unrealistic requirement, by definition a data-driven model is applicable to inputs and outputs that are covered by the training material. Data-driven sensitivity analysis will be valid only for ranges of values represented by the dataset at hand, the same way as the data-driven model is valid only for the input data from the mathematical vicinity of the training set. It has to be stressed, that sensitivity analysis in any form is connected to the range of inputs and, additionally in our case, the distribution of the samples.

Typical training/validation/testing split of the dataset is removed in our method and complete dataset is used to maximize the amount of samples and therefore coverage of the inputs' hyperspace. While developing a data-driven model this split is introduced to ensure that the model can generalize and not simply *memorize* the provided samples. In contrast, our method utilizes the data as a set of representative and realistic inputs to the model, and therefore the split is no longer necessary.

Limiting calculations to the data points existing in the dataset, instead of all possible combinations, apart from significant reduction in calculation time, avoids diluting results with impossible inputs. Our case study consists of multiple sequential data inputs organized along the increasing depth, for example inclination change of the well or surface torque. These data are continuous and never rapidly oscillate between maximum and minimum value, only smoothly rises at different rates. Checking sensitivity for unrealistic scenarios is non-productive, since the model is not valid for them.

Second assumption is that the existing dataset is balanced in terms of probability of inputs. That is, if a given model predicts daily mean temperature, the number of samples should be distributed evenly across all twelve months. This is done to ensure that any data skew that may exist in the dataset does not translate to a skew in sensitivity analysis. Additionally, an artificial limitation can be introduced to a dataset, for example to explore sensitivity related to data exclusive to February.

If a dataset is imbalanced, methods commonly used for dealing with such problem in the domain of machine learning can be applied. Undersampling the overrepresented data can be performed, where some samples are simply removed. Duplicating underrepresented samples can also be done, however this should be done with caution, so that the dataset does not become overly skewed towards specific data. For example, if there are 1000 samples for weather data for each month except for February, duplication would be acceptable if it extends the sample size from 750 to 1000. On the other hand, should February have only 100 data points, such process would be questionable.

Presented SA methods, while are rooted in one-at-a-time methods, are applied to an existing dataset populating the complete hyper-space of probable inputs. This means that the computational complexity in big-O notation is $O(n \cdot m)$ with n being the sample count and m being the input count. The amount of outputs does not influence the computational complexity. Our case study consisting of 643 inputs, 100 outputs, and with a dataset of 1300 samples required just over 1 h to calculate all the results.¹ Our method is also considered embarrassingly parallel, i.e. it is trivial to perform calculations on multiple computers at the same time. Therefore, it can be easily implemented in frameworks such as Hadoop or Spark should the data size require that.

The result of the presented methods is a probability distribution. One-at-a-time methods are applied to the many samples of the dataset generating a set of individual results that, as a whole, present global sensitivity of the model.

¹ Calculations were performed on Intel Core i7-8850H, 32 GB RAM, Python 3.6.9, SALib library in version 1.3.8.

2.1. Method introduction, flow through a nozzle

Basics of the method are presented using the example of flow rate through a nozzle shown earlier in Eq. (2) before exploring an oilfield related case study utilizing recurrent neural networks. Dataset was generated for inputs for this model containing 1000 samples with normal distribution at the center of the range indicated for Eq. (3) and standard deviation equal to 0.2 of total range width. For example, samples for *density*, ranging from 700 to 1300 kg/m³, were randomly generated using normal distribution with mean value of 1000 kg/m³ and standard deviation of 120 kg/m³. Note that this is done for illustration purposes only in-lieu of real dataset. Systems in practice rarely follow pure Gaussian distribution hence using true dataset is important.

Sensitivity Indices were calculated using Eq. (2) with previously generated sample data as a starting point. Value of ΔS was set to 0.1; the influence of the value selection is explored further in the paper. The following calculations were performed for each of the four inputs to gauge their respective sensitivities. Fig. 2 shows the distribution of sensitivity indexes over the dataset. To better understand the data behind this Fig. 3 is also plotted, where sensitivity index is shown as a function of the input value. What may be surprising is that it is not a continuous line. This is because sensitivity index, as described by Eq. (2), deals with absolute values. For example, when pressure across the nozzle is high, small increase in its diameter will cause a large nominal flow increase. Alternatively, at very low pressures, even big nozzle increase will not yield significant change.

To check the relative impact on the output one may calculate sensitivity divided by the model output at nominal input value, which this paper refers to as relative sensitivity index:

$$RSI_{RS} = \frac{[R(S_0 + \Delta S) - R(S_0 - \Delta S)]}{2\Delta S} \cdot \frac{1}{R(S_0)} \quad (4)$$

This will produce a relation seen in Fig. 4, where a fixed change in nozzle size will affect relative flow output more at smaller diameters, i.e. change from 1.5 mm nozzle to 2.5 mm, and less for bigger nozzles, i.e. from 20 mm to 21 mm nozzle. Since this response is disconnected from the absolute value of the output, it produces a continuous line.

One can further explore the system response with Elasticity Index, or elasticity of a function (Sydsaeter and Hammond, 1995), given by Eq. (5) below.

$$EI_{RS} = \frac{[R(S_0 + \Delta S) - R(S_0 - \Delta S)]}{2\Delta S} \cdot \frac{S_0}{R(S_0)} \quad (5)$$

The equation was again applied to all the samples and all the inputs, with results plotted in Fig. 5. Elasticity Index in practice uncovers the exponent of the function, hence in case of Eq. (3) it is equal to 2. Derivation of this relationship is presented in Appendix.

When exploring the produced figures one can see a number of issues with the calculated data. First and foremost, the scale of the Sensitivity Index values varies by orders of magnitude. For *nozzle radius* it is between 0 and 4, and for *pressure drop*, intuitively a very important parameter for flow through a nozzle, it is in range of 10⁻⁸. Note that the units themselves are different (m³/s/m, m³/s/Pa, and others) which mean that the values cannot be compared.

While in the case of a simple equation the relative sensitivity index and elasticity index bring interesting additional information, in our case study focus will be on the basic sensitivity index. Nevertheless, it is beneficial, for the sake of presentation of the method, that all three equations will be explored. They are a natural next step, and are needed to understand why the simpler, and seemingly less informative equation is used.

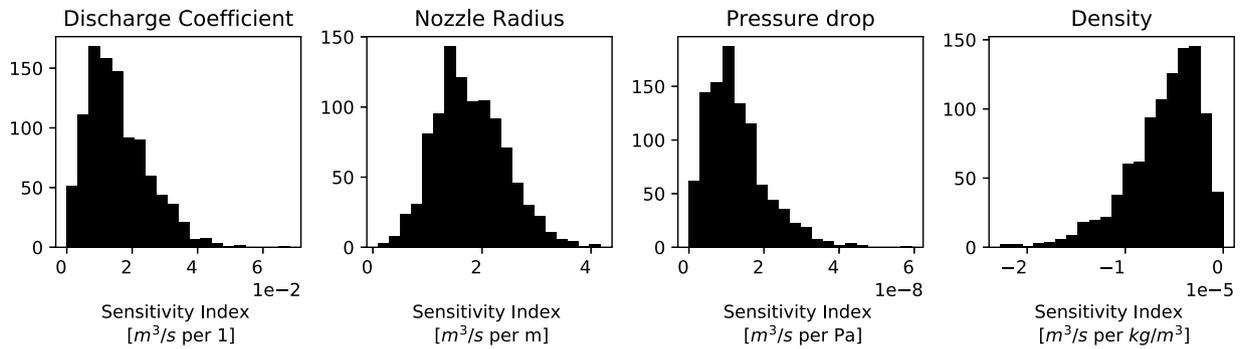


Fig. 2. Sensitivity index, histogram.

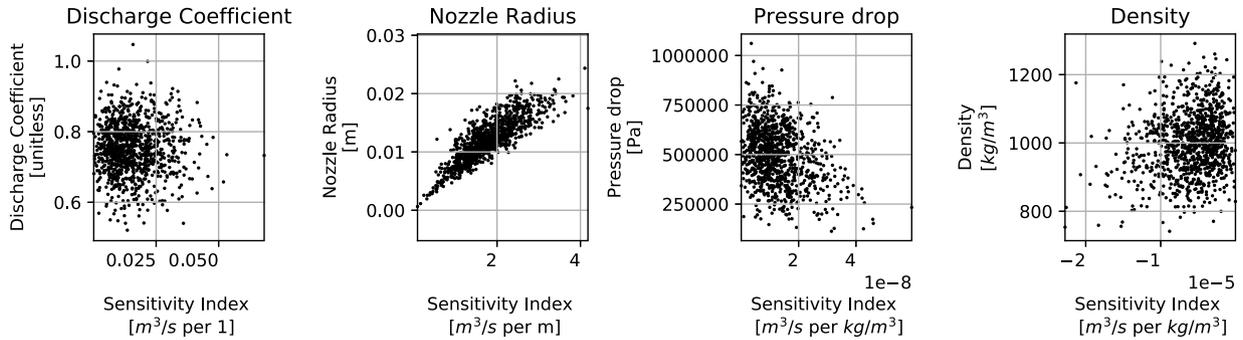


Fig. 3. Sensitivity index as scatter plot of dataset.

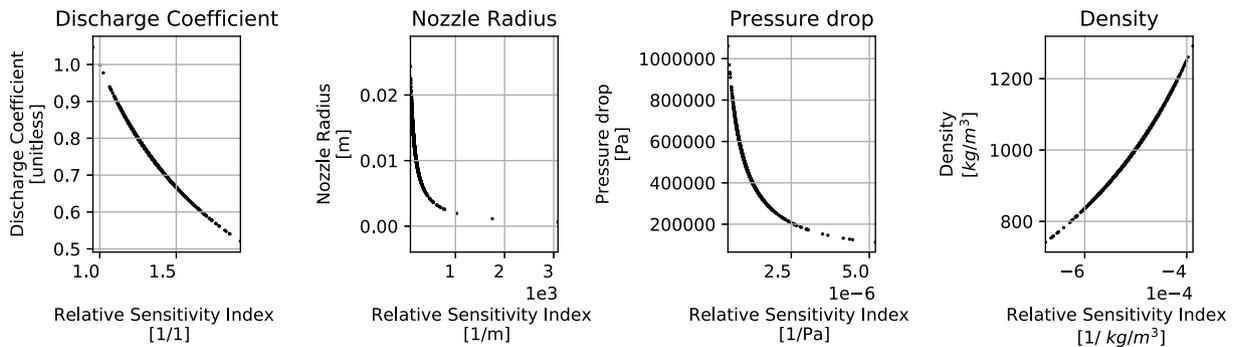


Fig. 4. Relative sensitivity index.

