NAM

Evaluation of a Machine Learning methodology to forecast induced seismicity event rates within the Groningen Field

Joint team from IBM and Shell

J. Limbeck, F. Lanz, E. Barbaro, C. Harris, K. Bisdom, T. Park, W. Oosterbosch, H. Jamali-Rad and K. Nevenzeel

Datum August 2018

Editors Jan van Elk, Taco den Bezemer & Dirk Doornhof

General Introduction

Several models have been developed for the forecasting of seismicity induced by the production of gas from the Groningen field. In 2013, a strain-partitioning seismological model was presented in the technical addendum to the Winningsplan 2013 (Ref. 1). This model is further described in a scientific peer-reviewed paper titled "A seismological model for earthquakes induced by fluid extraction from a subsurface reservoir", published in the Journal of Geophysical Research (Ref. 2). An alternative seismological model, an activity rate model, was developed in 2015 (Ref. 3, 4 and 5).

Encouraged to investigate alternative seismological models a set of models with increasing complexity was developed. These geomechanical models were calibrated using the historical earthquake record data of Groningen, in combination with the measured subsidence. Using prospective testing the best performing model was chosen for incorporation in the model for the hazard and risk assessment (Ref. 6 and 7).

Several other seismological models have been developed using different approaches. For instance, 3D-models, including a large number of individual faults mapped in the Groningen field have been developed (Ref. 1 and 8). These models are very complex and required large run-times making incorporation of these in the Monte Carlo approach for the hazard and risk assessment practically unfeasible.

In preparation of the workshop on the maximum magnitude of earthquakes in Groningen, another alternative seismological model was prepared by Suckale and Dempsey (Ref. 9).

Several other approaches that could be used to develop alternative seismological models have been reviewed. Machine Learning was seen as a not previously tested approach, with potential to deliver a seismological model for short term forecasting. The current report describes the effort to develop such a model using Machine Learning.

References

1. Technical Addendum to the Winningsplan Groningen 2013; Subsidence, Induced Earthquakes and Seismic Hazard Analysis in the Groningen Field, Nederlandse Aardolie Maatschappij BV (Jan van Elk and Dirk Doornhof, eds), November 2013.
2. A seismological model for earthquakes induced by fluid extraction from a subsurface reservoir, S. J. Bourne, S. J. Oates, J. van Elk, and D. Doornhof, Journal of Geophysical Research: Solid Earth, 119, 8991-9015.
3. An activity rate model of induced seismicity within the Groningen Field, (Part 1), Stephen Bourne and Steve Oates, February 2015.
4. An activity rate model of induced seismicity within the Groningen Field, (Part 2), Stephen Bourne and Steve Oates, June 2015.
5. Computing the Distribution of Pareto Sums using Laplace Transformation and Stehfest Inversion Break, C. K. Harris and S. J. Bourne, May 2015.
6. Extreme threshold failures within a heterogeneous elastic thin-sheet and the spatial-temporal development of induced seismicity within the Groningen gas field, S. J. Bourne, S. J. Oates,
7. The exponential rise of induced seismicity with increasing stress levels in the Groningen gas field and its implications for controlling seismic risk, S.J. Bourne, S.J. Oates and J. van Elk, Geophys. J. Int. (2018) 213, 1693–1700.
8. Groningen 2015 Geomechanical Analysis, Suvrat P. Lele, Jorge L. Garzon, Sheng-Yuan Hsu, Nora L. DeDontney, Kevin H. Searles and Pablo F. Sanz (ExxonMobil Upstream Research Company, Spring, TX), March 2016.
9. Dempsey, D., & Suckale, J. (2017). Physic-based forecasting of induced seismicity at Groningen gas field, the Netherlands. Geophysical Research Letters, 44(15), 7773-7782.

NAM

Title	**Evaluation of a Machine Learning methodology to forecast induced seismicity event rates within the Groningen Field**		**Date**	August 2018
			Initiator	NAM
Author(s)	J. Limbeck, F. Lanz, E. Barbaro, C. Harris, K. Bisdom, T. Park, W. Oosterbosch, H. Jamali-Rad and K. Nevenzeel	**Editor**	Jan van Elk, Taco den Bezemer and Dirk Doornhof	
Organisation	IBM and Shell Global Solution	**Organisation**	NAM	
Place in the Study and Data Acquisition Plan	Study Theme: Seismological Modelling Comment: Several models have been developed for the forecasting of seismicity induced by the production of gas from the Groningen field. In 2013, a strain-partitioning seismological model was presented in the technical addendum to the Winningsplan 2013. This model is further described in a scientific peer-reviewed paper titled "A seismological model for earthquakes induced by fluid extraction from a subsurface reservoir", published in the Journal of Geophysical Research. An alternative seismological model, an activity rate model, was developed in 2015. Encouraged to investigate alternative seismological models a set of models with increasing complexity was developed. These geomechanical models were calibrated using the historical earthquake record data of Groningen, in combination with the measured subsidence. Using prospective testing the best performing model was chosen for incorporation in the model for the hazard and risk assessment. Several other seismological models have been developed using different approaches. For instance, 3D-models, including a large number of individual faults mapped in the Groningen field have been developed. These models are very complex and required large run-times making incorporation of these in the Monte Carlo approach for the hazard and risk assessment practically unfeasible. In preparation of the workshop on the maximum magnitude of earthquakes in Groningen, another alternative seismological model was prepared by Suckale and Dempsey. Several other approaches that could be used to develop alternative seismological models have been reviewed. Machine Learning was seen as a not previously tested approach,			

	with potential to deliver a seismological model for short term forecasting. The current report describes the effort to develop such a model using Machine Learning.
Directly linked research	1. Reservoir engineering studies in the pressure depletion for different production scenarios. 2. Seismic monitoring activities; both the extension of the geophone network and the installation on geophones in deep wells. 3. Geomechanical studies 4. Subsidence and compaction studies.
Used data	Gas production data and reservoir pressure data KNMI Earthquake catalogue Subsidence and compaction data Geological maps of faults in the Rotliegend reservoir
Associated organisation	
Assurance	

Evaluation of a Machine Learning methodology to forecast induced seismicity event rates within the Groningen Field

by

J. Limbeck (GSNL-PTX/D/S)

F. Lanz (IBM Services)

E. Barbaro (IBM Services)

C. Harris (SUKEP-UPO/W/T)

K. Bisdom (GSNL-PTX/S/RM)

T. Park (GSNL-PTX/D/S)

W. Oosterbosch (IBM Services)

H. Jamali-Rad (GSNL-PTX/S/IA)

K. Nevenzeel (IBM Services)

1 Executive Summary

Business purpose:

Decades of gas production caused induced seismicity in the Groningen gas field, located in the Northern part of the Netherlands. The capability to forecast induced seismicity depending on production strategy is an essential element of the Probabilistic Seismic Hazard and Risk Assessment (PSHRA) for the impacted population. As part of the Study and Data Acquisition Plan in the context of the Measure and Control Protocol, this study evaluates a methodology based on machine learning (a branch of artificial intelligence in the field of computer science) to forecast production induced seismicity event rates for the Groningen Field. The methodology allows probing of a wide variety of possible linear and non-linear combinations and interaction terms of potential predictor variables (features), without assuming a priori knowledge on the nature of the relationships between the features. The features are selected based on domain expert advice and literature.

Approach:

A two-step approach is employed: first, a factorial experimental setup followed by meta analysis (analysis of the effectiveness of the experimental setup) is used to select robust and relatively well performing models and meta parameters. The factorial experimental design covers a large parameter space of ~4 million experiments in an exploratory phase and ~175 thousand experiments are analysed in more detail. Main experimental design parameters include the target (choices regarding seismicity event rate quantification, in particular minimum magnitude and starting time), the machine learning model used and time delays between potentially predicting physical variables and seismicity. From this large set of experiments, meta parameter values and machine learning models are selected based on three criteria: forecast performance, a minimum R^2 explanatory power threshold and stability under small changes in meta parameters. Second, the selected models and meta parameters are used for seismicity event rate forecasts.

Methodology evaluation:

- *The range of validity* of the methodology described are future production scenarios that are similar to past production behaviour. In particular, for the standard production scenario of the Production Plan 2016 the methodology generates forecasts similar to the default PSHRA forecasts, but it does not provide physically realistic forecasts for the average production scenario announced by the Ministry of Economic Affairs and Climate in March 2018.
- *Key improvements* which might extend the range of validity include:
  - Investigate usage of longer term (1-5 years) forecasts to validate model performance, instead of the short term (1-3 months) forecast performance evaluations used in this study. The latter have the advantage of maximizing statistical power to distinguish forecast performance of various models but require the assumption that short term performance is also indicative for long term performance. That assumption is not always satisfied and may lead to selecting models which perform well on the short term but not on the long term.
  - For non-extrapolating models, the feature set could be limited to features whose future values won't exceed beyond the convex hull of past feature values. In particular, this would exclude monotonically evolving features like the reservoir pressure and cumulative compaction for these models.

  - The automated model evaluation and selection criteria are mathematical criteria: forecast error, variance explained and forecast robustness. Extension of this criteria set with rules encoding physics based limitations would enable automated exclusion of unphysical forecasts.
- *Definiteness of conclusions* are pending validation on a hold-out set, as all data has been used for model meta analysis (and thus model selection). Consequently, model performance estimates of the selected models are possibly on the optimistic side and a hold-out set is required to validate these estimates. Two approaches would be available. First, a hold-out set will naturally be obtained over time. Second, training/testing up to 2012 and using the remaining data set as hold-out might enable validation as well. The first choice will require several years of additional data collection, the second choice might decrease the statistical power to distinguish between the performance of various models.

The authors advise that pending an increased range of validity and more definite conclusions on forecast performance the models should not be used for business decisions.

Next steps:

To further improve machine learning based seismicity forecasts for the Groningen field and to follow up on the leads from this study three suggestions are presented:

- Extend the range of validity of the methodology and the definiteness of conclusions by progressing the suggestions mentioned above.
- Investigate the forecast performance gain which hybrid models combining physics and machine learning models could provide.
- Extend the event rate methodology developed in this study to include areal and magnitude forecast capabilities.

Table of Contents

Table of Figures

Table of Tables

2 Introduction: Overview, Earlier Work & Study Goals

Discovered in 1959 with an initial recoverable reserve estimate of 2900 billion m^3 gas, the Groningen gas field is amongst the largest gas fields in the world (TNO, Geology Service Netherlands, sd). Production commenced by NAM in 1963, by 2015 around 2000 billion m^3 have been produced. The reservoir of the Groningen field is the Upper Rotliegend Group of Early Permian age, consisting of porous sandstone and located at a depth between 2600m and 3200m, with the water zone around 3000m deep. The gas in the reservoir is sealed by a thick impermeable salt and anhydrite layer of the overlying Zechstein Group, as depicted in Figure 1. The Groningen field has several fault systems with around 1500 known faults, whose existence doesn't impact permeability in a significant way.

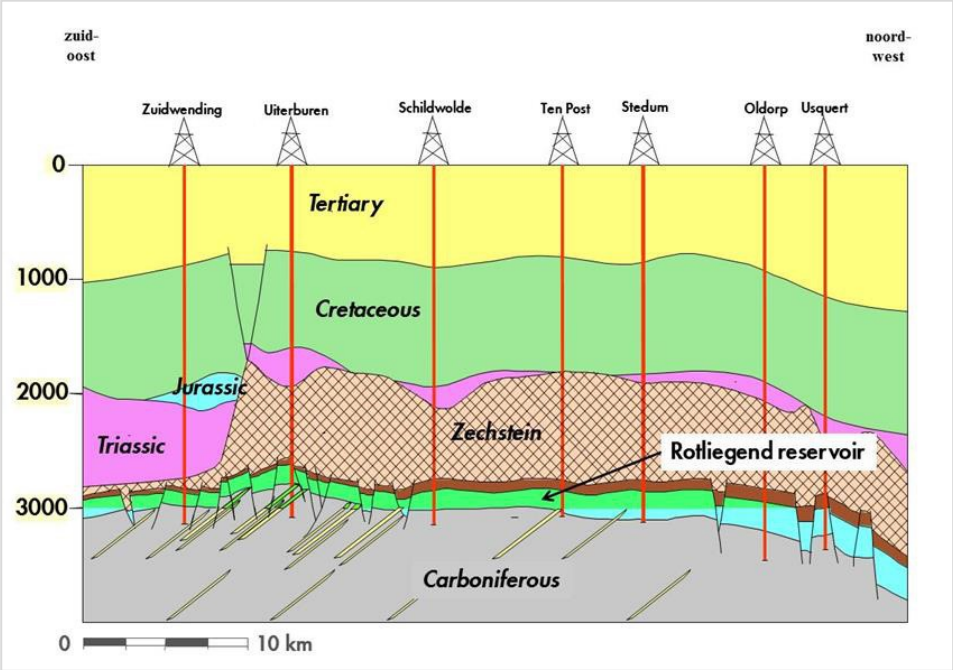


Figure 1: Geological cross-section of the Groningen Field (NAM, 2016)

Following decades of gas production, the historically aseismic region experienced induced earthquakes for the first time in 1991. The frequency and intensity of earthquakes increased steadily to around ten or more earthquakes pear year with a magnitude equal or larger than 1.5 as of 2003, see Figure 51 in Appendix 1. Following an earthquake of magnitude 3.6 on the Richter scale with an epicenter in the village of Huizinge in 2012, a Study and Data Acquisition Plan (NAM, 2016) was put in place to better understand how gas production at reservoir depth affects safety at the surface, and to test the effectiveness of mitigation measures. This led to an integrated Probabilistic Seismic Hazard and Risk Assessment (PSHRA) starting from gas production, sequentially followed by compaction, seismicity, ground motion, exposure, building strengthening and finally risk and safety of inhabitants, see Figure 2.