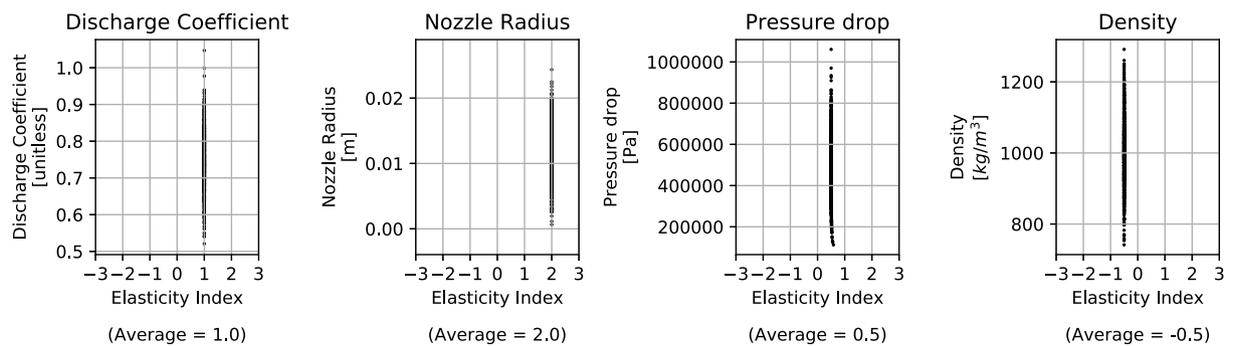


Fig. 5. Elasticity index.

2.2. Practical problem of scale

One problem that was identified while inspecting results produced in the previous section is units and scales. Referring back to Fig. 2 and

inspecting the x-axis it is clear that there is no straightforward information about which input has high sensitivity compared to others. This is the key information that sensitivity analysis is seeking to uncover when probing a given model.

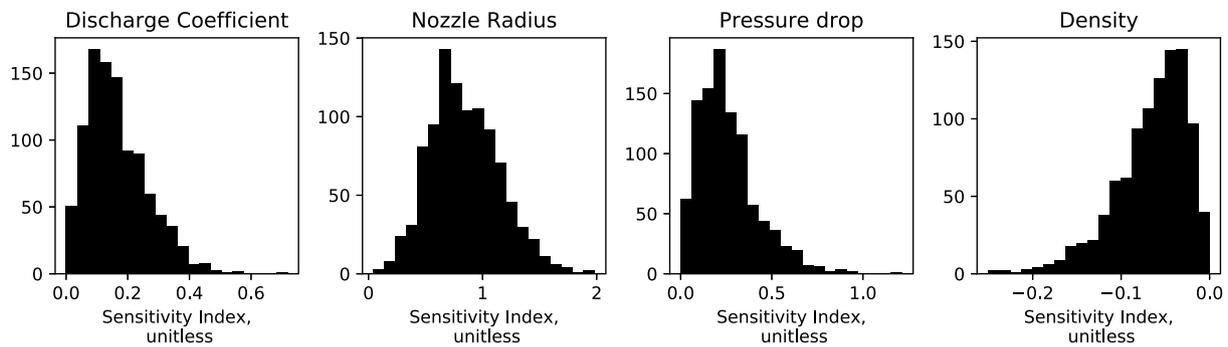


Fig. 6. Sensitivity index, scaled function, histogram.

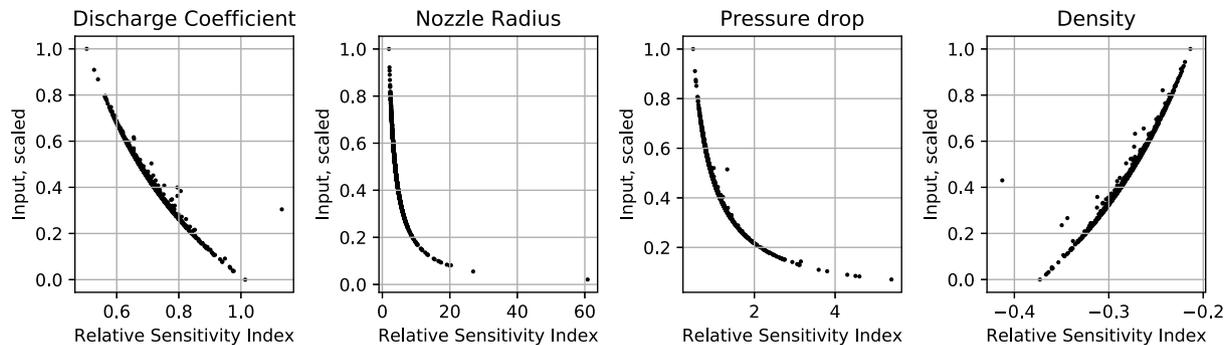


Fig. 7. Relative Sensitivity index, scaled function, scatterplot.

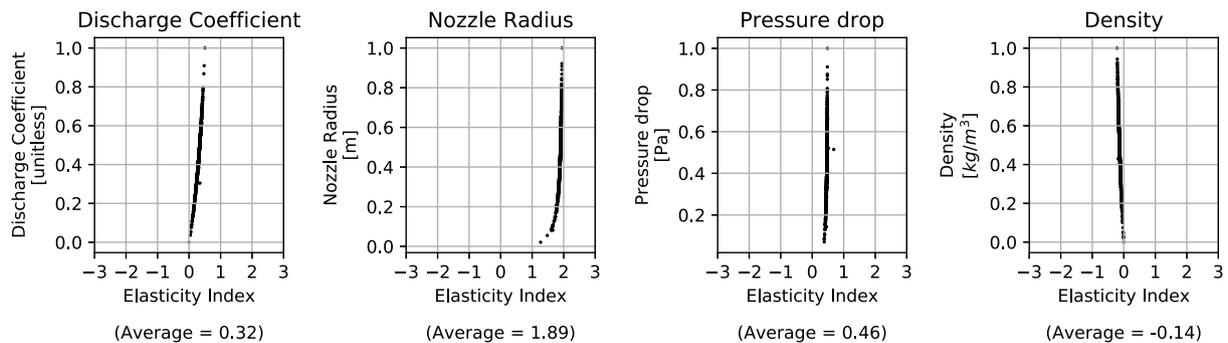


Fig. 8. Elasticity index, scaled function, scatterplot.

It is worth reiterating that machine learning models exist with a strong relationship to a dataset of samples, and in consequence, to the range in which inputs exist. Looking back at the sample nozzle flow Eq. (3), while mathematically sensitivity is defined by the equation alone, the application makes it tied to the range of inputs. For example, in practical terms, sensitivity of the *nozzle radius* input can be considered high if it is ranging from 1 to 50 mm, but it will be very low if it is bound between 5 mm and 5.1 mm.

Nozzle Equation (3) was encapsulated in a function such that all inputs are scaled between 0 and 1 in relation to generated samples. Output is also scaled between 0 and 1 for values generated from the samples. While this looks uncanny for a mathematical equation this is a very typical approach for a machine learning model. Histograms seen in Fig. 6 were generated the same way as described in the subsection before, and therefore are analogous to Fig. 2.

The scaling process did not affect the shape of the histograms in a significant way, but the sensitivity factor can now be evaluated in terms of expected range of inputs and outputs. For example, one can see that when radius input is changed by 10 percent of full scale, output can be expected to also change by 10 percent, since the most common

sensitivity value on the histogram is 1. In case of *discharge coefficient* sensitivity is smaller. 10 percent change will typically affect the output by 2.5 percent of expected full scale (one to 0.25). This can be tied to the spider plot back in Fig. 1, where the model response is presented. Linearly approximating the curves one can roughly see the same values as presented here — the steeper the curve the higher the sensitivity index. Note that this is a distribution of results, hence the response discussed above should be considered mean sensitivity of the model.

After scaling, inputs and outputs should be considered unitless with a scaler attached to them that describes the scaling parameters, i.e. original minimum and maximum values. Therefore the sensitivity index calculated with scaled data becomes unitless as well. Alternatively, scaled values can be considered percentage of range, making the unit of sensitivity index to be a percentage of range of output per percentage of range of input. This method was used before in relation to sensitivity analysis of neural networks (Lu et al., 2001) highlighting the same problems related to units and scale.

Analyzing scatter plots in Fig. A.27 showing relationship between the sample value and sensitivity index there are no immediate differences due to scaling of the dataset. For brevity this figure is reproduced in the Appendix.

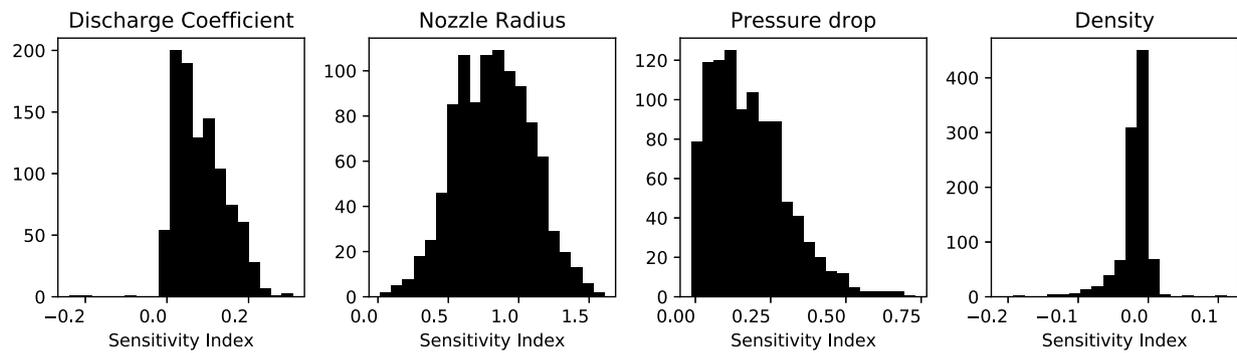


Fig. 9. ML model Sensitivity Index histogram.