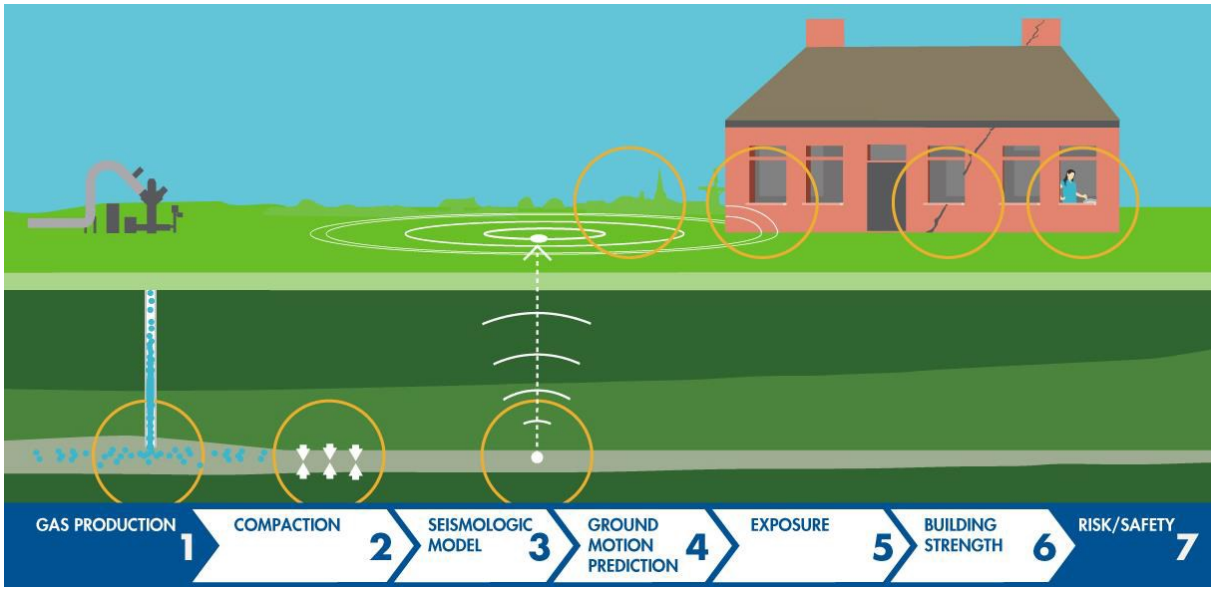


Figure 2: Causal chain from gas production to safety of people in or near a building (NAM, 2016)

PSHRA is realized via an extensive Study and Data Acquisition Plan which encompasses this study in the context of the Measure and Control Protocol (NAM, 2017). The focus of this study is seismologic modelling (element 3 of PSHRA) and forecasting.

Here, we explicitly use the word "forecast" instead of "prediction" as within seismology both terms refer to different approaches to gain more quantitative insights in future seismicity (Marzocchi & Zechar, 2011). Predictions refer to high confidence statements about the location, timing and magnitude of a future seismic event, whereas forecasts are used to describe quantitative statements about future event statistics. This study and all other studies to date which have been able to provide reliable statements about future seismicity in the Groningen field are forecasts – due to limitations in both available data and human understanding of geophysical mechanics triggering earthquakes. To avoid any confusion between audiences versed in different scientific fields, we observe that within the field of machine learning the term "prediction" is applied in a broader sense than in seismology – hence, to align the machine learning oriented expositions below with common practice in that community, in these sections we sometimes use the word "prediction".

A large body of literature on seismicity in Groningen already exists. Section 0 provides a literature overview of physical seismicity analysis and forecasting, including the default statistical physics PSHRA forecasting model and more deterministic physics modelling approaches. Section 2.2 elaborates on the statistical insights gained in literature so far, including the effects of shut-in, the possible existence of seasonality and correlations between seismicity and several physical quantities. To the best of our knowledge machine learning has not been used for seismicity forecasting in Groningen yet but it has been used in a variety of similar situations, a literature overview is provided in section 2.3. Building on all these studies, the approach taken in this study in described in section 2.4.

2.1 Physical Analysis and Forecasts for Seismicity in Groningen

The conceptual sequence of events leading to seismicity is illustrated in Figure 3: gas production Q results in reservoir pore pressure reduction ΔP, which both via reservoir compaction C and directly leads to fault stress changes, in turn causing seismicity. Compaction also results in surface subsidence S.

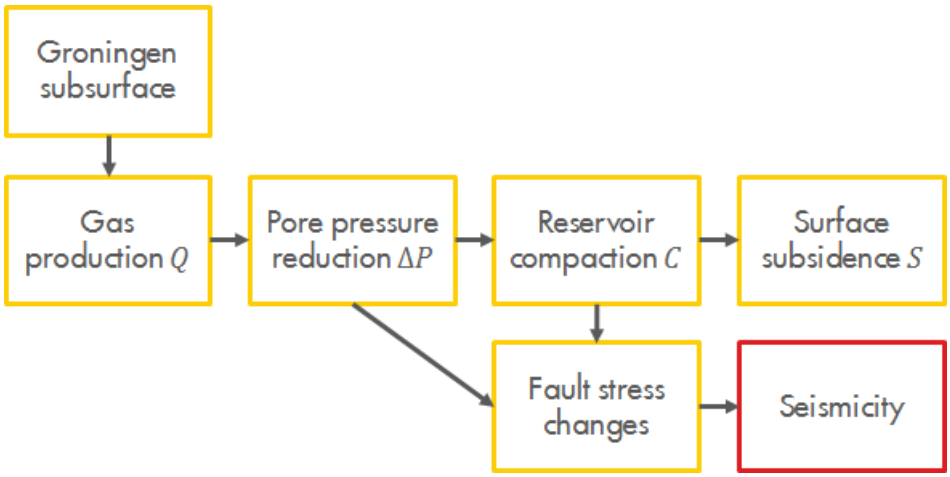


Figure 3: Conceptual sequence of events from gas production to seismicity. Adapted from (Van Thienen-Visser, Sijacic, Van Wees, Kraaijpoel, & Roholl, 2016).

For Groningen several statistical physical seismicity forecast models have been developed or are under development. The standard PSHRA model (Bourne & Oates, Development of statistical geomechanical models for forecasting seismicity induced by gas production from the Groningen field, 2017) is based on the empirical spatial-temporal magnitude evolution of induced seismicity, giving an exponential relation between seismicity and incremental Coulomb stress. It consists of three core elements (Bourne & Oates, Extreme Threshold Failures Within a Heterogeneous Elastic Thin Sheet and the Spatial-Temporal Development of Induced Seismicity Within the Groningen Gas Field, 2017), (Bourne & Oates, An activity rate model of induced seismicity within the Groningen Field (part 1), 2015), (Bourne & Oates, An activity rate model of induced seismicity within the Groningen Field (Part 2), 2015):

1. Pore pressure depletion resulting in reservoir compaction and in turn to changes in fault stresses is modelled via an elastic thin sheet approximation of the reservoir.
2. When fault stress changes lead to failure (and thus seismicity) depends on the sum of the fault stress changes and the unknown (and unobservable) initial distribution of stresses: higher initial stresses fail faster. These initial stresses of the fault network are approximated via Extreme Value Theory, which suggests an exponential relation for the distributional tail irrespective of the full (unknown) distribution.
3. The geospatial and temporal seismicity statistics are obtained via a heterogeneous Poisson Point Process.

These three elements combined provide estimates for the temporal, geospatial and magnitude distribution of seismicity. The foundations for this model were already visible in the previous PSHRA standard model (Bourne, Oates, Van Elk, & Doornhof, 2014), which describes the link between seismic moment and increment in reservoir strain. An alternative model based on a similar approach is proposed by (Dempsey & Suckale, 2017), who amongst others numerically calculate fault failure for 1D fault models of the largest 325 faults in the reservoir instead of deriving fault failure probabilities using Extreme Value Theory. It should be noted that also much smaller faults than the largest 325 faults are capable of generating earthquakes of the observed magnitudes and

even larger ones (Bourne & Oates, An activity rate model of induced seismicity within the Groningen Field (part 1), 2015).

As seismicity occurs at faults – planes of weakness in the field – a different approach towards seismological modelling is taken by more deterministic physics models. Such models aim to explain seismicity starting at individual faults and employ simplified geomechanical analytical finite element and rupture models (Van Wees, et al., 2017). A 3D finite element model was developed by (Lele, et al., 2015) and two other finite element models by (Van Wees, Osinga, Van Thienen-Visser, & Fokker, 2018) and (Postma & Jansen, 2017). The latter two specifically analysed whether abrupt production changes due to for example well shut-ins impact seismicity. Both studies concluded that production shut-ins do decrease seismicity on the short term but neither study is straightforward extendible to generate long term seismicity forecasts for the Groningen field.

(Van den Bogert P. A., 2015) developed a 2D rupture model that provided several physical insights in seismicity (e.g. that seismic ruptures occur with the lowest depletion levels where the normalized reservoir offset[1] across a fault equals 1, the identification of three rupture type classes, that moment magnitude of seismic events is influenced by the difference between initial and residual fraction coefficient of the fault as well as the normalized reservoir offset, …). Additionally, synthetic waveforms are generated by dynamic rupture simulations which might be calibrated against observed waveforms and as such ideally can be used to identify fault properties. For artificial faults numerical simulations with Groningen field like conditions are being conducted, see e.g. (Buijze, Van den Bogert, Wassing, Orlic, & Ten Veen, 2017). These simulations successfully reproduce fault failure and seismic wave generation but scaling from an artificial to real faults remains a major challenge. Furthermore, work is in progress to extend the 2D rupture model to a 3D rupture model (Van den Bogert & Yuan, 2017).

2.2 Statistical Analysis and Forecasts for Seismicity in Groningen

Next to the physics based analysis of induced seismicity a sizable body of literature developed approaching the problem from a largely statistical perspective, including development of a statistical forecast regression model and analysis of correlations between key physical quantities and seismicity. Seismicity event rates were analysed statistically to investigate for instance seasonality, shut-in effects and potential seismic epochs – aspects which provided e.g. insights on possible non-linear effects. In the below we provide a high-level overview.

A statistical forecast regression model based on the empirical relation between the cumulative number of events and the cumulative gas production was developed by (Hettema, Jaarsma, Schroot, & Van Yperen, 2017), who relate the ratio of activity rate over production rate versus the cumulative production. A limited forecast suggests that for the given forecast period the observed seismicity is in the 95% confidence interval of this predictive model. Their paper discussed that potential time delay between production changes and changes in physical variables down the line probably increases with decreasing reservoir pressure, but these effects are not included in the model.

Analysis of (Pijpers, Trend changes in tremor rates Groningen - update Nov. 2016, 2016) and references to earlier updates therein suggests that it is unlikely that the tremor rates are determined

[1] Normalized reservoir offset if the depth difference between the reservoir at both sides of a fault, measured in units of "reservoir depth". An offset of 0 implies that the reservoir has the same depth on both sides of the fault, whereas an offset of e.g. 1 implies that the reservoir on one side of the fault is 1 reservoir depths lower than the reservoir at the other side of the fault.

purely by a frame rate effect[2]. From previous updates a delay between Q and seismicity of around 3 months was found, this study finds an optimal delay time of 12 months between ΔQ and seismicity. (Nepveu, Van Thienen-Visser, & Sijacic, 2016) and (Van Thienen-Visser, et al., 2015) consider the cross-correlation between (seasonal) production changes and seismicity as well as the change in seismicity. They find a correlation between both – with an optimal delay of 2-8 months and 2 months. A clear anticorrelation between reservoir gas pressure P decrease and seismicity rates is found by (Pijpers, Interim report: correlations between reservoir pressure and earthquake rate, 2017), with correlation coefficient ρ more negative than -0.6 for half of the regions investigated. A time delay between 5 to 10 weeks between reservoir pressure changes and seismicity rates is reported by (Pijpers, A phenomenological relationship between reservoir pressure and tremor rates in Groningen, 2016). The relation between subsidence and seismicity was investigated by (Pijpers & Van der Laan, Trend changes in ground subsidence in Groningen - update November 2015, 2016), who found that changes in production result in changes in subsidence, typically after 9 weeks.

A recent study investigating seasonality in event rate time series is (Bierman S. , Seasonal variation in rates of earthquake occurrences in the Groningen field, 2017), who use Schuster's spectrum method to test for a range of periodicities (such as daily and monthly). They concluded that strong evidence for seasonality exists for earthquakes with $M \leq 1.0$, some for $1.0 \leq M < 1.5$ and none when $M \geq 1.5$. Furthermore, if only data post January 2014 is used no sign of seasonality remains regardless of magnitude. An analogous conclusion on the magnitude dependence of seasonality was reached in their earlier work, see (Bierman, Paleja, & Jones, 2015) and (Bierman, Paleja, & Jones, 2016). A delay of 3-4 months between seasonal production and seasonal seismicity is found.

(Nepveu, Van Thienen-Visser, & Sijacic, 2016) and (Van Thienen-Visser, et al., 2015) also address in detail the question whether there are any change points in seismicity due to e.g. production measures. Two change points in seismicity are found: one in early 2003 and possibly one in early 2014 – the first can be confirmed with statistical significance. The second one is of most direct interest as it might relate to the production measures taken around that time, but for statistical confirmation more data would be required than what was available at the time of the study. A different approach to the same question is taken by (Paleja & Bierman, 2016), who analyse changes to inter-event rate pre- and post-shut-in. This is done by counting the events in the post-period and then dividing the pre-period in groups with an equal earthquake count as the post-period, effectively creating groups that show how much time it took for a certain number of earthquakes to occur. The study concludes that there is evidence of a further decrease in the inter event rate for the Loppersum and North region of the field, while for the South West region the activity rate has increased in the post "shut down" period.

2.3 Machine Learning Seismicity Forecasts Elsewhere

Machine Learning (ML) is a branch of statistical computer science which over the last decade has been applied successfully in a wide variety of domains (Jordan & Mitchell, 2015). In the context of physics, machine learning allows for experimental control over a vast number of factors (Langley, 1988) making it suitable for physical modelling (Liu, 2018). Due to their nature, ML models tend to perform well in situations where underlying processes are not fully understood (Melnikov, 2018) or are complex (Carrasquilla & Melko, 2017). Machine learning has proven to accurately predict the behaviour of large spatiotemporal chaotic physical system where the mechanical description of

[2] If seismicity follows a frame rate, the total number of earthquakes only depends on the total gas production and not on the rate of production.

the dynamics is limited (Pathak, Hunt, Girvan, Zhixin, & Ott, 2018) – accurate predictions up to eight times the regular prediction horizon could be achieved. In view of that, machine learning seems a viable tool to complement physical and statistical seismicity modelling efforts and has become increasingly popular for seismic analysis. Three main ways in which machine learning has been applied within seismicity studies are (i) earthquake identification, (ii) catalogue based seismicity forecasting and (iii) model parameter inference (e.g. the Gutenberg-Richter b-value). On top, we note that in the context of PSHRA machine learning is used already for optimisation of the production distribution over the Groningen field to reduce Seismicity, see (NAM, 2017). A non-exhaustive review aimed to shed light on the role of ML in seismic analysis is given below. The reader unfamiliar with some of the algorithms mentioned is referred to chapter 6 for a high-level overview.

Earthquake identification is often done by acoustic or ground vibration wavelet analysis of seismic detection sensors. A recent study of (Rouet-Leduc, et al., 2017) predicted time to fault failure based on a local moving time window signal emitted by laboratory faults. In their study, a wide number of potential predictors was computed for every single time window (e.g. $0^{th}/1^{st}/2^{nd}$ order statistics) and the most useful features are used in a Random Forest model achieving a high determination coefficient ($R^2 = 0.89$). Interestingly, the Random Forest model accurately predicted failure not only when failure is imminent but throughout the failure cycle. Features which quantify signal amplitude distribution (e.g. variance and higher-order moments) are highly effective predictors, despite their high variability. The authors acknowledge that this effort remains academic, however. We note that if a connection between seismic wavelets and fault properties could be identified, it would help development of deterministic geomechanical models. (Perol, Gharbi, & Denolle, 2017) employed a scalable Neural Network to consistently detect and localize earthquakes based on a single waveform. They claim to detect 20 times more earthquakes as previous earthquake catalogues, which is important to make seismic catalogues more complete, in turn improving Hazard and Risk Assessments for induced seismicity in Oklahoma. A possible caveat of this study is the fact that it requires pre-existing history of catalogued seismicity and is therefore less suitable for areas of lower activity or more recent instrumentation. (Ramirez Jr., 2011) used a kernel ridge-regression algorithm to study seismic phases from seismic recordings. Their method consists of a multi-scale potential predictor extraction on low-dimensional manifolds. In addition, they merged their regression scores across the potential predictor manifolds. The authors concluded that their algorithm could correctly predict around 75% of the classification rates for seismic data collected in the US during 2005 and 2006.

Seismicity forecasting via earthquake catalogues uses dates, locations and magnitudes of earthquakes to forecast future earthquakes. (Panakkat & Adeli, 2009) forecast earthquake times and locations for earthquakes for magnitude $M \geq 4.5$ using a wide variety of Neural Networks. These networks were offered multiple seismicity indicators derived from an earthquake catalogue (e.g. Gutenberg-Richter's b-value, the average magnitude of the last n events, the mean-square deviation about the regression line based on Gutenberg-Richter's inverse power law curve for n events, etc.) as parameters. The magnitude of their error in forecasting the epicentral location of high magnitude events was always within 20-40 miles, which the authors claim to be useful for emergency management and planning. (Rouet-Leduc, et al., 2017) utilized a random forest algorithm on lab-induced earthquakes to investigate hidden signals preceding the events. They suggest that previous literature only based on earthquake catalogues may be incomplete.

(Asencio Cortez, Martinez-Alvarez, Morales-Esteban, & Reyes, 2016) proposed a meta-analysis setup to find out the best set of parameters and concluded that it is possible to use ML techniques to calculate the b-value. (Last, Rabinowitz, & Leonard, 2016) focused on understanding whether future maximum earthquake magnitude exceed the median of maximum yearly magnitudes (for the

same region). Several ML algorithms used here are also utilized in their study (Decision Trees, K Nearest Neighbours, Support Vector Machines and Neural Networks). Their results point out to a variant of a decision tree as the most accurate machine learning model. Their features are based on observed earthquake catalogues and derived relations, e.g. the Gutenberg-Richter law.

2.4 This Study: Machine Learning Seismicity Forecasts for Groningen

To the best of our knowledge, machine learning based seismicity forecasts have not yet been developed for Groningen. In light of the successful application of machine learning to various intricate physical problems, the main goal of this study is to develop a methodology for machine learning based induced seismicity event rate forecasts for the Groningen Field. Furthermore, so far machine learning seismicity forecasts for other areas seem to be based on event rates only whereas from the physical and statistical work mentioned above we know that for the Groningen case physical quantities like compaction C, reservoir pressure P and production Q carry significant information on induced seismicity as well. We use these quantities within our framework to obtain an as high as possible predictive performance. Key advantages of machine learning based seismicity forecast methodology developed in this study are:

- Most underlying models do not depend on a predetermined functional relationship and can capture a wide variety of possible linear and non-linear combinations and interaction effects;
- A factorial setup allows probing a large parameter space of plausible modelling assumptions and meta-parameter choices, followed by statistical meta-analysis to ensure statistical significance and robustness of results;
- Semi-automated implementation enables straightforward testing of hypotheses whether a specific data source or variable increases predictive performance or not.

Main limitations of the current setup include:

- This study only concerns event rate forecasts – a full seismicity model for Groningen should additionally be able to forecast event locations and magnitudes.
- The range of validity of the methodology reported here is limited by three key aspects: (i) the assumption that a short term (1-3 months) forecast performance is also indicative for long term (1-5 years) forecast performance; (ii) non-extrapolating model forecasts are constrained to the range of historically observed feature values; (iii) the purely mathematical nature of the model evaluation and selection rules, which do not encode (high level generally agreed upon) physics based boundary conditions.
- Due to the overall small data set size, it was considered unfeasible at the time of writing to reserve data for an additional hold-out set. In absence of such a set, performance estimates of the model(s) should be seen as tentative and have to be verified with a hold-out set at a later point in time.
- Pending an increased range of validity and more definite conclusions on forecast performance the models reported on here should not be used for business decisions.
- Contrary to most physics or linear regression models, several machine learning models are black boxes and are not straightforwardly interpretable (Breiman, Statistical Modeling: The Two Cultures, 2001).

Following our methodology as shown high-level in Figure 4, this study is structured as follows:

- **Data sources** are selected and potential predictors (features) are generated from these data sources in chapter 3. Given the physics and statistics work described in previous sections, the following data sources are incorporated at least: earthquakes, production, reservoir pressure, faults, compaction and subsidence.
- **Meta-parameters** define the experimental setup within which models are trained and do forecasts – they are defined in chapter 4. Our meta-parameters can be divided in two sets: (i) those related to our prediction target like minimum magnitude, which are discussed in section 3.2; (ii) those describing our experimental setup, like potential time delays mentioned in several earlier studies, which are described in chapter 4.
- **The model evaluation strategy** is developed in chapter 5. In summary, we use a walk-forward evaluation strategy with two standard evaluation metrics[3], including the associated standard error estimates.
- **Machine learning models** are generated for each of the experiments that is carried out, see chapter 6 for an overview of models used. Loosely based on empirical performance studies at least the following algorithms are tested: Random Forests, SVMs, KNNs, GLMs and variants, GBMs, Arima's and Neural Networks. Machine learning model analysis tools are described in chapter 7.
- **Meta-Analysis** is employed on top of factorial runs of experiments to analyse the impact of model and meta-parameter choices on predictive performance. Based on the meta-analysis robust models with meta-parameter sets are selected for each target. These models are subsequently trained and used for seismicity predictions. Details are described in chapter 8.
- **Results of the Meta-Analysis (relatively well performing robust models)** are provided in chapter 9, following a short overview of our hyperparameter tuning[4] approach. Support Vector Machines and GLM variants perform relatively well. Random Forests and K Nearest Neighbours perform relatively well for short-term forecasts, but we note that as these models cannot extrapolate, for future scenarios which are markedly different from past scenarios their longer term forecast performance can be impacted. Time delays larger than zero seem not to add statistical significant predictive power.
- **An evaluation of the seismicity event rate forecasts** is presented in chapter 10. Three aspects are considered: (i) a quantitative evaluation based on forecast performance; (ii) a qualitative evaluation based on forecast behaviour over a longer period of time; (iii) the range of validity of the methodology as presented.
- **Conclusions and a discussion** follow in chapter 11, with an extensive discussion on the diverging forecasts of various models for the post-March 2018 average production scenario and possible mitigation directions.
- **Next steps** in chapter 12 discuss three main directions: (i) extend the range of validity of the methodology and the definiteness of conclusions via the suggestions mentioned above; (ii) investigation of potential performance gain given by hybrid physics + machine learning models; (iii) extend the event rate methodology developed in this study with areal and magnitude resolution.

---

[3] The evaluation metrics guiding us throughout this study are the Mean Absolute Error (MAE), a standard choice in machine learning, and the Root Mean Square Logarithmic Error (RMSLE), particularly useful for count data with a large low-end tail as is the case here.

[4] Hyperparameters are the "control knobs" of machine learning algorithms, more details can be found in Appendix 4.

Throughout this study we point the interested reader to the appendices for additional background and details.

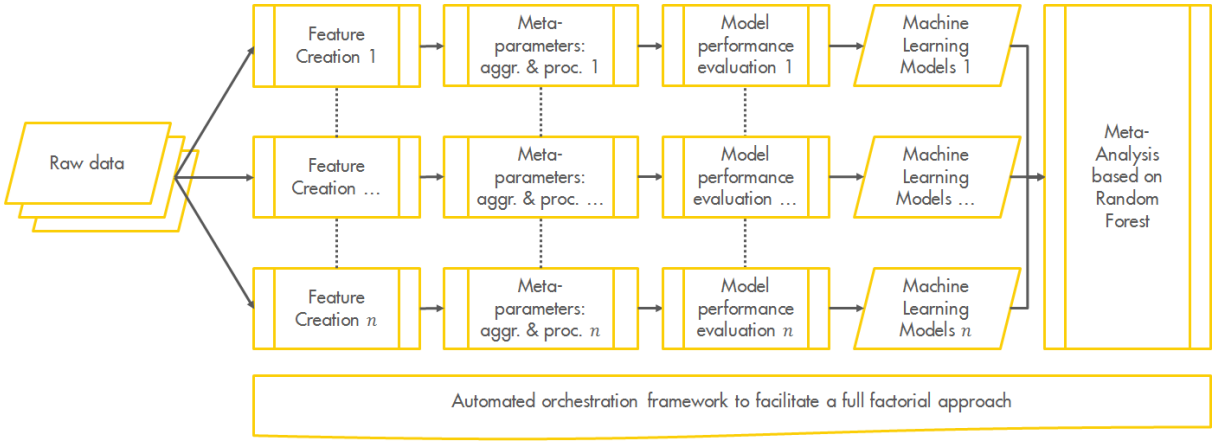


Figure 4: High-level overview of the forecast methodology of this study.

3 Data: Sources and Features

Machine learning models are fully based on the data provided and the features generated from it. This provides freedom in terms of functional relationships between features that can be explored but results in a strong dependency on the data. This dependency takes several forms: variables for which no data is provided cannot be included and will never show up as predictors; data granularity of a variable can impact detectability of effects due to that variable; biases in the observation of the target bias predictions; etc.

Machine learning methods require as input a finite set of feature instances generated from the data – this in contrast to e.g. physics based methods which are usually based on continuous variables. For example, where physics based formulae might use the (continuous) reservoir pressure gradient, machine learning algorithms would require a discretization of the same. The geospatial and temporal discretization, bin size and potential aggregation depend on the problem under study and are limited by the number of events available. These discretized and aggregate measures are called features and are offered to the algorithm as potential predictors, and one of them as the target: the feature we aim to predict. For completeness, we note that one input variable can result in multiple features or the opposite, e.g. the aggregated variable itself, its time derivatives, a variable to quantify regional variation, etc.

Most machine learning algorithms[5] require the features to be specified explicitly and we use this approach here. Consequently, our machine learning algorithms are restricted to view the data in terms of the features specified, which makes both aggregation choices and the choice which features to offer crucial. Such choices are usually guided by intrinsic data restrictions (e.g. a limited number of events) and domain expertise based assumptions about potential relevance of certain features. In this study we largely restrict ourselves to features which are directly related to the physical quantities of the data sources included, e.g. the mean aggregated reservoir pressure P, the mean aggregated pressure difference $\Delta P/\Delta t$, etc. In future work features related to specific physical processes could be introduced and tested for explanatory power (e.g. rate and state friction theories in which the seismicity rate $\sim\exp(P)$).

The first section of this chapter discusses which data sources are included. Each of the following sections describe one of these data sources in more detail, including how it is measured and where applicable, modelled, whether there are any uncertainties and which features are generated. Special attention is given to the target data source (seismicity) in section 3.2: the (for machine learning purposes) limited number of events (earthquakes) guides the aggregation choices to generate features from input data.

For those less familiar with the data sources under discussion, an exploratory overview of the data can be found in Appendix 1.

3.1 Data Overview & Selection

Selecting the right data is important: clearly data sources which are likely related to seismicity should be included. But with data sources of which the relation with seismicity is less evident one should be more restrictive, as adding more data sources does not necessarily improve predictions. For example, adding a data source that has no relation with seismicity might decrease predictive performance as algorithms will try to base predictions based on seismicity-independent variations in the unrelated data. These unrelated data sources could just by chance be correlated with the prediction target and thus lead to spurious results. Of course, for some data sources it is unclear a

[5] Some machine learning algorithms like neural networks can also generate features autonomously but this requires large numbers of events, much more than is available in the context of this study.

priori whether they would increase predictive power, so the following two-step approach usually works well and is also followed in this study.

1. A base data set is created by adding the data sources of which it is known or likely that there is a relation with the target and which is readily available;
2. Once models and predictions are made based on this base data set, additional more speculative or less readily available data sources can be added one by one. If such a data source increases predictive performance it can be kept, otherwise it can be discarded again.

This study only concerns step 1 above – suggestions for more speculative data sources are discussed in chapter 12 below.

Based on earlier work (see chapter 2 above) and domain expert discussions, we know or expect the following data sources to be related with induced seismicity: production, dynamic reservoir data (reservoir gas pressure, hydrocarbon column thickness, …), faults, compaction, subsidence and earthquakes themselves. As such all these data sources are included, see Table 1 below for an overview.