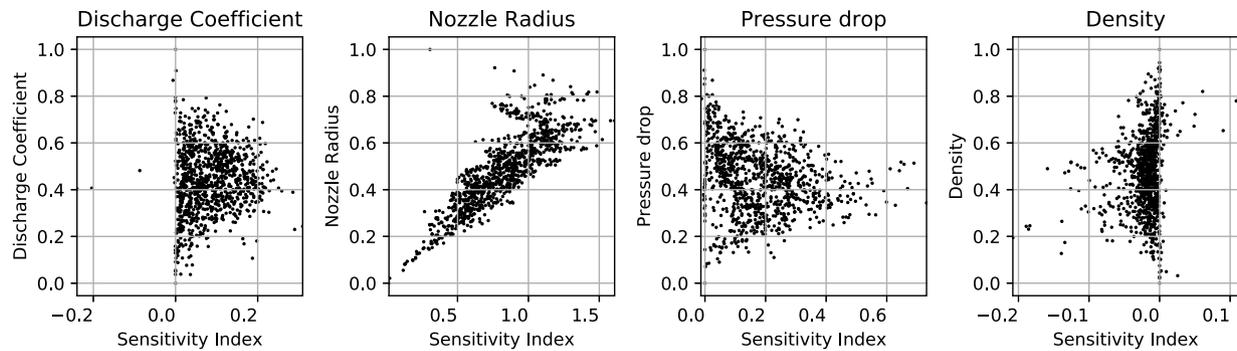


Fig. 10. ML model, scatter plot of Sensitivity Index.

Moving to Relative Sensitivity Index in Fig. 7 and comparing to non-scaled charts seen previously in Fig. 4 one can notice that the result is no longer a continuous line. There is some disturbance introduced due to scaling and this effect is even stronger when inspecting Elasticity Index in Fig. 8, where EI is strongly pulled towards zero at inputs close to zero. The average still indicates what kind of function is behind the model, but it is no longer precise, as indicated by the average values below the plots.

Analysis of results of the scaled inputs and outputs is easier, as it can be done in relation to the full scale of these inputs and outputs. It is easy to see that sensitivity to the *nozzle radius* is highest, followed by *pressure drop*, *discharge coefficient* and with *density* being the lowest, which additionally is identified as inversely correlated with the output.

2.3. Applying method to a machine learning model

While sensitivity index and the other methods used provide clear insights to the inner working of a model when applied to simple equations, a study was done in relation to machine learning models. To evaluate method's suitability for such scenario, an ML model of the previously used flow through a nozzle equation was created using a Random Forest regression model (Breiman, 2001) made with previously generated samples. Common best practices were used while making the model, including a train/test dataset split (here 50/50 split) as well as basic hyperparameter tuning to find the optimal amount of estimators and forest depth.² Model results of the testing portion of the dataset evaluated against the equation output have an R^2 value of 0.976. This can be considered a very good fit for a machine learning model and can be considered representative of performance of the best case real-world ML models.

Scaling of inputs and outputs in the range of (0, 1) was also applied. This is considered standard practice when creating ML models,

although not strictly necessary for the algorithm used here. The transformation was nevertheless applied, since our method of evaluating sensitivity works best on a scaled model. Note that there is no practical reason to create an ML model of a known equation and here it is done only for the purpose of evaluating the method.

Same type of analysis was performed on the ML model as it was on the mathematical model to evaluate how the intrinsic inaccuracies of the ML model influence the results. Figs. 9 and 10 showing results for the ML model correspond to Figs. 2 and 3 produced for the equation-based model. Most important takeaway is that the histograms of sensitivity indices for both models show approximately the same shapes and values, while the scatter plots are significantly different. When calculated for mathematical model, scatter plots for all but one variables seen in Fig. 3, showed approximately Gaussian distribution along both axes. For the ML model, Fig. 10, scatter plots are significantly different taking various shapes that were not present before.

Relative Sensitivity Index was also calculated for the ML model, seen in Fig. A.28; same charts for mathematical model are in Fig. 4. Clear smooth lines are no longer visible, except for the *nozzle radius*. For other three parameters it is hard to identify even the direction of the original curve. This means that Relative Sensitivity Index is very susceptible to the inaccuracies even in a case of very accurate ML model and its usefulness will be limited for data-driven models. Lastly, the Elasticity Index was analyzed, as seen in Fig. A.29; mathematical model-based results are in Fig. 5. Average values became closer to zero than in the mathematical model with scaling and data is overall difficult to interpret. Conclusions related to the exponent of the equation used can no longer be drawn.

The conclusion from ML model analysis is that our proposed method focuses mainly on the distribution of the sensitivity index. Calculating additional metrics, such as relative sensitivity index and elasticity index in our experience did not bring forward any further valuable information.

² Identified values are: $n_estimators = 93$, $max_depth = 3$.

2.4. Uneven distribution

Sensitivity analysis is closely tied not only to the range of inputs, but also to their distribution. To uncover it, another simulation was performed with the nozzle flow equation, but this time the distribution of the *discharge coefficient* was changed, with 800 samples generated at fixed value of 0.5 and 200 samples generated at a fixed value of 1, making a strongly bi-modal distribution. This does not change the range of the inputs, but significantly alters how the model behaves on average.

For a direct comparison (0, 1) scaling was performed and machine learning model created. Analysis of histograms and scatter plots (Figs. A.30 and A.31, Appendix) shows reduced sensitivity of all inputs. This is caused by overrepresentation of samples with low *discharge coefficient* and therefore low flow rate, and in return — low sensitivity in terms of percentage of output range. This situation can only be uncovered via analysis on the inputs' distribution, and later by analyzing sensitivity for both clusters separately. This is outside of the scope of this study.

2.5. Summary of the method

Different aspects of dataset-based sensitivity analysis were evaluated on a synthetic example in the previous sections. Our conclusion is that the most condensed information is retained within median value and distribution of Sensitivity Index when calculated on a (0, 1) scaled values. Coincidentally, scaling of values, both inputs and outputs, is a standard practice in machine learning, making application of the method very simple.

Relative sensitivity index was found to not bring significant additional information when exploring even our very accurate machine learning model. Furthermore, this parameter is best evaluated on a scatter plot, what becomes impractical when hundreds of inputs are being analyzed. Elasticity Index can potentially bring information about the type of equation behind the model, it is however sensitive to scaling and can suggest misleading values. Additionally, it cannot be used when probing multiple inputs, which is useful in case of recurrent neural networks, as presented in further chapters.

2.6. Pseudocode

To aid in the implementation of the presented method pseudocode is provided in Algorithm 1 demonstrating calculation of Data-Driven Sensitivity Index. Note that the output of the algorithm is a 3 dimensional array that contains all local sensitivity indices for all samples, inputs, and outputs. One can visualize this data in a number of ways, for example calculating median sensitivity of individual inputs by calculating it over samples and outputs such as in Fig. 17. In this case, inputs along one channel were probed individually.

If a particular input is of interest, median sensitivity on outputs can be calculated along samples on that one input, such as in Fig. 18, where one particular input in the inclination channel is investigated. It is possible to visualize data as a heatmap, where median sensitivity is calculated of all samples per individual inputs and outputs, resulting in a 2D map of sensitivity between inputs and outputs, as seen in Figs. 20 and 21. If samples are sorted by a meaningful dimension, depth, time, distance, etc., it may be useful to calculate mean of all inputs, retaining the sample and input dimension, to investigate the distribution of sensitivity along that meaningful dimension. In our case study, such analysis was performed and is presented as a 2D map in Fig. 22.

Pseudocode for calculating sensitivity index per complete channel is provided in Algorithm 2. It is particularly useful in relation to RNNs, where inputs are often given in a series along depth or along time. It can also be suitable when inputs may be grouped together for other

Algorithm 1: Data-Driven Sensitivity Index (DDSI), per single input

```

Data:  $X := [X_1, X_2, \dots, X_n]$ 
 $X_n := [x_{n,1}, x_{n,2}, \dots, x_{n,m}]$ 
an array of n samples with m inputs
 $\Delta$  typically 0.1, or 10% of input range
Result: DDSI :=  $[SI_1, SI_2, \dots, SI_n]$ 
 $SI_n := [SI_{n,1}, SI_{n,2}, \dots, SI_{n,m}]$ 
 $SI_{n,m} := [si_{n,m,1}, si_{n,m,2}, \dots, si_{n,m,w}]$ 
an array of individual, local Sensitivity Indices, through n samples, m inputs, and w outputs.

1 Function Model( $[x_{n_i,1}, x_{n_i,2}, \dots, x_{n_i,m}]$ ):
2   | model calculations;
3   | return  $[y_1, y_2, \dots, y_w]$  ;
4 for  $i=1$  to  $n$  do
5   | for  $j=1$  to  $m$  do
6   |   samplePlus =  $X[i]$ ;
7   |   samplePlus[j] +=  $\Delta$ ;
8   |   outputPlus = model(samplePlus);
9   |   sampleMinus =  $X[i]$ ;
10  |   sampleMinus[j] -=  $\Delta$ ;
11  |   outputMinus = model(sampleMinus);
12  |   oneSensitivity = (outputPlus - outputMinus) /  $2\Delta$  ;
13  |   DDSI[i,j] = oneSensitivity;
14  | end
15 end

```

reasons, for example in weather models sensors can be grouped into temperature, barometric, and rainfall channels.

The output of the algorithm is a 3 dimensional array, with dimension for samples, channels, and outputs. Visualizing one channel at a time, one can visualize distribution of channel's sensitivity via 2D heatmap, as in Fig. 13, or potentially easier to read median and percentile statistics reducing the sample dimension, as seen in Fig. 14.

3. Case study model — inclination in directional drilling

3.1. Introduction of the model

To evaluate proposed methods of sensitivity analysis a case study machine learning model was employed which was previously used for prediction of continuous inclination data in directional drilling (Tunkiel et al., 2020); data and source code is available on GitHub (Tunkiel). This model predicts inclination values that are artificially made lagging 23 m behind the bit. It is a mixed approach of trend prediction as well as regression by correlation with other parameters assumed available real-time. Flowchart from aforementioned paper showing simplified model structure is shown in Fig. 11.

Data were taken from the Volve dataset open sourced by Equinor (2018), which covers an offshore oil field off the coast of Norway. Part of a well F9 A, starting from 500 m to 845 m measured depth was selected as containing the longest directional drilling section performed with a bent motor. Three parameters assumed available all the time were selected in addition to the past inclination data: *Surface Torque*, *Rotary Speed* and *Rate of Penetration*. Data were split into training and validation, 500 m to 800 m, with 80/20 split, and testing, 800 m to 845 m, which were used only for final score calculation.

All inputs and outputs of the model are scaled between 0 and 1, and all calculations presented in this paper are done in relation to those scaled values. Additional normalization step was introduced to the inclination input channel. To normalize the samples in relation to

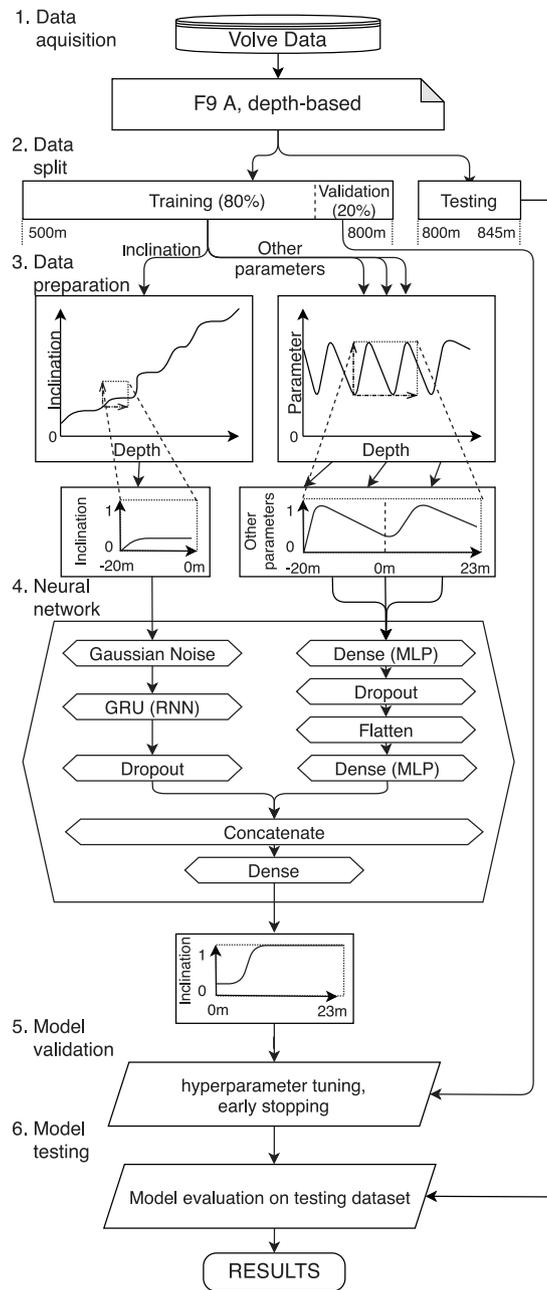


Fig. 11. Case study model flow chart.

absolute inclination, a local coordinate system was employed. This way, inclination input channel samples always start at zero, see Fig. 12 for reference. Global scaling of the inclination was selected such, that after introducing the local coordinate system the maximum value of inputs in the inclination channel should be maximum 1.

The model contains a branched neural network, with one branch with Gated Recurrent Unit (Cho et al., 2014) RNN taking 85 inputs, the inclination data, and a second branch with multi-layer perceptron (MLP) taking 3×186 inputs from the additional available parameters. Inputs and outputs are considered in terms of relative position from the introduced start of the gap. The RNN branch receives inclination values from position -20 m to zero meters relative to the gap. The MLP branch receives values from three different measurements, the *Rotary Speed*, *Surface Torque*, and *Rate of Penetration*. That data are input from position of -20 m to $+23$ m relative to the gap. The model returns 100 output values as prediction of individual inclination values at location from 0 m to $+23$ m.

One can consider this model as utilizing four input channels and one output channel. This way the model can be reduced, for the purposes of the analysis, to four separate vector inputs and one vector output. Example results from the case study model can be seen in Fig. 12. In the top subplot dashed line denotes inclination data input, while the solid line is a prediction, and the dotted line is ground truth. The other three plots present the surface parameters that are considered available all the time. Note that the output is a continuous line. This result is not forced through algorithm, but achieved purely through neural network training.

For the purpose of the sensitivity analysis the case study model was considered as follows:

$$Y = f(X) \tag{6}$$

To reflect the structure of the input consisting of four distinct channels, further expansion to $X_t = f(X_t^A, X_t^B, X_t^C, X_t^D)$ is made, where:

Algorithm 2: Data-Driven Sensitivity Index, per channel

```

Data: X := [X1, X2, ..., Xn]
Xn := [Xn,1, Xn,2, ..., Xn,k]
Xn,k := [Xn,k,1, Xn,k,2, ..., Xn,k,mk]
an array of n samples with k channels, and mk inputs on
k-th channel.
Δ typically 0.1, or 10% of input range

Result: DDSI := [SI1, SI2, ..., SIn]
SIn := [SIn,1, SIn,2, ..., SIn,k]
SIn,k := [SIn,k,1, SIn,k,2, ..., SIn,k,w]
an array of individual, local Sensitivity Indices through n
samples, k input channels, and w outputs.

1 Function Model(
  [[xni,1,1, xni,1,2, ..., xni,1,mk],
  [xni,2,1, xni,2,2, ..., xni,2,mk],
  ...,
  [xni,k,1, xni,k,2, ..., xni,k,mk]]);
2   model calculations;
3   return [y1, y2, ..., yw];
4 for i=1 to n do
5   for r=1 to k do
6     samplePlus = X[i];
7     sampleMinus = X[i];
8     for j=1 to mk do
9       samplePlus[r][j] += Δ;
10      sampleMinus[r][j] -= Δ;
11    end
12    outputPlus = model(samplePlus);
13    outputMinus = model(sampleMinus);
14    oneSensitivity = (outputPlus - outputMinus) / 2Δ;
15    DDSI[i,r] = oneSensitivity;
16  end
17 end

```

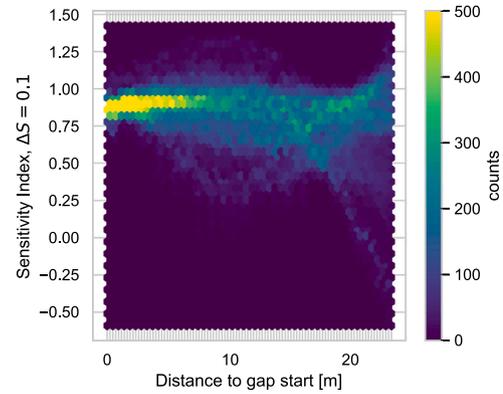


Fig. 13. Statistical sensitivity to 10 percent value change on Inclination channel, heatmap.

where

$$Y_t = \begin{bmatrix} y_{t-m_0|t}, \dots, y_{t-1|t}, y_{t|t} \end{bmatrix}^T \quad (8)$$

$$X_t^A = \begin{bmatrix} x_{t-m_1|t}^A, \dots, x_{t-m_0-2|t}^A, x_{t-m_0-1|t}^A \end{bmatrix}^T \quad (9)$$

$$X_t^n = \begin{bmatrix} x_{t-m_1|t}^n, \dots, x_{t-1|t}^n, x_{t|t}^n \end{bmatrix}^T, \quad n = B, C, D \quad (10)$$

and $m_0 = 100$ and $m_1 = 186$, $y \in R$ and $x \in R$.

3.2. Sensitivity index of input channels

To analyze the model top to bottom, sensitivity of the complete channel is explored first, i.e. related to multiple inputs at the same time. This is done the following way:

$$SI_t^A = \frac{f(X_t^A + \Delta S, X_t^B, X_t^C, X_t^D) - f(X_t^A - \Delta S, X_t^B, X_t^C, X_t^D)}{2\Delta S} \quad (11)$$

where SI_t^A is sensitivity of the channel A, and

$$\Delta S = \begin{bmatrix} \Delta s, \dots, \Delta s, \Delta s \end{bmatrix}^T, \quad \Delta s = 0.1 \quad (12)$$

This generates multidimensional results and there are many ways to visualize them. The model is predicting 100 continuous data points (23 m), hence they are plotted along x axis as a distance to the start of data gap. Complete results can be presented as a heatmap, as seen in Fig. 13, however this makes quantitative analysis difficult. It is clear that sensitivity is approximately 0.9 at distance to data gap 0 (bright yellow area) and becomes more spread out further along, but it is in general hard to read. To facilitate evaluation of the results median, 15th, and 85th percentile, maximum, and minimum values were used. In Fig. 14 results are presented from calculating sensitivity index related to the complete inclination channel. Referring to Eq. (2), all inputs in the inclination channel are varied by $\Delta S = 0.1$, which is 10 percent of the total input scale, as the inputs are scaled between 0 and 1. This is akin to using a miscalibrated sensor where an offset is introduced to all inputs on a channel.

Particularly interesting insight is that while sensitivity of the inclination input starts slightly below 1 with very little change in median value, other channels' sensitivity index starts at zero for the first predicted data point (one example in Fig. 15) and the magnitude of it rises as the prediction is done for further points. Both of those results indicate that the developed model acts as it should. The output is a continuous prediction, a continuation of the inclination input, therefore necessarily if whole input channel is shifted by a given value, the first

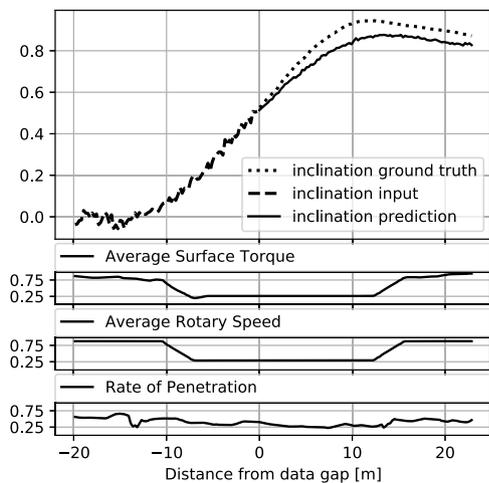


Fig. 12. Input and output sample from the case study model.

X^n denotes the n th input channel. Let us consider the model to be:

$$Y_t = f(X_t^A, X_t^B, X_t^C, X_t^D) \quad (7)$$

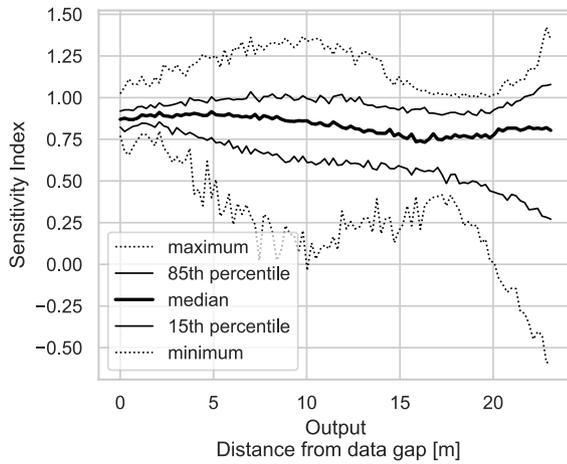


Fig. 14. Statistical sensitivity to 10 percent value change on Inclination channel, distribution.