Data source	Example features	Source	Geo coverage & resolution	Temp. cov. & resolution	Notes
Target: Earthquakes	EQ rate	KNMI	Groningen area, ±0.5-1 km	1986 – 2017, incident based	Geo resolution time dependent
Production Q	$Q, \Delta Q/\Delta t, var(Q)$	NAM Energy Components	Per production cluster	1965 – 2016, per day	
Dynamic Res.: • **Pressure P** • **HCT, HCM** [6]	$P, \Delta P/\Delta t, HCT, \frac{\Delta HCM}{\Delta t}$, …	MoReS v4.0, NAM Energy Components	Grn. reservoir, irregular grid (10s x 10s m)	1965 – 2016, monthly	Measured at wells only, model interp. to grid
Subsidence S	$S, \Delta S/\Delta t, var(S)$	Shell GSNL	Groningen area, grid of 2.5 km² – 5.0 km²	1972 – 2013, per 5 years	Optical levelling used, other data sources available
Compaction C	$C, \Delta C/\Delta t, var(C)$	Shell GSNL	Groningen area, reg. grid of 2.5 km²	1972 – 2013, per 5 years	Estimated from subsidence, see section 3.4
Faults	Used as geospatial filter	Petrel, Shell GSNL	Grn area, above seismic resolution (see sec. 3.7.2)	Assumed static	Properties e.g. thickness, azimuth, dip angle, …

Table 1: Overview data sources used for this study, red: target (to be predicted) and yellow & white features (potential predictor).

To train a machine learning algorithm, i.e. let the algorithm find the association between the target (values to be predicted) and features (potential predictors), the target and features are integrated into a tabular structure as shown earlier in Table 2. Here each row of the table represents a "learning instance" or bin, representing an aggregated space-time interval. Each feature is a yellow column with each cell showing the feature value for a specific bin. The red column shows the target value

[6] HCT stands for Hydrocarbon Column Thickness, the hight of the hydrocarbons in the reservoir. HCM stands for Hydrocarbon Column Mass, the mass of a column of hydrocarbons.

for the bin. For historical bins the red target column is filled, for predictions it is up to the algorithm to fill the target column values based on the associations between features and target the algorithm learned on the historical values.

It is worth noting that compared with many other situations in which machine learning is applied, the number of target events on which the algorithms can be trained is relatively limited (several hundreds of earthquakes).

Region	Temporal Interval	Target, e.g. EQ count (#)	Feature 1, e.g. Q (10^9 m^3)	…	Feature m, e.g. var. C (10^{-11} m)
GFO	1995-Q1	1	3.62	…	4.44
GFO	1995-Q2	2	1.64	…	2.50
GFO	1995-Q3	0	0.68	…	2.08
…	…	…	…	…	…

Table 2: Schematic representation of data aggregation and integration. Blue: aggregation parameters; red: target (to be predicted); yellow: features (potential predictors).

The next sections describe each of the data sources above in more detail. Given the limited number of events, the choice for temporal and geospatial aggregation depends on target processing choices. Therefore, aggregation choices are discussed in conjunction with the discussion on earthquake data.

3.2 Earthquake Data and Defining the Target

The goal of this study is to predict seismicity event rates: the number of earthquakes within a certain time interval, within a certain region, above a certain minimum magnitude. This section describes earthquake measurements and the choices made to generate the target from these measurements.

3.2.1 Earthquake measurements

The KNMI (the Royal Netherlands Meteorological Institute) has seismicity monitoring stations throughout the Netherlands and specifically in Groningen[7]. The network is described in more detail in e.g. (Dost, Goutbeek, Van Eck, & Kraaijpoel, 2012) and (Dost & Haak, 2002). Measurements from this network are automatically processed by KNMI and earthquakes detected are formally published in a catalogue[8], which we use as source for earthquake detections. The induced earthquake catalogue has a straightforward structure as shown in Table 3: the data is provided in tabular form which each row representing an event. Of each event its date and time, location, latitude, longitude and depth as well as magnitude and evaluation mode are given. Here most fields are self-explanatory, possibly except for the location field[9] but that field isn't used in our analysis.

[7] For an overview of these stations, see https://www.knmi.nl/nederland-nu/seismologie/stations.

[8] Catalogue available at https://www.knmi.nl/kennis-en-datacentrum/dataset/aardbevingscatalogus.

[9] Up to November 30, 2016 the location field described the city or village centre nearest to the event, whilst as of December 1, 2016 the municipality border within which the event took place is registered.

Date	Time	Location	Lat	Lon	Depth	Mag	Eval mode
1986-dec-26	07h47m51s	Assen	52.992	6.548	1	2.8	Manual
1987-dec-14	20h49m48s	Hooghalen	52.928	6.552	1.5	2.5	Manual
…	…	…	…	…	…	…	…

Table 3: KNMI induced earthquake catalogue data structure

3.2.2 Uncertainties

The number of sensors in the seismic sensor network, their locations and the data processing procedures used influence detection sensitivities and location uncertainties. As the network has been extended over time, detection sensitivity and location uncertainties vary over time. Table 4 provides an overview of sensitivities as reported by the KNMI, see e.g. (Dost, Goutbeek, Van Eck, & Kraaijpoel, 2012), (Kraaijpoel, Caccavale, Van Eck, & Dost, 2015), (Dost, Ruigrok, & Spetzler, 2017), (Spetzler & Dost, 2017) and the overview of stations referred to above. In general, the horizontal location uncertainty is around 1 km and the vertical uncertainty is between 1-2 km. Given the large vertical uncertainty, vertical locations are pre-set to 3 km for nearly all events.

Time	Detection	Localisation	Comments
Since 1995	≥ 1.5	≥ 2.3-1.5	Network installed (8 borehole stations in Northern Netherlands)
±2010	Processing software upgrade, real-time continuous data transmission		
2009-2010	≥ 1.0	≥ 1.5	6 additional borehole stations in Northern Netherlands
2015-2017		≥ ~0.5	Major extension: 64 additional borehole stations in Northern Netherlands

Table 4: KNMI Seismic Sensor Network developments over time

3.2.3 Choice of minimum magnitude M_{min}, temporal interval and temporal aggregation period T_{agg}

The magnitude of completeness M_c of a sensor network is usually defined as the lowest value of the moment magnitude of an event for it to be detected with 100% reliability. Event counts with a moment magnitude below M_c are incomplete, which in principle doesn't need to pose a problem for machine learning algorithms as long as M_c is constant over time: algorithms would simply predict observed seismicity. However, with the detection sensitivity increasing over time as evident from Table 4, an increase in the detection of earthquakes is a combination between a change in seismicity and a change in detection sensitivity. As this effect is strongest for low magnitude seismicity a minimum magnitude cut-off M_{min} is chosen, only earthquakes with a magnitude equal or higher than M_{min} are taken into account. A sensible choice for M_{min} is the magnitude of completeness M_c – this choice would ensure that all signal picked up comes from seismicity instead of sensor network sensitivity changes. Given the improvements in the sensor network over time, the choice of M_c and the start of the temporal interval T_{start} are coupled: a later T_{start} might allow for a lower M_c and vice versa. The choice for both parameters is, of course, driven by the desire to use as much of the data as possible, while avoiding the introduction of bias.

In literature various authors made various estimates for M_c:

- Following the KNMI reported values (see references in section 3.2.2), the default PSHRA seismological model (Bourne & Oates, Extreme Threshold Failures Within a Heterogeneous Elastic Thin Sheet and the Spatial-Temporal Development of Induced Seismicity Within the Groningen Gas Field, 2017) uses $M_c = 1.5$ from 1995 onwards.
- A probabilistic method based on empirical detection probabilities (Van Thienen-Visser, Sijacic, Van Wees, Kraaijpoel, & Roholl, 2016) leads to the M_c contour plots shown in Figure 5. The plots suggest that for the Groningen field prior to 2010 $M_c = \sim 1.5$, whereas between 2010 and 2014 $M_c = \sim 1.3$.
- Using a Hill-plot (Post, 2017) estimates $M_c = 1.3$ between 1995-2010 and $M_c = 1.1$ between 2010 and 2017.
- Based on the maximum curvature method (Paleja & Bierman, 2016) estimate $M_c \leq 1.2$ from 2003 onwards and indicate that given the limited number of events prior to 2003, an estimate for M_c is statistically not possible.

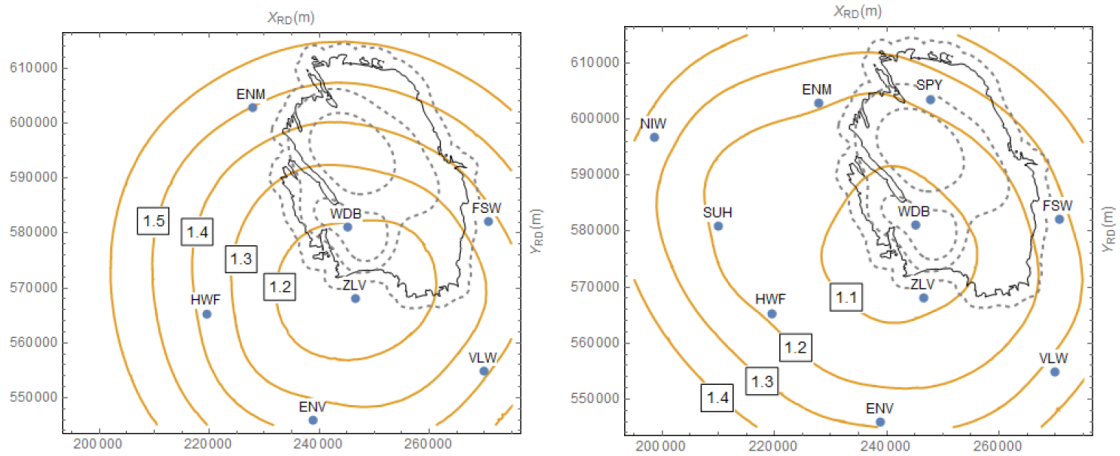


Figure 5: magnitude of completeness contours for the Groningen borehole network in the period 1996-2010 (left) and 2010-2014 (right) based on a probabilistic model for event detection (Van Thienen-Visser, Sijacic, Van Wees, Kraaijpoel, & Roholl, 2016). For this model the magnitude of completeness is defined as lowest magnitude that has a 95% probability of being detected in 3 or more borehole stations. Figures © TNO.

Given the wide variety of choices in literature we proceed with our own analysis. There are various methods for estimating the magnitude of completeness given a catalogue of events. A summary is provided in (Mignan & Woessner, 2012). Most of the methods assume the validity of the Gutenberg-Richter law[10], including the two methods we use: (i) the maximum curvature technique and (ii) the b-value estimates.

First, the maximum curvature technique (Wiemer & Wyss, 2000) requires relatively few events to reach a stable result and is statistically robust but tends to underestimate M_c in bulk data (Mignan & Woessner, 2012). Figure 6 shows the frequency of *observed* events with magnitude M or greater versus M, for the Groningen catalogue between the dates of 1st May 1995 and 31st December 2016. The log-linear plot levels off from the linear relationship corresponding to the Gutenberg-Richter law at lower magnitudes and this is attributed to an increasing fraction of the actual events

[10] The Gutenberg-Richter law is an empirical law in seismology stating that the frequency of events of magnitude M or greater decreases with increasing M as 10^{-bM}, where b is a constant called the b-value.

remaining undetected as the magnitude decreases. An estimate of the moment of completeness M_c can be obtained by taking M_c just above the "knee" in the curve, suggesting $M_c = 1.2$.

Second, the b-value stability estimate (Cao & Gao, 2002) tends to overestimate M_c. This technique has a relatively high uncertainty (Mignan & Woessner, 2012). We proceed by computing the b-value as a function of minimum magnitude used in the computation. The maximum likelihood estimator given in (Harris & Bourne, 2015) is used and the results are shown in Figure 7. It is seen that b increases more-or-less linearly with magnitude and then stabilizes. Another estimate of M_c is the value at which stabilization is observed, suggesting $M_c = 1.2$ as well.

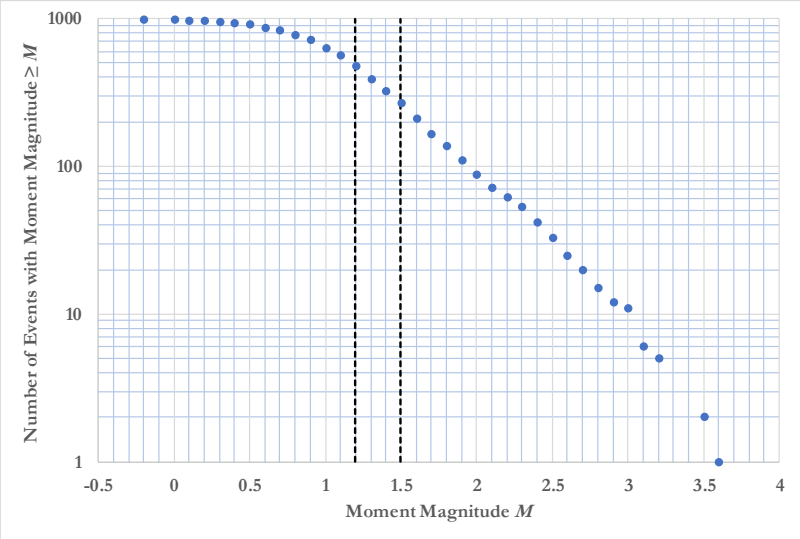


Figure 6: Number of events in the Groningen catalogue (May 1st,1995 to December 31st, 2016) with a moment magnitude $\geq M$ as a function of M. The Gutenberg-Richter law corresponds to a straight line on the log-linear plot, which flattens off at low magnitudes due to the detectability of events falling below 100%. The dashed lines highlight the values $M = 1.2$ and $M = 1.5$.