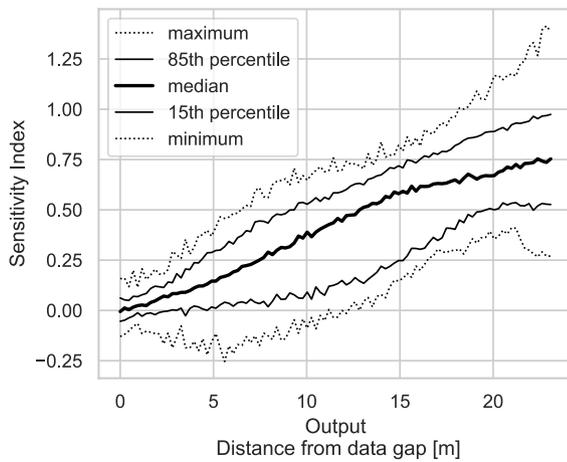


Fig. 15. Statistical sensitivity to 10 percent value change on Torque channel.

output will shift by nearly identical value. Inclination cannot change sharply, only gradually, hence near zero sensitivity on the other three channels for the first prediction point.

For channels other than inclination, sensitivity index grows as the prediction distance increases. This is consistent with the modeled process, as the inclination can be influenced more in the outputs farther from the initiated gap.

3.3. Magnitude of change

In calculating sensitivity index there is a potential influence of the ΔS value. To evaluate this matter, sensitivity index was calculated at $\Delta S = 0.1$ and $\Delta S = 0.5$. It must be noted that this value corresponds to scaled parameters, hence are equivalent to 10 percent and 50 percent of full scale change respectively. Results can be inspected in Fig. 16. Median sensitivity can be considered the same for both selected ΔS . The main difference is that the result is more stable, i.e. the 15th and 85th percentile values are closer to median. This suggests that the selection of specific ΔS does not affect the results of the analysis significantly.

3.4. Sensitivity to single inputs

Another type of sensitivity analysis proposed is sensitivity to single inputs, which can also uncover sensitivity to outliers or sensitivity to a

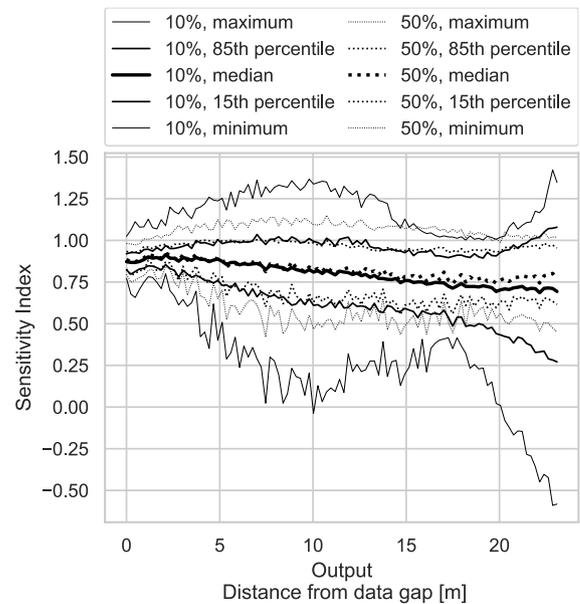


Fig. 16. Evaluation of the importance of magnitude of change in sensitivity analysis, inclination channel.

specific input. A function was developed that calculated the sensitivity index, as per Eq. (2), again using the existing dataset of inputs.

$$SI_t^A(k) = \frac{f(X_t^A + \Delta S_k, X_t^B, X_t^C, X_t^D) - f(X_t^A - \Delta S_k, X_t^B, X_t^C, X_t^D)}{2\Delta s} \quad (13)$$

where

$$\Delta S_k = \begin{bmatrix} 1 & 2 & \dots & k-1 & k & k+1 & \dots & 643 & T \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & \end{bmatrix} \quad (14)$$

Per given sample only one singular input was modified. This was done for all singular inputs and channels separately, from the first input to the last one, altering original value by 0.1.

To provide a two dimensional, easy to read chart, for this method sensitivity that is averaged over all outputs is used. This way, an indication of how all the outputs change on average in relation to a change on a specific input is provided. For further clarity result plots are bundled into input channels. Two charts are reproduced here, one for the inclination channel, Fig. 17, and one for the torque channel, Fig. 19. What is interesting in the first figure, is that there is a decrease in value close to the input points at about -3 m from the gap. Additionally, points farther than -10 m have much lower response, which suggests that it may be possible to shorten the input to the model without a significant loss in performance. It must be noted, though, that the case study model was developed through hyperparameter tuning and the input length of 23 m was selected as providing the smallest error. Fig. 17 suggests that this window can be shortened with minimal influence on the output, and therefore minimal influence on the error. Such changes may be considered if, for example, shortening that input would generate significant savings in the deployment phase of a project. It may also be used in guiding sensor accuracy selection, by showing relationship between measurement error and output. Additionally, charts similar to Figs. 17 and 19 may be generated while tuning hyperparameters of a data-driven model, giving an instant insight in the inner workings of the neural network, allowing to narrow down a range of possible hyperparameters.

To explore results from Fig. 17 further, a chart was generated that investigates the point of maximum sensitivity at approximately -1 m from data gap, presented in Fig. 18. This allows us to evaluate how the point of maximum sensitivity, in relation to all outputs, is distributed between individual outputs. One can see that output at approximately

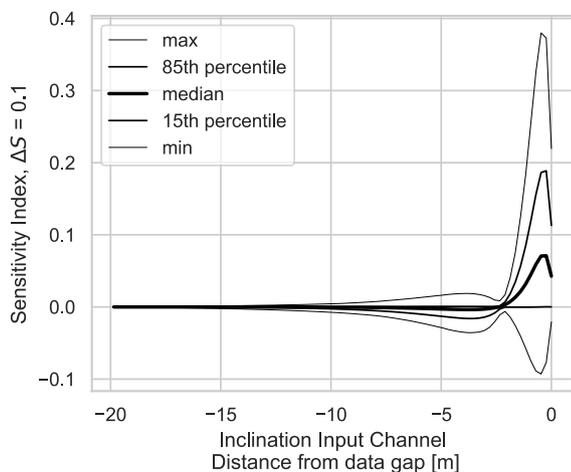


Fig. 17. Output channel sensitivity to a change in a singular point on inclination input channel.

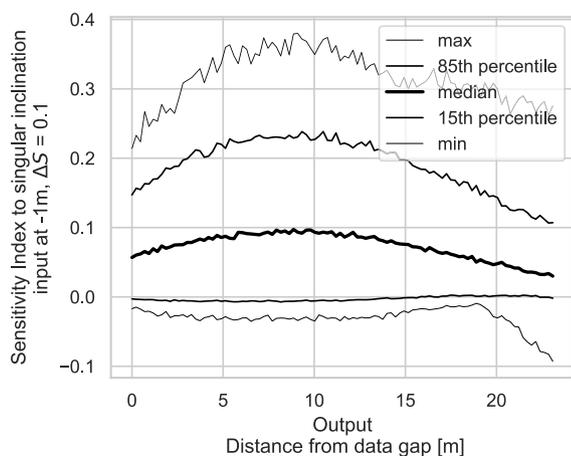


Fig. 18. Sensitivity to inclination input at -1 m to data gap, point of maximum sensitivity.

8 m after the data gap is the most sensitive, with further points' sensitivity dropping. This is in line with previously explored sensitivity of the inclination channel as a whole in relation to individual outputs, Fig. 14. Additional takeaway here is that 15th percentile line lies almost perfectly at zero sensitivity, suggesting that there is a relatively sharp cutoff with sporadic negative sensitivity, which is in line with the physical aspect of the model, i.e. higher inclination just before the data gap will suggest higher inclination after it.

Similar analysis was performed for input channel related to torque, with results in Fig. 19. Note that this channel has more inputs, as in our case study this parameter is considered available both before and after the data gap, while the inclination only before. Interestingly, there is a clear minimum between -6 m and 3 m. This is approximately the same area that has a local maximum in the inclination input. Higher response is visible for the points from 0 m forward. This again confirms the intuitive behavior of the model. Physically, the inputs above 0 m are co-located with the output, and therefore logically will have the highest impact.

Another interesting insight here is that again the 15th percentile line often lies at zero change, also in line with the physics of the modeled system. Attention must be paid to the scale of the y-axis on these charts, as the sensitivity to the inclination input channel is an order of magnitude higher than the one calculated for rotary speed.

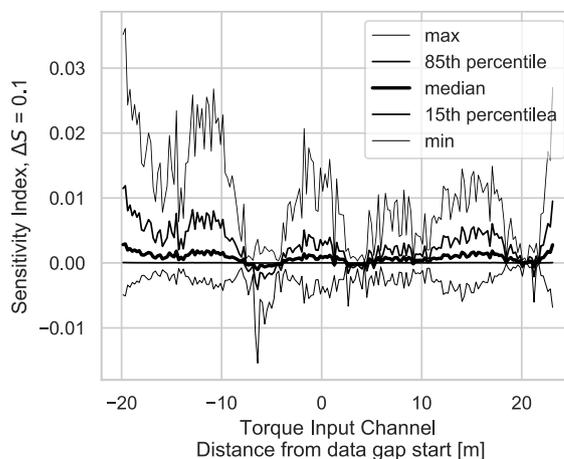


Fig. 19. Output channel sensitivity to a change in a singular point on torque input channel.

3.5. Input-output heatmap

It is possible to plot median sensitivity of outputs to specific input channel as a heatmap. This allows to see the influence of individual inputs to individual outputs. It must be highlighted that this is done on a point-by-point basis, one-at-a-time. Note that in this case the color scales are not synchronized between the charts as the results vary by an order of magnitude. Due to nature of the plot it is the median sensitivity calculated over the complete dataset that is visualized.

Looking at Fig. 20 one can consider it an expanded version of Fig. 17, with the output dimension now visible. One can, for example, identify, that inclination input at -5 m from the gap positively influences the output close to the gap, but negatively influences the outputs further from the gap. Similarly, in Fig. 21, the weakest response between inputs between -6 m and -3 m can be identified, first seen in Fig. 19, as a white plateau over the whole range of outputs. In this figure one can also identify a potential problem with the analyzed model. As explained before, this input physically overlaps with the output. Therefore input at location of n meters cannot have influence on the outputs before location n meters. There practically should be a triangle between points (0, 0), (23, 0) and (23, 23) with zero sensitivity — yellow overlay in that area was added in Fig. 21.

This apparent contradiction can be explained through correlation and causation. It is likely that during operation it is the achieved inclination (here output) that influences the rotary speed (here input), and not the other way round. The correlation however still exists and the model can use it. This means that the model will correctly predict inclination during typical drilling, but will fail if the inputs will deviate too much from normal operation. This is understandable considering that the training dataset covers only typical drilling.

3.6. Channel sensitivity as a function of well depth

The core of the presented method is statistical analysis of the models' sensitivity with the training/validation/testing dataset at its foundation. It is, however, possible to investigate how the calculated statistical sensitivity is affected by the contents of available dataset. Our case study models BHA-formation interaction while drilling a curved section of a well. Sensitivity analysis can be performed on a channel as a whole, as done in the section Sensitivity Index of Input Channels, and evaluated along the length of the well. Results are presented in Fig. 22. The sensitivity analysis was performed for a complete channel, as previously presented in Figs. 14 and 15 and using Eq. (11). Sensitivity was calculated separately for all four channels. The heatmap presented in Fig. 22 shows the median value of sensitivity for a given output.

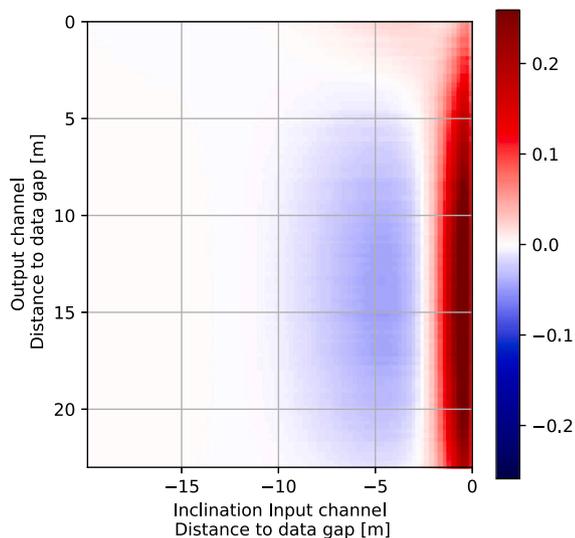


Fig. 20. Sensitivity heatmap between Inclination input channel and output. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

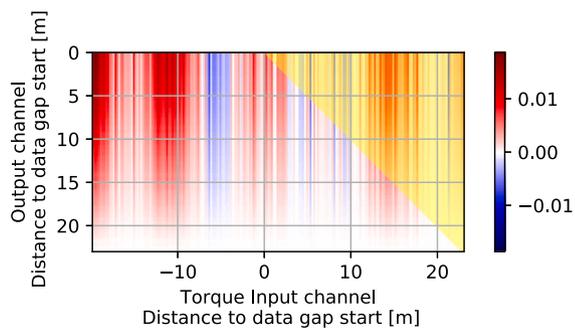


Fig. 21. Sensitivity heatmap between Torque input channel and output. Yellow overlay shows area that should have sensitivity zero. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Such a graphic enables evaluation of sensitivity changes with the varying base value in the dataset. It is possible to identify areas of significantly high or low sensitivity and investigate those specific samples. This can provide further insights of how the system behaves — in our case study it is possible separately analyze the sensitivity during rotating and sliding modes of the BHA. It is also possible to evaluate how well the dataset represents potential inputs of the system by looking for unusual responses, with an assumption that unusual sensitivity response coincides with unusual set of inputs. In our case this can be seen in the inclination channel between samples 500 and 600 and further on at approximately sample 950, where there is negative inclination channel sensitivity in outputs close to 23 m. This is the area responsible for minimum sensitivity below zero visible earlier in Fig. 14.

3.7. Comparison to Sobol' indices

Sobol' indices developed in the early 90s (Sobol, 1993) are a form of variance-based global sensitivity analysis. This method is used to calculate how much of the output's variance can be attributed to different inputs. With this method one can calculate how important are individual inputs, and if second order indices are calculated, how important is interaction between two inputs.

Our data-driven sensitivity analysis method used $n = 1300$ samples. To calculate sensitivity index 2n model evaluations must be performed,

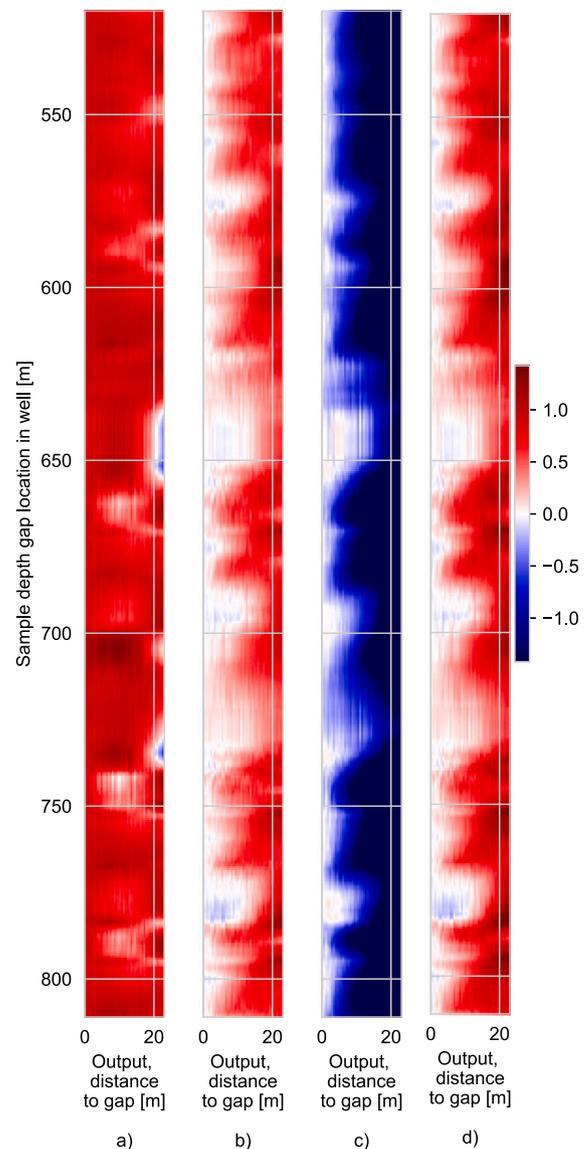


Fig. 22. Channel sensitivity per sample, for: (a) Sensitivity over Inclination channel, (b) Surface torque channel, (c) Rotary speed channel, (d) Rate of Penetration channel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

as a base value has to be increased and decreased. With 643 inputs, to calculate sensitivity for all of them individually, this results in 1 674 400 evaluations. For comparison, calculated Sobol' indices were calculated using SALib Sensitivity Analysis Library in Python (Herman and Usher, 2017), using methods developed by Sobol (2001), Saltelli (2002) and Saltelli and Annoni (2010). $n = 5000$ samples were used, limited by memory in available computer. Note that in common practice 10^4 samples are required to estimate Sobol' indices within 10% uncertainty (Iooss and Lemaître, 2015). Our setup resulted in 3 225 000 sub-samples for first order calculations. It took 10 min to generate required samples, 38 min to evaluate the model for all the samples and 40 min to calculate the resulting indices.³ Second order calculations require double the amount of samples and were not calculated. Other, faster algorithms were considered, however in general literature suggests that those methods remain costly, unstable and biased for models

³ Calculations were performed on Intel Core i7-8850H, 32 GB RAM, Python 3.6.9, SALib library in version 1.3.8.

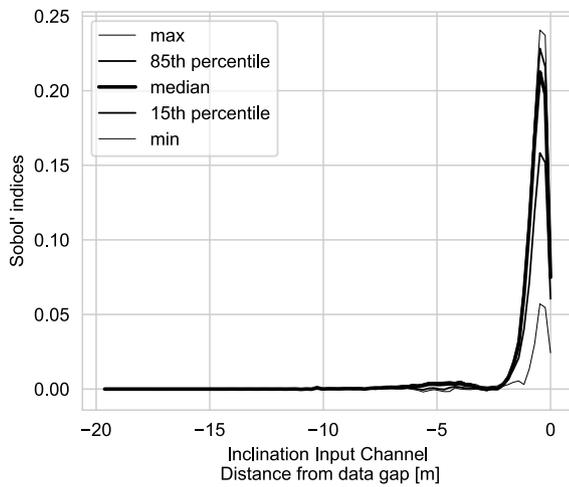


Fig. 23. Average sobol indices for inclination inputs.

with more than 10 inputs (Tissot and Prieur, 2012; Iooss and Lemaître, 2015).

While computation time was not significantly different between our method and Sobol' indices, our method allows for evaluating partial results, that is sensitivity of a single input can be calculated for a model with thousands of inputs very quickly. Note that since our case study model has 100 outputs representing 100 individual depth-steps in the model, each input will have 100 Sobol indices related to all separate inputs. Sobol indices are calculated assuming singular output; since our case study produces 100 outputs as a complete depth-sequence prediction, Sobol indices had to be calculated as if there were 100 separate models with single output to evaluate. Minimum and maximum boundaries for Sobol' indices calculation were taken from boundary values for specific input found in the dataset. This is typically in range (0, 1) for all inputs except for the ones responsible for inclination input. Due to scaling and moving the center of coordinate system employed in the model they vary in minimums and maximums.

For comparison purposes, a number of figures that were originally developed with our proposed method were recreated to compare results. Note that it is not possible to evaluate the sensitivity of a complete channel (multi-input) with Sobol' indices, hence it is not possible to compare those. First comparison is between our Figs. 17 and 23 generated with Sobol' index data. It can be seen that results are different. First, Sobol indices in principle cannot be negative, hence negative sensitivity at -5 m is not seen. The chart overall looks much more noisy and difficult to read.

Chart type that was used to investigate the maximum sensitivity on that channel, Fig. 18, was also plotted again, which can be seen in Fig. 24. Note that there is no statistical distribution to the new chart, as at this level singular Sobol' indices are being plotted.

As with previous comparison, general insights are still here, but some of the information available with our data-driven method is not there. Our method uncovers the spread in sensitivity, is based on actual data, and additionally uncovers that in some rare cases the sensitivity can be negative, which is not visible using Sobol' indices.

Lastly, the Sensitivity heatmap for inclination channel, Fig. 20, was re-done as a heatmap of Sobol' indices in Fig. 25. When using our method one can see exactly which areas are sensitive, and which are not. Using Sobol' indices this information is again missing. While our method uncovered an area of negative sensitivity, it is nearly invisible in Fig. 25, and the information about the sign of sensitivity is necessarily gone.