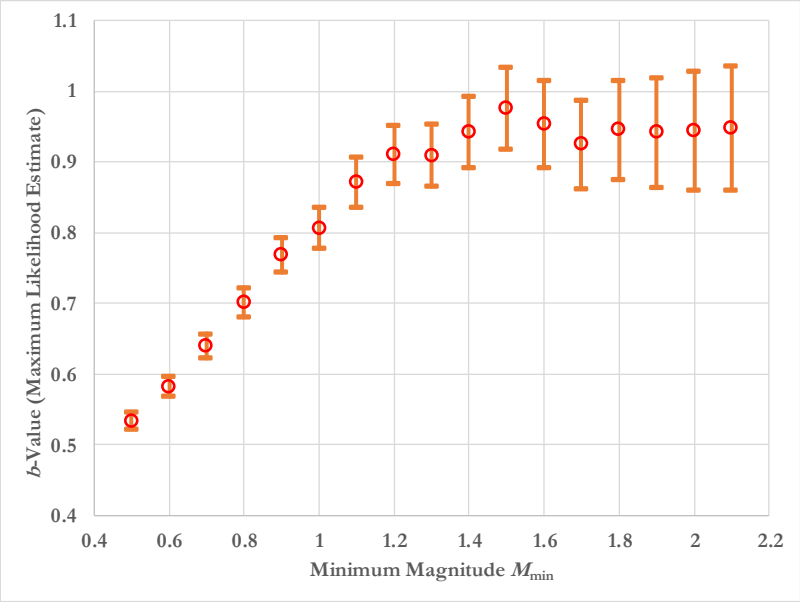


Figure 7: Maximum likelihood estimator of Gutenberg-Richter b value versus minimum magnitude for the Groningen catalogue (1st May 1995 to 31st December 2016). Moving from right to left in the plot the b value estimates remains relatively constant until they decrease due to detectability of events falling below 100%. The error bars correspond to $\pm$ one standard deviation and are obtained using bootstrap simulation.

Considering pure bin-counts of the bins used in the analysis above, Figure 8 shows the bin counts for observations as well as for theoretical Gutenberg-Richter relationships assuming $M_{min} = 1.5$ and $M_{min} = 1.2$.

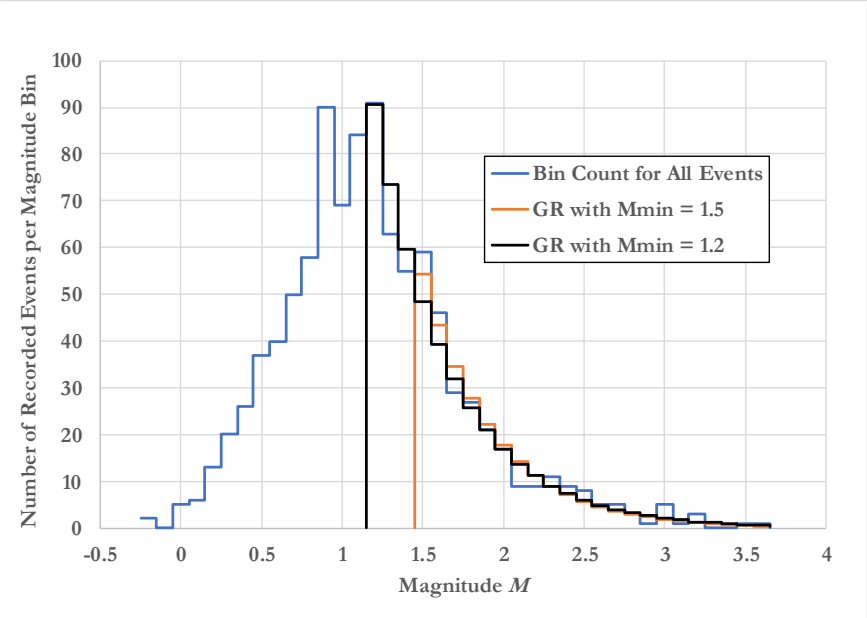


Figure 8: Bin counts used in the analysis of M_c for observations (blue), as well as the theoretical counts for a Gutenberg-Richter relation with $M_{min} = 1.5$ (orange) and with $M_{min} = 1.2$ (black) using the maximum likelihood estimators for b.

The starting time of the temporal interval chosen depends on the choice of M_c. As end of the temporal interval we take the last year which was fully available at the start of this study, being 2016. Risk in the context under consideration is an annual property, so the largest temporal aggregation period we choose is one year. As production is seasonal, there could be seasonal variations in earthquake rate. To be able to capture any seasonal effect, temporal aggregation needs to be at sub-year level, ideally at least 4 data points per year. In light of the number of events, a temporal resolution below one month might not provide additional insights. Combining the above, a temporal aggregation level of 1, 3 or 12 months is chosen.

Following the analysis above we proceed with the following choices for M_{min}, T_{start} and T_{agg}:

- In line KNMI reported M_c values and following the PSHRA default we choose $M_{min} = 1.5$ with $T_{start} = 1995$. We experimented with a T_{agg} of 1, 3 and 12 months: 1 month aggregation intervals gave too many zero bins to be useful whilst 12 months aggregation gave insufficient bins. As such, we only progress $T_{agg} = 3$ months. This combination gives us 265 events spread over 88 bins.
- Following the above discussion of M_c we find $M_{min} = 1.2$ to be worth considering as an alternative to $M_{min} = 1.5$, whilst acknowledging the possibility that M_c could exceed this choice of M_{min}. This would mean, in turn, that the detectability function would differ between training and forecasting period. With $T_{start} = 1995$ this choice of M_{min} nearly doubles the number of events to 464 over the same amount of bins (staying with $T_{agg} = 3$ months).
- Additionally, following (Paleja & Bierman, 2016) we also take $M_{min} = 1.2$ as of the first date M_{min} can be statistically evaluationed, conservatively giving us $T_{start} = 2004$. Keeping $T_{agg} = 3$ months this gives 392 events over 52 bins.

- Finally, for comparative reasons with other reports (e.g. (Pijpers, Interim report: correlations between reservoir pressure and earthquake rate, 2017), (Van Thienen-Visser, et al., 2015) and (Paleja & Bierman, 2016)) we take $M_{min} = 1.0$ with $T_{start} = 2004$. We highlight that the choice of $M_{min} = 1.0 < M_c$ and as such we risk that the signal the models picks up is a combination of seismicity and changes in the detectability function. This results in 513 events, for which we attempt $T_{agg} = 1$ month, resulting in 156 bins.

3.2.4 Choice of geospatial interval and aggregation

This study is about gas production induced seismicity on the Groningen field, so the geospatial area we choose is delineated by the outline of the Groningen Field Outline (GFO), see Figure 9 below for a geographical overview.

Figure 9: Groningen Field Outline (GFO) geospatial view Google Maps (2018)[11]

We can bin the region in piece-wise constant subregions, but this leads to more bins. Given the relatively limited number of events (earthquakes) and high number of bins some combination of choices for minimum magnitude, temporal interval and temporal aggregation, we restrain ourselves in this study to GFO only. Although this choice follows from a practical machine learning limitation, we note that as the Groningen gas field is considered to be a communicating vessel differences in dynamic reservoir properties between regions are expected to be limited. A more detailed analysis of the trade-off between temporal and geospatial aggregations and what might be useful geospatial aggregations are left as a subject for further study.

3.2.5 Choice of target quantity

Three possible target quantities have been generated for this study:

- Earthquake count: the number of earthquakes equal or larger than the minimum magnitude within the temporal and spatial intervals;
- Earthquake rate: earthquake count divided by the length of the temporal interval (equivalent with earthquake count for uniform temporal intervals);

[11] Map data © GeoBasis-DE/BKG (© 2009) Google. Google Maps image retrieved from: http://maps.googleapis.com/maps/api/staticmap?center=53.5,7&zoom=9&size=640x640&scale=2&maptype=roadmap&language=en-EN&sensor=false

- Earthquake energy: let $M_i \geq M_{\min}$ be the magnitude of earthquake i, then the earthquake energy is given by: $\sum_i 10^{M_i}$, where the sum is over all earthquakes within a temporal or spatial interval.

Although the processing for the methodology developed in this study is largely automated, analysis and results interpretation for a given target quantity remains a time intensive human endeavour so far. As such, for this study we focus on earthquake rate.

3.3 Production Data

The root cause of induced seismicity is production, so production data forms the starting point of our analysis.

3.3.1 Production measurements

NAM has 20 production facilities spread over Groningen. Most of these facilities have multiple production wells (around ten to twenty). The total volume produced is measured at well level and aggregated to daily production cluster level. The general structure of the data is shown in Table 5: for each production well (associated to a production cluster) located in a certain region and area for a certain date the amount of gas produced (in m^3) and the amount of water produced (in m^3) are included, just as the BHP and THP in bar.

Well name	Prod. cluster	Region	Area	Date	Gas (m^3)	Water (m^3)	BHP (bar)	THP (bar)
WAMR1	AMR	East	Central	1956-feb-01	0	0	345.442	0
WAMR1	AMR	East	Central	1956-mar-01	0	0	345.448	0
…	…			…	…	…	…	…
WLRM12	LRM	Loppz	Northwest	2015-nov-1	2.45E6	30.514	87.647	0
…	…			…	…	…	…	…

Table 5: Production data structure

Two future production scenarios are investigated: (i) the Winningsplan 2016 [Production Plan 2016] production policy scenario (NAM, 2016) and (ii) the average production scenario announced by the Ministry of Economic Affairs and Climate in March 2018 [hereafter the average post-March 2018 production scenario] (Ministry of Economic Affairs and Climate, 2018). Both scenarios start from January 1st, 2017 and continue up to December 31st, 2025 or thereafter, see Figure 10. As is clear from the figure, the Production Plan 2016 scenario assumes steady state gas production up to the forecast horizon. The average post-March 2018 production scenarios were commissioned by the Minister of Economic Affairs and Climate (Ministry of Economic Affairs and Climate, 2018) following an earthquake with magnitude 3.4 in the village of Zeerijp. This production scenario reduces gas production from the Groningen field to below 12 billion Nm^3 by 2022 and to zero by 2030.

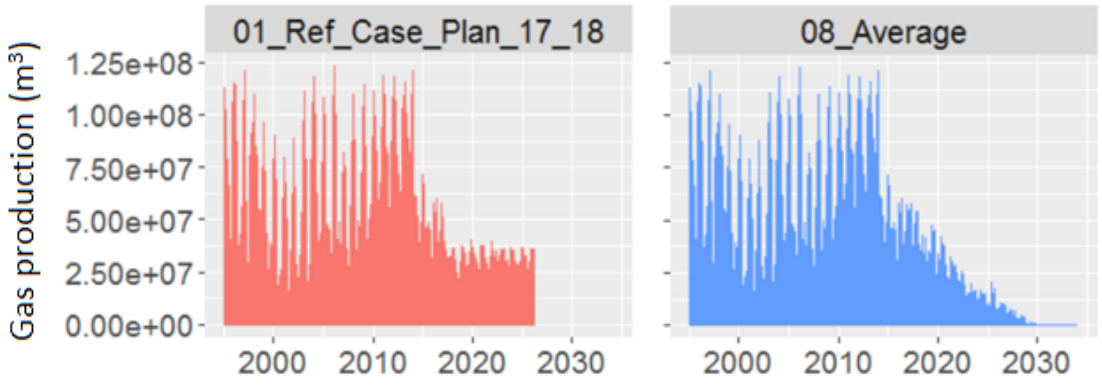


Figure 10: Visualization of production scenarios with left the Production Plan 2016 production scenario and right the average post-March 2018 production scenario.

3.3.2 Uncertainties

There are few uncertainties regarding the historical production data since the values for amount of gas extracted are measured at the well level using precise sensors. For this reason and unlike other data sources where modelling assumptions and interpolations need to be made we have a high degree of certainty that the production data is accurate with respect to its historical values.

Future production scenarios depend on policy. This study incorporates the latest policy as of writing.

3.3.3 Feature generation

From the production data the following features are derived:

- Gas production Q in GFO during the temporal aggregation interval $[m^3]$;
- Geospatial variance in gas production between production clusters in GFO $var(Q)$ during the temporal aggregation interval $[m^6]$;
- The first and second order temporal differences of the quantities above, with difference lengths the same size as the aggregation window used (3 months for most targets).

The correlation of these features with seismicity is shown in Figure 11.

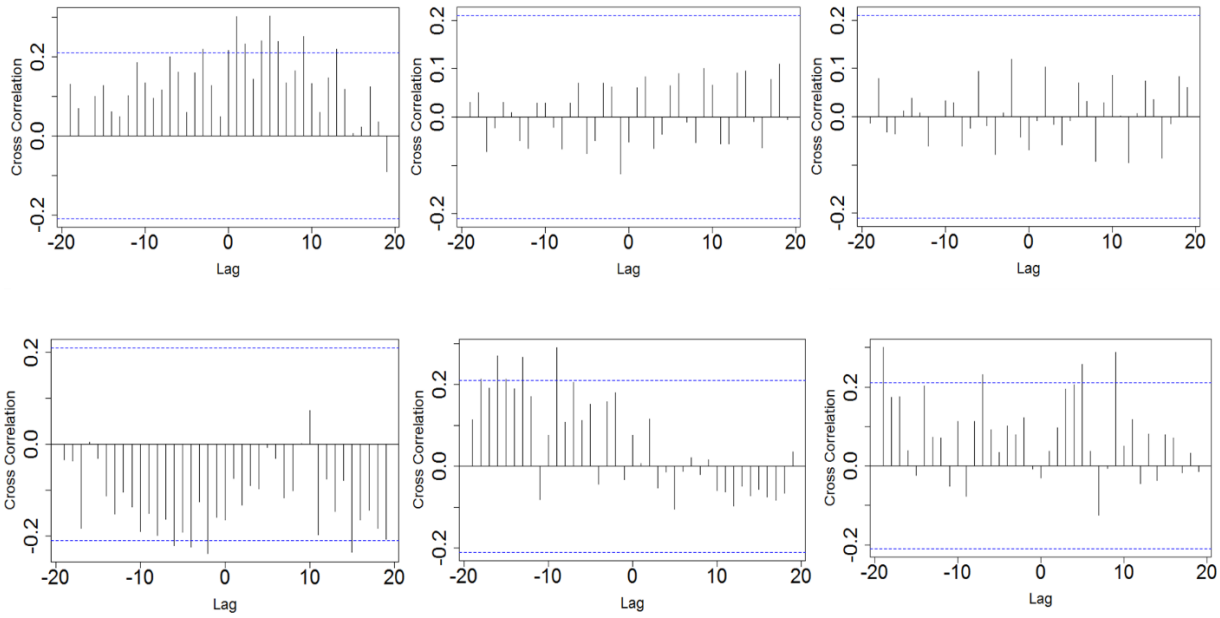


Figure 11: Sample cross-correlation between earthquake rate and production features for earthquakes with $M \geq 1.5$ from 1995 onwards for 3 month intervals. Top row: gas production Q (left) and its first and second temporal difference (middle and right); bottom row: geospatial variance in gas production and its first and second temporal derivative. Blue lines: 95% confidence intervals, only cross-correlations which extend beyond these lines are statistically significant.

3.4 Dynamic Reservoir Data

Gas is produced by extraction of gas from the reservoir. This influences the time-varying or dynamic reservoir properties like reservoir pressure.

3.4.1 Dynamic reservoir measurements

The dynamic reservoir quantities used in this study are the reservoir pressure, the hydrocarbon column length (HCL, the height of the hydrocarbon column in the reservoir) and the hydrocarbon column mass (HCM, the mass of the hydrocarbon column). The quantities have been obtained from version v4 of the MoReS asset model for Groningen (Van Oeveren, Valvatne, Geurtsen, & Van Elk, 2017), which takes as input reservoir pressure measurements at production clusters and observation wells, additional pressure measurements directly after drilling (repeat formation tests), the water level at well locations, subsidence measurements and the static subsurface model (e.g. faults, rock compositions, etc) based on seismic reflection imaging[12]. The model is history matched, meaning that it is in broad agreement with the available historical measurements the measured quantities. The model and supporting scripts were made available by NAM Groningen Development, who use the model for forecasting and reporting purposes. For each production scenario as described in section 3.3.1 a MoReS run is available.

We note a limitation in the way HCT is calculated: HCT will only change value once the HC saturation of a complete grid cell in MoReS is below a certain threshold. Consequently, HCT

[12] A geophysical subsurface imaging technique where seismic waves are artificially created via controlled seismic source (e.g. explosions, a specialized air gun, …). Reflections of the seismic waves against interfaces between two materials are measured by detectors and can be used to create a 3D subsurface map. The technique is similar to sonar.

changes very slowly in time. HCM does not have this limitation and changes continuously over time.

The properties are not exported per 3D MoRes grid cell but first averaged in the vertical direction by taking individual grid cell volumes into account. Note that taking the volumes into account is important since the model uses local grid refinement which leads to much smaller grid cell volumes around the wells compared to the rest of the cells that are further away from the wells. The structure of the data obtained per timestamp is shown in Table 6, providing the grid cell *x* and *y* centers, the reservoir top, the grid block volume, the pressure, the hydrocarbon column height and the hydrocarbon column mass.

XCenter	YCenter	Ztop	Grid block vol	Pressure	HC col. height	HC col. mass
m	m	m	m^3	Bar	m	kg
265070	565990	-3332.1	1886400	396.42	0	0
265360	566070	-3428.3	3587300	411.88	0	0
264060	566020	-3323.9	1076200	395.87	0	0
264330	566080	-3321.7	4690400	395.38	0	0
264660	566130	-3295.2	8505600	391.75	0	0
…	…	…	…	…	…	…

Table 6: Dynamic Reservoir data structure, January 31st 1958

3.4.2 Uncertainties

Even though the MoReS model is history matched over a long production history (Van Oeveren, Valvatne, Geurtsen, & Van Elk, 2017), inevitably several uncertainties remain in the reservoir model. For forecasting purposes and for uncertainty management NAM Groningen Development normally uses a P10, P50, and P90 realization based on the known uncertainty space. However, these realizations are currently only available for an earlier version than v4 of the reservoir model. They have been made available to us, hence at a later moment in time those could be revisited and used to obtain a better assessment of the known uncertainties of the reservoir and made part of the dataset since some methods can leverage this additional prior information for improved modelling results and better uncertainty quantification in the predictions (e.g. Bayesian methods).

It should be noted that major uncertainties remain regarding rock properties like porosity and permeability away from the wells. However, given their static nature they have so far not been used directly in this study. It can be assumed though that key dynamic properties, like pressures, are rather constrained through the long period of production history to which the model has been history matched – meaning the model is calibrated using historical data. Consequently, some future information leakage has occurred and the prediction uncertainties presented later might be an underestimate of future uncertainties.

3.4.3 Feature generation

Since the data from the reservoir simulator only outputs data in monthly intervals it can happen that no direct sample is available for a specific moment in time in which aggregation should take place. For this purpose, we utilize linear and also optionally cubic spline interpolation to fill in the gaps. The following dynamic reservoir data features are generated:

- The grid cell volume weighted mean reservoir pressure $\text{weighted mean}(P)$ in GFO during the temporal interval [bar];
- The grid cell volume weighted mean reservoir pressure length $\text{weighted mean}(PL)$ in GFO during the temporal interval [bar m];
- The grid cell volume weighted mean hydrocarbon column thickness $\text{weighted mean}(HCT)$ in GFO during the temporal interval [m];
- The grid cell volume weighted mean hydrocarbon column mass $\text{weighted mean}(HCM)$ in GFO during the temporal interval [kg];
- The first and second temporal difference of the quantities above, with difference lengths the same size as the aggregation window used (3 months for most targets);
- The first and second relative temporal difference (temporal difference divided by the quantity itself) of the quantities above. The relative differences have been included following (Pijpers, A phenomenological relationship between reservoir pressure and tremor rates in Groningen, 2016).

The correlation of these features with seismicity is shown in Figure 12.

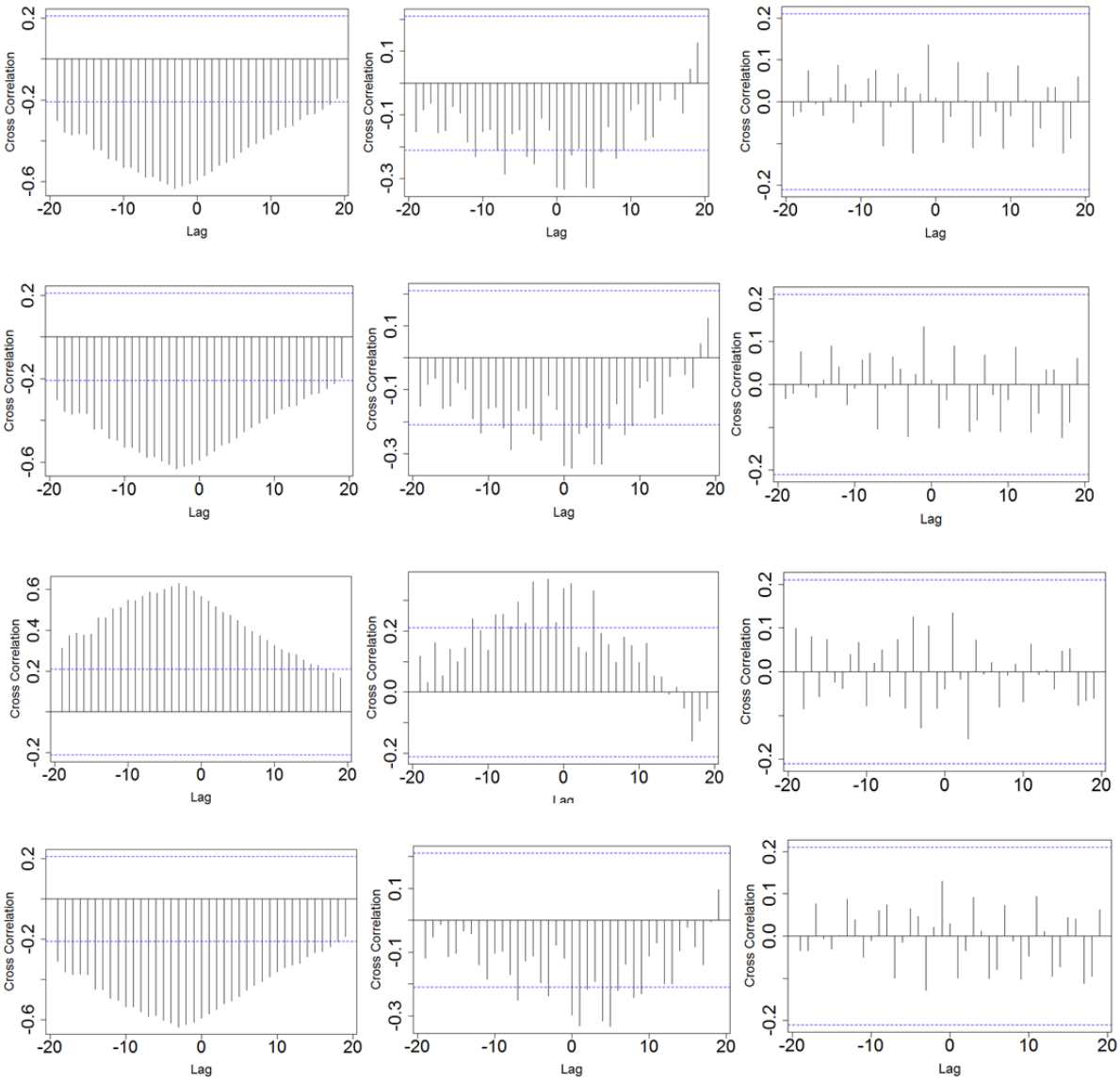


Figure 12: Sample cross-correlation between earthquake rate and dynamic reservoir features for earthquakes with $M \geq 1.5$ from 1995 onwards for 3 month intervals. Top row: weighted mean reservoir pressure P (left) and its first and second temporal difference (middle and right); top-middle row: weighted mean pressure length PL and its first and second temporal difference; bottom-middle row: HCT and its first and second temporal difference; bottom row: HCM and its first and second temporal difference. Blue lines: 95% confidence intervals, only cross-correlations which extend beyond these lines are statistically significant.

3.5 Compaction Data

Decrease in reservoir pressure means the reservoir cannot sustain the gravitational load from the mass above the reservoir, due to which the reservoir compacts. Compaction can hardly be measured directly and therefore is often derived from reservoir pressure or subsidence.

3.5.1 Compaction measurements

The compaction data that is used in this study has been derived by Shell GSNL (Bierman, Randell, & Jones, Bayesian methods for reservoir compaction estimation, applied to the Groningen gas

field, 2017) and is based on the pressure data for a given production scenario exported from version v4 of the MoReS asset model for Groningen (Van Oeveren, Valvatne, Geurtsen, & Van Elk, 2017). Subsidence levelling data has been used to calibrate the model. The compaction data is modelled at 5813 RD_XY locations which provide a fairly granular representation of the compaction in the reservoir. Table 7 shows the resulting data structure: a set of x and y coordinates followed by subsidence estimates of these coordinates for different time instances.

RD_X	RD_Y	1958-01-30 (m)	…	2025-12-30 (m)
228000	611500	2.392548e-06		0.09008615
228500	587000	2.392548e-06		0.06588574
…	…	…	…	…

Table 7: Compaction data structure

3.5.2 Uncertainties

The compaction data that is available to us is the result of a model which is based on the simulated values for pressure from the MoReS asset model (Van Oeveren, Valvatne, Geurtsen, & Van Elk, 2017), as such in general any uncertainties and limitations that are relevant to the dynamic reservoir data will also have an impact on the quality of the compaction data that has been modelled after them. Moreover, this realization of the compaction data is but one of several possible ways in which the compaction can be estimated, for example compaction has been estimated using a Bayesian approach as well as regularized direct inversion (Bierman, Kraaijeveld, & Bourne, 2015). We choose for the Bayesian approach as the method performs equally good or better than other methods in an out-of-sample cross-validation test with different prediction horizons (varying from 1 to 10 years) and different datasets (optical levelling and PS-InSAR). We note that the compaction data available to us is intended for future predictions and therefore has been calibrated using matched reservoir pressures up to December 2016 and optical levelling survey data up to March 2013. This means that for all predictions up to December 2016, some future information leakage has occurred and the prediction uncertainties presented later might be an underestimate of future uncertainties.

3.5.3 Feature generation

The compaction data is provided at the start and end of every month, for practicality we only use the records from the first day of each month. When data is required for a time instance for which no data is available (e.g. halfway the month) linear interpolation is applied to obtain an estimate. The following compaction features are generated:

subsidence features are generated:

- The mean total compaction since the start of the gas production $\mathrm{mean}(\text{cumulative } C)$ during the temporal interval $[m]$;
- The mean compaction $\mathrm{mean}(C)$ in GFO during the temporal interval $[m]$;
- The mean first temporal difference of the compaction $\mathrm{mean}(\Delta C/\Delta \mathrm{t})$ in GFO during the temporal interval $[m/s]$, with the difference length the same size as the aggregation window used (3 months for most targets);
- The geospatial variance of instead of the mean for the three quantities above.

The correlation of these features with seismicity is shown Figure 13.