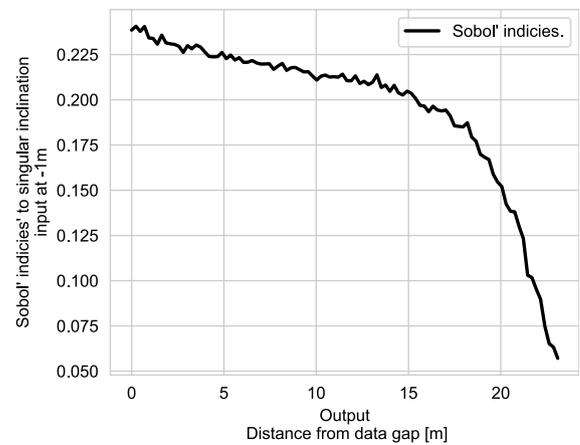


Fig. 24. Sobol indices for all individual outputs for inclination input at -1 m from data gap.

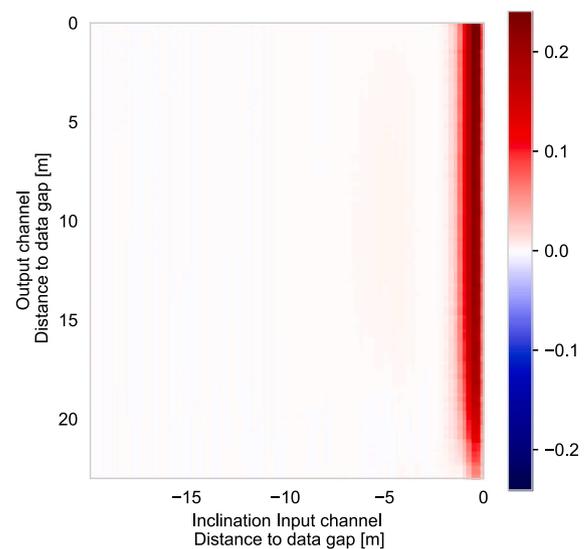


Fig. 25. Sobol indices heatmap of Inclination input channel and output. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

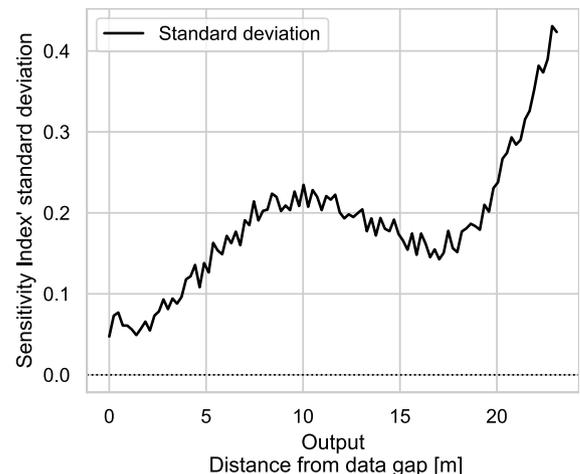


Fig. 26. Statistical sensitivity, Inclination channel, standard deviation.

4. Discussion

Approaching sensitivity analysis in the domain of neural networks, especially those with a high number of inputs, require a novel approach. This is especially visible in the case of recurrent neural networks that have highly interdependent inputs; it is where variance-based methods cannot be reliably used (Saltelli and Tarantola, 2002). Proposed data-driven approach exploits the fact that ML methods are created when there is a dataset available that is sufficiently big to cover the possible inputs and outputs.

While geometric arguments are brought up proving insufficiency of one at a time methods (Saltelli and Annoni, 2010) due to inability to cover the possible inputs, the proposed method does not suffer from this shortcoming. By performing analysis on a single input, as in the section on *Sensitivity to single inputs*, through repeating calculations on multiple starting points, the whole scope of potential inputs is covered. It is crucial to understand that the presented method deals with sensitivity of the model, not sensitivity of the real system. It is a significant distinction and expert interpretation is needed to decide whether a specific insight belongs to the system, or does it only exist in the model.

Additional benefit of data-driven sensitivity analysis is mathematical simplicity of the approach. It utilizes very basic methods and expands them through repetition. It becomes more and more common to bypass math-heavy approaches where analytical solutions exist with Monte Carlo type methods due to their simplicity (VanderPlas, 2016). Additionally, since the basic output of the presented method is a dataset, as opposed to a single value, further, deeper analysis is possible than presented here. For example, referring back to Fig. 14, median sensitivity of all outputs is the same, approximately 0.8. It is however possible to extract much more information, such as standard deviation or variance of the sensitivity. Result of such calculation is shown in Fig. 26, where standard deviation is shown for sensitivity on the inclination channel. This gives a very good indication of how different sensitivity can be between outputs, even though the median value is the same. Such calculations can be tailor made for each case, providing even more tools to analyze a model compared to common methods.

5. Conclusion

Classical sensitivity analysis methods are often ill-suited for data-driven models, such as recurrent neural networks, due to high number of interdependent inputs. Common, basic, one-at-a-time methods fail to provide meaningful results when high number of inputs is considered. These can, however, be enhanced by employing datasets originally used for training, validation, and testing the models. These datasets necessarily cover the possible inputs, often with statistical distribution matching reality.

In our case study, the proposed sensitivity analysis method was able to pinpoint many new insights into performance and limitations of the model that would otherwise be difficult to uncover. It was discovered that sensitivity exists where it should not. This suggests that the model

performs well when directional driller follows a certain procedure of operation, and the model can exploit correlations that are tied to that behavior. At the same time, the model would perform poorly if used as a simulator, where users are free to do as they please. We believe that the presented method may be a very useful tool in developing ever more complicated machine learning models, as well as, in evaluation of other black box solutions.

CRedit authorship contribution statement

Andrzej T. Tunkiel: Conceptualization, Investigation, Data curation, Methodology. **Dan Sui:** Formal analysis, Supervision, Writing - original draft. **Tomasz Wiktorski:** Supervision, Validation, Writing - original draft.

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.

Appendix

A.1. Elasticity index as input exponent

Derivation of Elasticity Index being the exponent of the input:

Consider Elasticity index of a function $y = f(x)$ as:

$$\mathcal{E}_{y,x} = \frac{dy}{dx} \cdot \frac{x}{y} = f'(x) \cdot \frac{x}{y} \quad (15)$$

A function in a general form $f(x) = C \cdot x^\alpha$ has derivative in relation to x as $f'(x) = \alpha \cdot C \cdot x^{\alpha-1}$. Therefore $\mathcal{E}_{y,x} = \alpha \cdot C \cdot x^{\alpha-1} \cdot \frac{x}{C \cdot x^\alpha}$. Since $x^{\alpha-1} = \frac{x^\alpha}{x}$:

$$\mathcal{E}_{y,x} = \alpha \cdot C \cdot \frac{x^\alpha}{x} \cdot \frac{x}{C \cdot x^\alpha} \quad (16)$$

which can be reduced to simply $\mathcal{E}_{y,x} = \alpha$. Bear in mind, that this is true for ΔS being significantly smaller than S_0 .

A.2. Additional figures

Some less important figures are reproduced in the [Appendix](#) only.

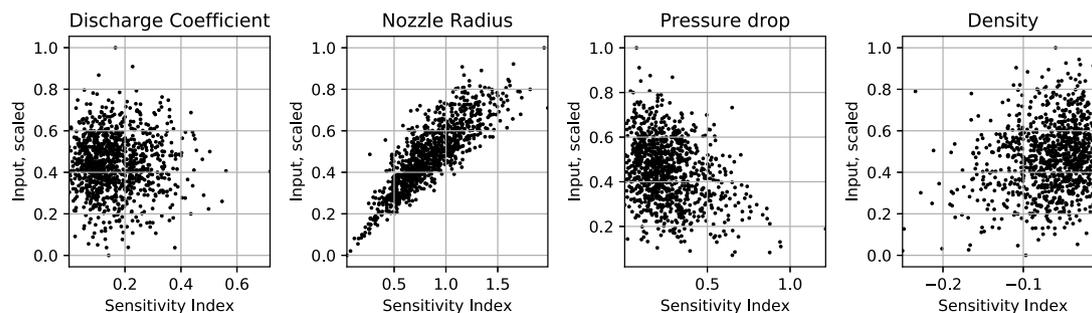


Fig. A.27. Sensitivity index, scaled function, scatterplot.

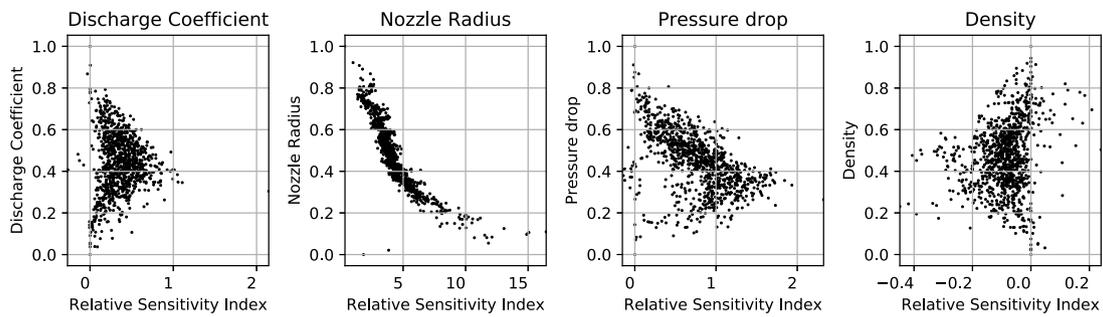


Fig. A.28. ML model, relative Sensitivity Index.

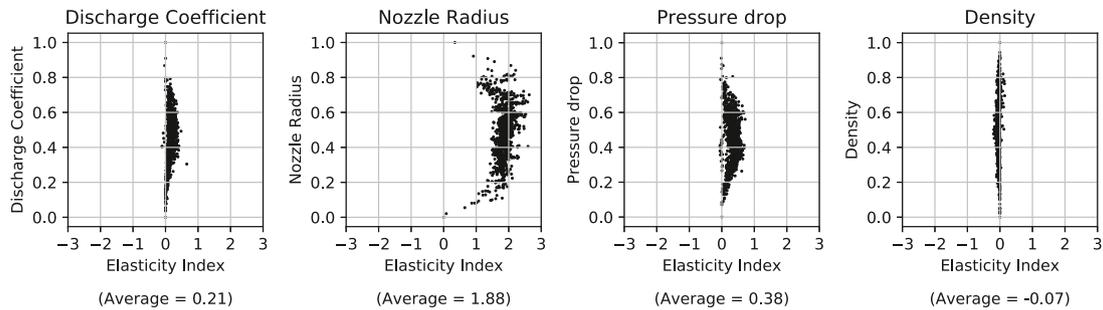


Fig. A.29. ML model, Elasticity Index.