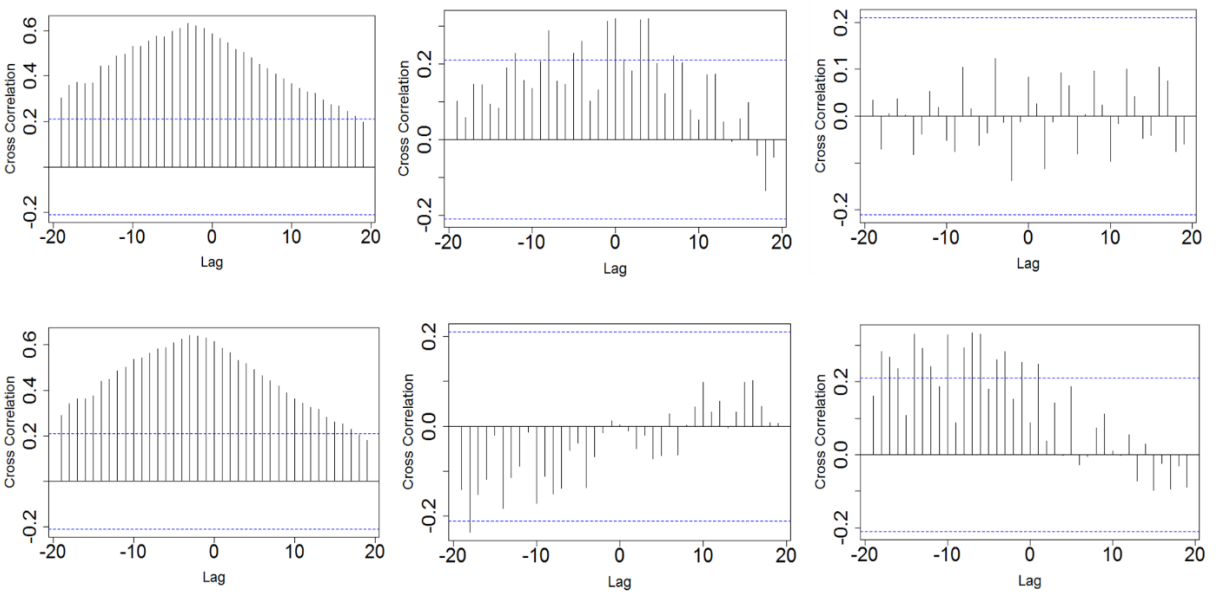


Figure 13: Sample cross-correlation between earthquake rate and dynamic reservoir features for earthquakes with $M \geq 1.5$ from 1995 onwards for 3 month intervals. Top row: mean total compaction C (left), the mean compaction during the time interval (middle) and the first temporal difference between time intervals (right); bottom row: the total geospatial variance of compaction $var(C)$, the geospatial variance of compaction during the time interval and its first temporal derivative. Blue lines: 95% confidence intervals, only cross-correlations which extend beyond these lines are statistically significant.

3.6 Subsidence Data

Reservoir compaction leads to surface subsidence, which is a relatively easy measurable quantity.

3.6.1 Subsidence measurements

The subsidence data was made available in pre-processed model form by Shell GSNL (Bierman, Randell, & Jones, Bayesian methods for reservoir compaction estimation, applied to the Groningen gas field, 2017). This pre-processed model data is based on a deterministic function to transform reservoir compaction into surface subsidence. Subsidence estimates are available on 943 measurement points of optical levelling surveys: height difference measurements between pairs of benchmarks in closed loops, see Figure 14. Table 8 shows the resulting data structure: a set of x and y coordinates followed by subsidence estimates of these coordinates for different time instances.

RD_X	RD_Y	1958-01-30 (m)	…	2025-12-30 (m)
228000	611500	-8.65e-06		0.0888
228500	587000	-2.05e-07		0.1641
…	…	…	…	…

Table 8: Subsidence data structure

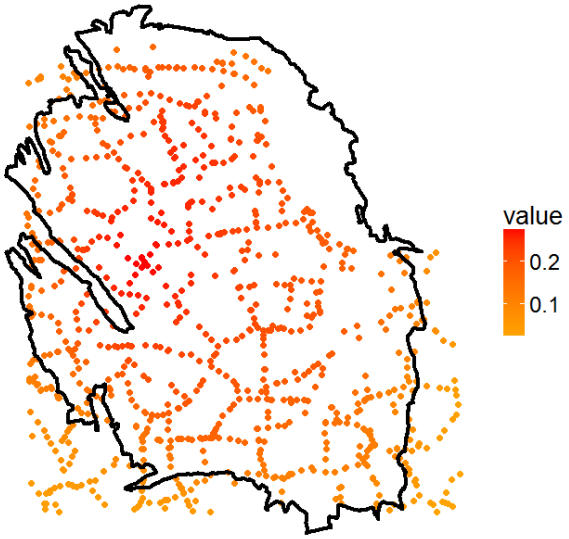


Figure 14: Subsidence graphical representation example for January 1st, 2004. Dots indicate measurement points, the dots show the standard routes on which subsidence measurements are obtained.

3.6.2 Uncertainties

The subsidence data provided is based on the result of two models: (i) a deterministic function which transforms compaction in subsidence and (ii) the simulated values for pressure from the MoReS asset model (Van Oeveren, Valvatne, Geurtsen, & Van Elk, 2017), on which the compaction estimates are based. As such any uncertainties that are relevant to either models will also have an impact on the quality of the subsidence data that has been modelled after them.

Analogously to the compaction data, we note that the data available to us has been calibrated using matched reservoir pressures up to December 2016 and optical levelling survey data up to the March 2013. This means that for all predictions up to December 2016, some future information leakage has occurred and the prediction uncertainties presented later might be an underestimate of future uncertainties.

3.6.3 Feature generation

The subsidence data is provided at the start and end of every month, for practicality we only use the records from the first day of each month. When data is required for a time instance for which no data is available (e.g. halfway the month) linear interpolation is applied to obtain an estimate. The following subsidence features are generated:

- The mean total subsidence since the start of the gas production $\mathrm{mean}(\mathrm{cumulative}\ S)$ during the temporal interval $[m]$;
- The mean subsidence $\mathrm{mean}(S)$ in GFO during the temporal interval $[m]$;
- The mean first temporal difference of the subsidence $\mathrm{mean}(\Delta S/\Delta \mathrm{t})$ in GFO during the temporal interval $[m/s]$, with difference lengths the same size as the aggregation window used (3 months for most targets);
- The geospatial variance of instead of the mean for the three quantities above.

The subsidence and compaction data turn out to have a correlation higher than our correlation threshold (section 4.4), so offering them both to the machine learning algorithms will not provide additional signal. To be able to leverage both subsidence and compaction data, we also define the following feature:

- The difference between mean subsidence and mean compaction $\Delta SC = mean(C) - mean(S)$.

The cross correlation of this feature with seismicity is shown in Figure 15.

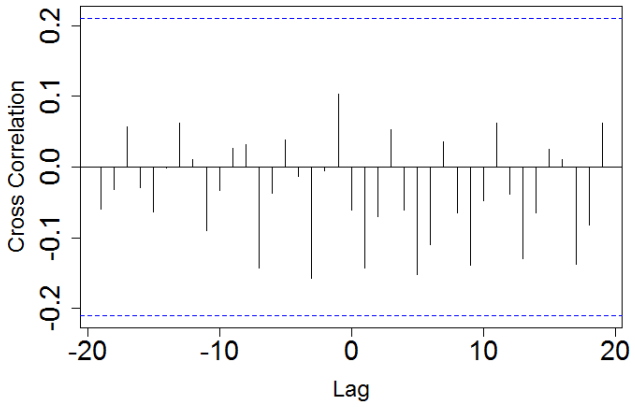


Figure 15: Sample cross-correlation between earthquake rate and the difference between compaction and subsidence for earthquakes with $M \geq 1.5$ from 1995 onwards for 3 month intervals. Blue lines: 95% confidence intervals, only cross-correlations which extend beyond these lines are statistically significant.

3.7 Fault data

Earthquakes occur at faults, which are considered to be temporally static over the timescales considered in this study. As such, we cannot use the fault data directly in our temporal varying approach: the feature would be constant for each target instance and thus not provide discriminatory information to the machine learning algorithms. Therefore, instead we have used the fault data to filter the dynamic data feature maps (section 3.4) such that only the grid cells whose centres are less than 500 meters away from the mapped fault lines (the horizontal localization uncertainty of earthquakes) are used. The 500 meter is chosen based on the horizontal location uncertainty of seismicity measurements (section 3.2.2) and as such represents the area most likely impacted by earthquakes on mapped faults.

3.7.1 Fault measurements

Seismic imaging is used to obtain 3D cubes of the subsurface from which faults are interpreted. The latest fault maps (at the time of writing) (Visser, 2012), (NAM, 2015) have been made available by NAM Groningen Development. Even though the faults are available in 3D, we restrain ourselves to a 2D representation of the faults at a specific depth, as they were picked along a reflector that was clearly visible in the seismic data. The fault locations above and especially below are decidedly less certain. A visualization of a 2D cut of the fault data is shown in Figure 16 (left). In Figure 16 (right), the grid centre locations are shown which are within 500 meters of the mapped faults.

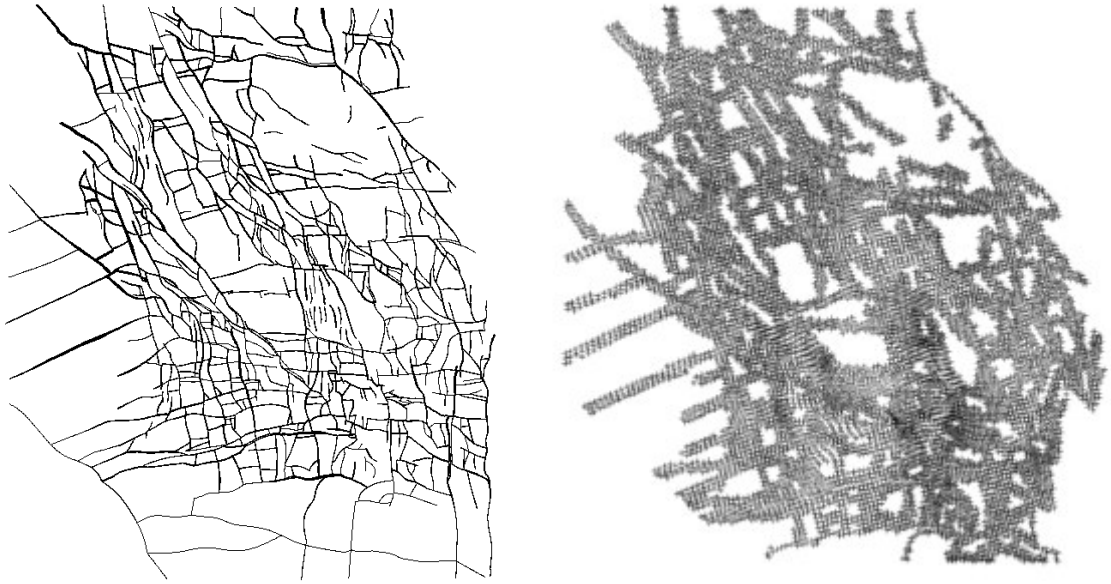


Figure 16: Left: interpreted faults in the Groningen Petrel model; right: grid centre locations which are within 500 meters of fault polygonal lines

3.7.2 Uncertainties

Due to typical resolution limits of seismic reflection imaging, faults with a throw[13] of 15 meter or less cannot be reliably identified and mapped, although such faults are probably still capable of generating an earthquake with magnitude 5 (Bourne & Oates, An activity rate model of induced seismicity within the Groningen Field (part 1), 2015). Although earthquakes with $M \geq 1.5$ seem to have a slight but statistical bias towards mapped faults in all likelihood many earthquakes originate from unmapped faults. This implies that features defined via fault maps will be biased towards large faults, despite that these faults are only responsible for a part of the seismicity. Work is underway to improve fault maps (Kortekaas & Jaarsma, 2017) and these improved fault maps could be included at a later moment in time.

3.7.3 Feature generation

The faults are exported as polygonal lines from which the distance to each of the grid cells is calculated. Based on that, only the grid cells which are within 500 meter of faults are kept. Subsequently, based on these grid cells only the dynamic reservoir features as described in section 3.4 are calculated.

3.8 Other Features

Some features are not directly derived from a single or a geological data source but are based on other data sources or use already integrated data elements to derive new features. These features are discussed here.

3.8.1 Temporal features

Two temporal features are implemented: (i) "chronological order" and (ii) "seasons". The chronological order is simply a counter, which starts at 1 and simply increases by 1 every timestep. This might be relevant in case of a strong monotonically increasing component in seismicity. The

[13] The fault throw is the vertical separation of layers on either side of the fault.

season divides the year in four periods of three months and might be relevant in case of potential seasonal seismic behaviour.

3.8.2 Global regional features

In case of a geospatial subdivision, since Groningen is considered to be a communicating vessel dynamics in one area might affect seismicity in another. To make best use of the available data and as an attempt to mitigate some of the potential caveats of having strict regional boundaries as geospatial bins would yield, we decided to optionally make the data for the entire Groningen field available for all subregions.

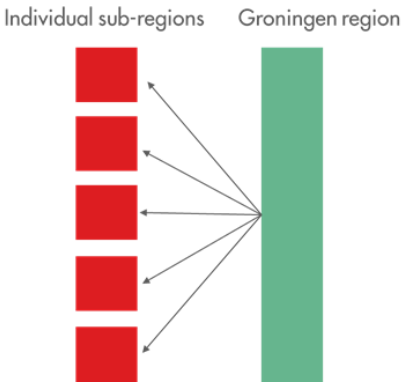


Figure 17: Diagram showing Global regional features logic

Furthermore, we have also implemented the option of creating a single dataset that contains all selected regions simultaneously and adds an additional identifier column for the region. This dataset can be in turn used during modelling. Whereas in the standard experimental setup each region is modelled individually, in the case of this realization of the data the region is just another feature to be used and as such an alternative that allows us to use "fixed" geological data like the faults as well as other spatially relevant elements that do not change over time.

In the current experimental setup no subregions are defined, as such global-regional features are not included in the current runs.

4 Methodology: Defining Meta Parameters

The previous chapter described the data sources used, their uncertainties and the features generated. These features are integrated into a tabular structure as shown earlier in Table 2, where each feature is a yellow column and the target is shown in the red column. Within our machine learning setup, each row of the table represents a "learning instance" for machine learning algorithms.

Prior to feeding the tabular structure of Table 2 to algorithms some additional processing steps can be applied to it in a parametrized way to strengthen the signal. The corresponding parameters can be seen as meta-parameters: they are not related to a specific algorithm but set the boundary conditions within which the algorithm has to work. This chapter describes these meta-parameters, in order of application to the tabular structure: time delays (section 4.1), smoothing (section 4.2), lags (section 4.3), the feature correlation threshold (section 4.4), data transformations (section 4.5) and the feature significance threshold (section 4.6).

4.1 Time delays

Earlier statistical work (section 2.2) suggests typical time delays between the change of an input variable and the impact of that on seismicity. For example, changes in production behaviour might impact seismicity directly but it might also take a certain amount of time. To investigate the potential improvement in predictive power we allow a time delay for each data source via the time delay meta-parameters for production Δi_Q, dynamic reservoir data Δi_{DRD}, subsidence Δi_S and compaction Δi_C. Concretely, let $z \in \{Q, RDR, S, C\}$ be the set of data sources as described in chapter 3, $x_i \in \mathbb{R}^m$ the m feature values at time instance i, x_i^z the subset of feature values based on data source z and Γ_{td_z} the time delay operator applied to (all time instances of) data source z, than the time delayed time series are given by:

$$\Gamma_{td_z}(x_i^z) = x_{i-\Delta i_z}^z$$

The effect of time delay meta-parameters on the tabular structure offered to the algorithm (see Table 2) can be visualized as "shifting the columns of the respective data source downward", see Figure 18.