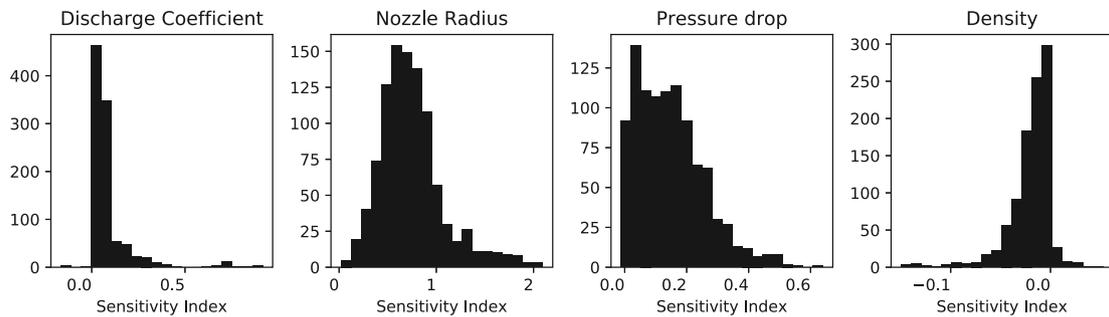


Fig. A.30. Uneven sample distribution, sensitivity index histogram.

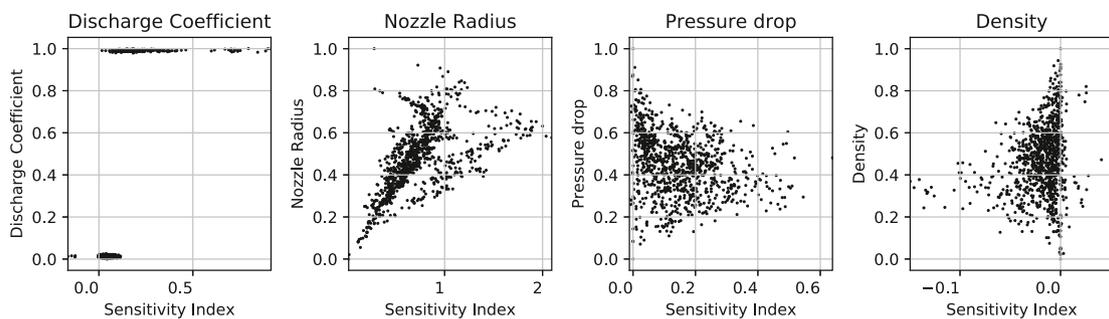


Fig. A.31. Uneven sample distribution, sensitivity index scatterplot.

References

Babakhani, P., Bridge, J., Doong, R.-a., Phenrat, T., 2017. Parameterization and prediction of nanoparticle transport in porous media: A reanalysis using artificial neural network. *Water Resour. Res.* 53 (6), 4564–4585. <http://dx.doi.org/10.1002/2016WR020358>, URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016WR020358>.

Breiman, L., 2001. Random forests. *Mach. Learn.* 45 (1), 5–32.

Caflich, R.E., 1998. Monte Carlo and quasi-Monte Carlo methods. *Acta Numer.* 7, 1–49. <http://dx.doi.org/10.1017/S0962492900002804>.

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. In: *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. pp. 1724–1734. <http://dx.doi.org/10.3115/v1/d14-1179>.

Detienne, K.B., Detienne, D.H., Joshi, S.A., 2003. Neural networks as statistical tools for business researchers. *Org. Res. Methods* 6 (2), 236–265. <http://dx.doi.org/10.1177/1094428103251907>.

Dutta, S., Gupta, J.P., 2010. PVT correlations for Indian crude using artificial neural networks. *J. Pet. Sci. Eng.* 72 (1–2), 93–109. <http://dx.doi.org/10.1016/j.petrol.2010.03.007>.

- Equinor, 2018. Volve field data (CC BY-NC-SA 4.0). URL <https://www.equinor.com/en/news/14jun2018-disclosing-volve-data.html>.
- Franceschini, S., Tancioni, L., Lorenzoni, M., Mattei, F., Scardi, M., 2019. An ecologically constrained procedure for sensitivity analysis of Artificial Neural Networks and other empirical models. *PLoS One* 14 (1), e0211445. <http://dx.doi.org/10.1371/journal.pone.0211445>, URL <https://pubmed.ncbi.nlm.nih.gov/30699204https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6353184/>.
- Herman, J., Usher, W., 2017. SALib: An open-source python library for sensitivity analysis. *J. Open Source Softw.* 2 (9), <http://dx.doi.org/10.21105/joss.00097>.
- Iooss, B., Lemaitre, P., 2015. A review on global sensitivity analysis methods. *Oper. Res./ Comput. Sci. Interfaces Ser.* 59, 101–122. http://dx.doi.org/10.1007/978-1-4899-7547-8_5.
- Jamal, A., Ali, R., Nouh, A., Hossam, F., 2014. Visualization analysis of feed forward neural network input contribution. *Sci. Res. Essays* 9 (14), 645–651. <http://dx.doi.org/10.5897/SRE2014.5895>, URL <http://academicjournals.org/journal/SRE/article-abstract/F21242846447>.
- Kurowicka, D., Cooke, R., 2006. Uncertainty analysis with high dimensional dependence modelling. In: *Uncertainty Analysis with High Dimensional Dependence Modelling*. Wiley Series in Probability and Statistics. <http://dx.doi.org/10.1002/0470863072>.
- Loucks, D.P., van Beek, E., Stedinger, J.R., Dijkman, J.P., Villars, M.T., 2005. 9 model sensitivity and uncertainty analysis. In: *Water Resources Systems Planning and Management: An Introduction To Methods, Models and Applications*. ISBN: 92-3-103998-9, pp. 254–290, URL https://www.utwente.nl/ctw/wem/education/afstuderer/Loucks_VanBeek/09_chapter09.pdf.
- Lu, M., AbouRizk, S.M., Hermann, U.H., 2001. Sensitivity analysis of neural networks in spool fabrication productivity studies. *J. Comput. Civ. Eng.* 15 (4), 299–308. [http://dx.doi.org/10.1061/\(ASCE\)0887-3801\(2001\)15:4\(299\)](http://dx.doi.org/10.1061/(ASCE)0887-3801(2001)15:4(299)), URL <http://ascelibrary.org/doi/10.1061/>.
- Nourani, V., Sayyah Fard, M., 2012. Sensitivity analysis of the artificial neural network outputs in simulation of the evaporation process at different climatologic regimes. *Adv. Eng. Softw.* <http://dx.doi.org/10.1016/j.advengsoft.2011.12.014>.
- Radaideh, M.I., Surani, S., O'Grady, D., Kozlowski, T., 2019. Shapley effect application for variance-based sensitivity analysis of the few-group cross-sections. *Ann. Nucl. Energy* 129, 264–279. <http://dx.doi.org/10.1016/j.anucene.2019.02.002>.
- Saltelli, A., 2002. Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Comm.* [http://dx.doi.org/10.1016/S0010-4655\(02\)00280-1](http://dx.doi.org/10.1016/S0010-4655(02)00280-1).
- Saltelli, A., Annoni, P., 2010. How to avoid a perfunctory sensitivity analysis. *Environ. Modell. Softw.* 25 (12), 1508–1517. <http://dx.doi.org/10.1016/j.envsoft.2010.04.012>.
- Saltelli, A., Tarantola, S., 2002. On the relative importance of input factors in mathematical models. *J. Amer. Statist. Assoc.* 97 (459), 702–709. <http://dx.doi.org/10.1198/016214502388618447>.
- Shojaeefard, M.H., Akbari, M., Tahani, M., Farhani, F., 2013. Sensitivity analysis of the artificial neural network outputs in friction stir lap joining of aluminum to brass. In: Vilar, R. (Ed.), *Adv. Mater. Sci. Eng.* 2013, 574914. <http://dx.doi.org/10.1155/2013/574914>.
- Simpson, T.W., Peplinski, J.D., Koch, P.N., Allen, J.K., 2001. Metamodels for computer-based engineering design: Survey and recommendations. *Eng. Comput.* 17 (2), 129–150. <http://dx.doi.org/10.1007/PL00007198>.
- Sobol, I.M., 1993. Sensitivity estimates for nonlinear mathematical models. *Math. Modell. Comput. Exp.* 1 (4), 407–414.
- Sobol, I.M., 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simulation* 55 (1–3), 271–280. [http://dx.doi.org/10.1016/S0378-4754\(00\)00270-6](http://dx.doi.org/10.1016/S0378-4754(00)00270-6).
- Sorgun, M., Ulker, E., 2016. Modeling and experimental study of solid-liquid two-phase pressure drop in horizontal wellbores with pipe rotation. *J. Energy Resour. Technol. Trans. ASME* 138 (2), <http://dx.doi.org/10.1115/1.4031743>.
- Sydsaeter, K., Hammond, P.J., 1995. *Mathematics for Economic Analysis*. (HB135 S98).
- Tissot, J.Y., Prieur, C., 2012. Bias correction for the estimation of sensitivity indices based on random balance designs. *Reliab. Eng. Syst. Saf.* 107, 205–213. <http://dx.doi.org/10.1016/j.res.2012.06.010>.
- Tunkiel, A.T., Github for OMAE2020-18154, URL <https://github.com/AndrzejTunkiel/OMAE2020-18154>.
- Tunkiel, A.T., Wiktorski, T., Sui, D., 2020. Continuous drilling sensor data reconstruction and prediction via recurrent neural networks. In: *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering, OMAE* (in preparation).
- Ulker, E., Sorgun, M., 2016. Comparison of computational intelligence models for cuttings transport in horizontal and deviated wells. *J. Pet. Sci. Eng.* 146, 832–837. <http://dx.doi.org/10.1016/j.petrol.2016.07.022>.
- VanderPlas, J., 2016. *Statistics for Hackers*. YouTube, URL <https://www.youtube.com/watch?v=Iq9DzN6mvYA>.
- Yu, Y., Si, X., Hu, C., Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* 31 (7), 1235–1270. http://dx.doi.org/10.1162/neco_a_01199.