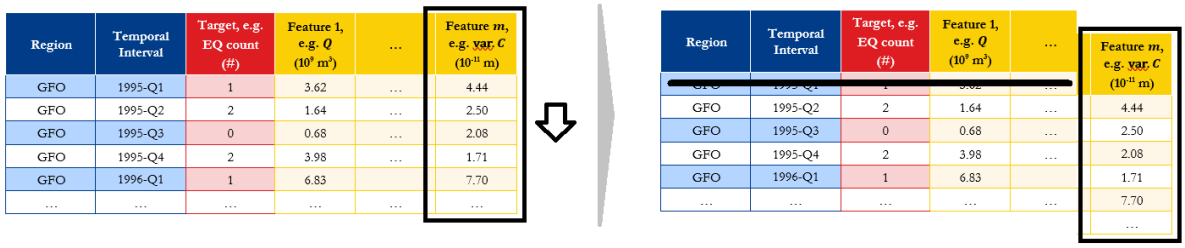


Figure 18: Visual illustration of impact of time delay meta-parameters on the tabular structure offered to the machine learning algorithms. For example the compaction time delay meta-parameter Δi_C "shifts all features derived from compaction data Δi_C rows downwards". Left the pristine tabular structure, right the resulting tabular structure for $\Delta i_C = 1$.

The value ranges to probe are based on the works referenced in section 2.2. As the references suggest time delays in the order of months for production, reservoir pressure and subsidence, for all time delay parameters a value range between 0 and 12 months has been explored.

We note two limitations with respect to the time delays. First, the physics interpretation of a time delay is not straightforward: earthquakes occur at faults which are spread throughout the reservoir. Unless there is a dominant fault which causes most of the seismicity (of which the authors are not aware), the information travel time from production site or reservoir location to faults (possibly resulting in seismicity) should have a wide range – hence there should not be a specific preferred

time delay. Second, the references which inspired the time delay meta-parameters suggest different time delays for variables and their derivatives (for example P and dP/dt have different time delays). However, to limit an exponential growth in our experimental setup we only allowed one time delay meta-parameter for the entire set of features derived from a data source (so P and $\Delta P/\Delta t$ have the same time delay).

4.2 Smoothing

Smoothing of data can potentially improve predictability since anomalies and outliers are smoothed out. When implementing smoothing it is important that the temporal structure of the data is honoured and no information leakage from the future to the past can occur, which implies that the smoothing with a window width $w \geq 2$ will be applied asymmetrically to historic data only.

Concretely, let $x_i \in \mathbb{R}^m$ be the m feature values at time instance $i \geq w$ and let Γ_s be the smoothing operator, than the smoothed values are given by (where "nts" stands for "no target smoothing"):

$$\Gamma_{s,\,\mathrm{nts}}(x_i) = \frac{1}{w}\sum_{j=0}^{w-1} x_{i-j}$$

Note that no values are calculated for intervals in time $i < w$ to make sure that that all data points are calculated in a consistent way.

When also the prediction target is smoothed with window width w the situation is slightly more involved since we again need to ensure that there is no information leakage. The prediction target at time interval $i < 2w$ gets calculated as outlined above however the smoothed feature indices need to be shifted such they are overlap free with the lookback period w. Hence, in this case the smoothed features values are given by (where "wts" stands for "with target smoothing"):

$$\Gamma_{s,\,\mathrm{wts}}(x_i) = \frac{1}{w}\sum_{j=0}^{w-1} x_{i-w-j}$$

4.3 Lags

The tabular structure shown in Table 2 provides the machine learning algorithms one instance of each feature to predict a target instance. For example, in Table 2 the algorithms will have to relate the seismicity of Q4 2015 with the production Q of Q4 2015, …, up to the compaction variance $var(C)$ of Q4 2015. Introducing time delays changes the timing of the feature instances (e.g. to Q3 2015) but it will not change that for each target instance only one instance of each feature is available to the algorithm.

The information travel time from production site or reservoir location to faults (possibly resulting in seismicity) probably has a wide range of values. Consequently, it might be that the target is not best predicted by only one (possibly delayed) instance of a feature but by multiple time instances of the feature. E.g. seismicity in Q4 2015 might be better predicted by having not just the production at Q4 2015 available, or just the production at Q3 2015, but by having both the production at Q4 and Q3 2015. To make information on a wider range of time available for each prediction instance, the lag parameter introduces the possibility to add multiple time delayed instances of to the feature set. More formally, let $x_i \in \mathbb{R}^m$ be the m feature values at time instance i and $\Gamma_{\mathrm{lag},z}$ the z-lag operator, than applying a lag of $0 < z < i$ will result in:

$$\Gamma_{\mathrm{lag},z}(x_i) = \begin{pmatrix} x_i \\ x_{i-z} \end{pmatrix} \in \mathbb{R}^{2m}$$

A visual illustration of the impact of the lag parameter is shown in Figure 19.

Region	Temporal Interval	Target, e.g. EQ count (#)	Feature 1, e.g. Q (10^9 m^3)	…	Feature m, e.g. var C (10^{11} m)
GFO	1995-Q1	1	3.62	…	4.44
GFO	1995-Q2	2	1.64	…	2.50
GFO	1995-Q3	0	0.68	…	2.08
GFO	1995-Q4	2	3.98	…	1.71
GFO	1996-Q1	1	6.83	…	7.70
…	…	…	…	…	…

Region	Temporal Interval	Target, e.g. EQ count (#)	Feature 1, e.g. Q (10^9 m^3)	Feature 1, Lag 1 (10^9 m^3)	…	…	Feature m, e.g. var C (10^{11} m)	Feature m, Lag 1 (10^{11} m)
~~GFO~~	~~1995-Q1~~	~~1~~	~~3.62~~	…	…		~~1.71~~	
GFO	1995-Q2	2	1.64	3.62	…	…	2.50	4.44
GFO	1995-Q3	0	0.68	1.64	…	…	2.08	2.50
GFO	1995-Q4	2	3.98	0.68	…	…	1.71	2.08
GFO	1996-Q1	1	6.83	3.98	…	…	7.70	1.71
…	…	…	…	6.83	…		…	7.70
				…		…		…

Figure 19: Schematic illustration of impact of the lag meta-parameter on the tabular structure offered to the machine learning algorithms. For illustrative purposes, left the pristine tabular structure, where for the target instance at Q4 1995 the machine learning algorithms only have access to all features instances at the same time instance. Right the tabular structure with lag = 1, where Q4 2015 can be predicted using two time instances of each feature (the Q4 1995 and the Q3 1995 instance).

Some variables are auto-correlated over longer timescales than the time intervals chosen. To ensure that lags added do provide additional information, only lags which are below the correlation threshold (section 4.4) are included.

4.4 Correlation threshold

Obtaining insights in the correlation between variables and features is an important exploratory analysis step in most statistical and machine learning approaches. For an overview of correlations within the context of this study see e.g. Figure 20 and Appendix 1. The correlation values presented in this section have been computed using the available data between 1995 and 2017. It is important to note that a second correlation analysis is automatically performed during the experimental setup to further discard features that are found to still be highly correlated, as could be the case for example with lagged features which might show high autocorrelation. Hence the correlation based feature selection is done twice, the first time offline for all features and a second time online each time the experiments are performed.

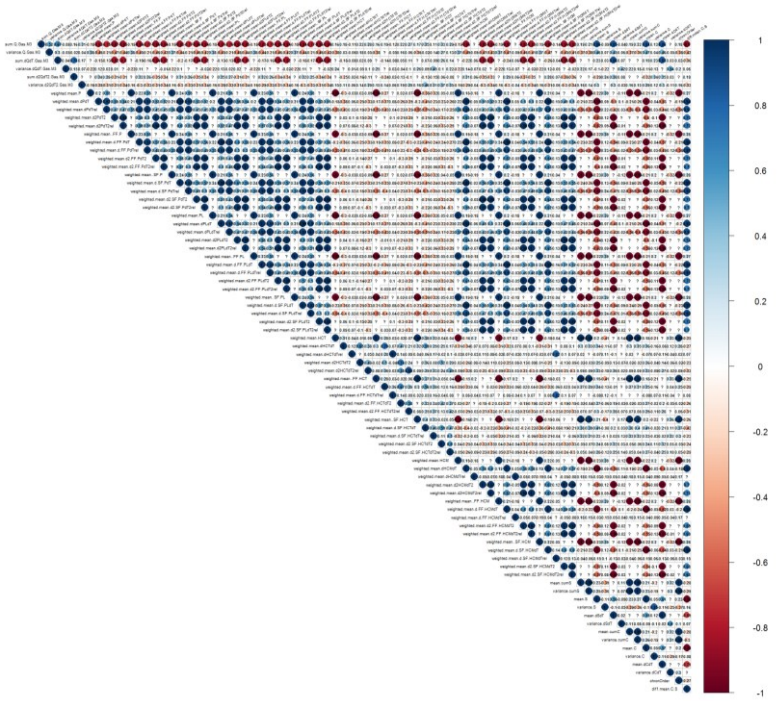


Figure 20: Example correlation matrix, where the horizontal and diagonal axes contain all features and the circles at the intersection show the correlation between two features. Correlation is colour coded from blue (highly correlated) via white (no correlation) to red (highly anticorrelated).

As can be appreciated from Figure 20 there are quite some features highly correlated with other features. This might get several machine learning algorithms confused, give numerical issues and/or deliver subpar results, mainly because for such algorithms those features would be indistinguishable in terms of added information.

Therefore, we perform dimensionality reduction by means of a correlation threshold: only features which have a correlation with other features below this threshold are included. We use a threshold of 0.9 based on two considerations: (i) a general acceptance of 0.75 (Udovičić, 2007) as being the starting point to consider 2 variables to be "highly correlated"; (ii) as we want to find subtle effects and as such not discard too many features early on, we heightened the general accepted threshold.

Dimensionality reduction is performed in two steps. First, several features have a relatively high autocorrelation, see Figure 21. To guard against this, only lags whose correlation with the unlagged feature is below the correlation threshold are taken forward.

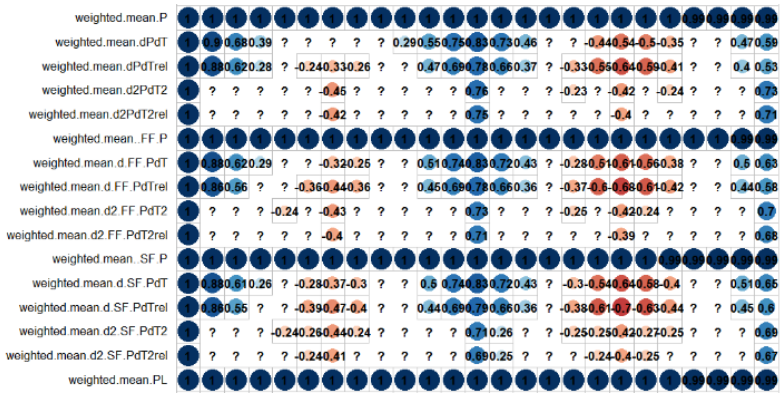


Figure 21: Sample lag correlation table. Each row is a feature (only a small part of the overall feature list is shown). For each feature, the first column shows the autocorrelation with one time interval delay, the second with two time intervals delay, etc. Colour coding is identical to Figure 21: from highly correlated (blue) via non-correlated (white) to highly anti-correlated (red). This table shows e.g. that the "weighted.mean.P" is highly correlated over the 24 time intervals shown, whereas its first and second derivate show a more seasonal pattern. A question mark indicates that the autocorrelation was not statistically significant.

Second, we have implemented a grouping function to group features that have a correlation equal or higher to the correlation threshold into a correlation group. Only statistically significant correlations are considered. From each resulting correlation group we select a representative feature. This is a manual process and as rule-of-thumb the least processed or transformed feature is chosen. For example, in the feature group shown in Table 9 we select "weighted.mean.P" as the representative feature since the others are either features that have been filtered using the faults or pertain to other derived variables like HCM. A full overview of all correlation groups can be found in Appendix 2.

Selected feature	Highly correlated features
Weighted.mean.P	Weighted.mean.FF.P Weighted.mean.PL Weighted.mean.HCM

Table 9: Illustrative example of feature group feature selection. All features in the table form a correlation group. The feature "weighted.mean.P" is chosen as it is the least processed feature.

Following dimensionality reduction we are left with a list of not-highly-correlated features, ensuring we are not taking forward redundant information that might impact our ability to identify

representatives of key drivers. In the steps beyond this point (e.g. variable importance) results about a feature should always be interpreted in the context that any feature in the same correlation group could be an equivalent replacement (e.g. model driver). Also note that correlation in general does not imply causation, meaning that latent not-included features could exist that are the physical drivers but which are highly correlated with the features that have been identified.

We note in particular the very high correlation between Compaction and Subsidence (0.998 at a monthly aggregate level), this is due to the fact that one is a linear model in the other. With actual compaction and subsidence measures the correlation might turn out to not be as strong but we are limited to use the model values. Given the choice between both, we decided to keep compaction and exclude subsidence from the feature list. Key reason for this choice is that compaction drives seismicity, see section 0. Additionally, to capture any remaining variability that might potentially be predictive of our target, the difference between Subsidence and Compaction s included as feature.

4.5 Data transformations

Certain data transformations can benefit certain algorithms by making the information contained in features better accessible than the raw values itself. First the data is normalized: it is centered and scaled and for some features divided by the cumulative sum. This is beneficial for some algorithms (e.g. regression), which otherwise might identify features with high values as more important purely based on their magnitude and not on their relative influence over the target.

Second, we investigate by visual inspection whether the signal could be strengthened by for example transforming to a logarithmic or square root distribution. When we plot our features over time (Figure 22) or look at their distribution (Figure 23) no skewness or trends are spotted that would make certain transformations a reasonable choice to try.

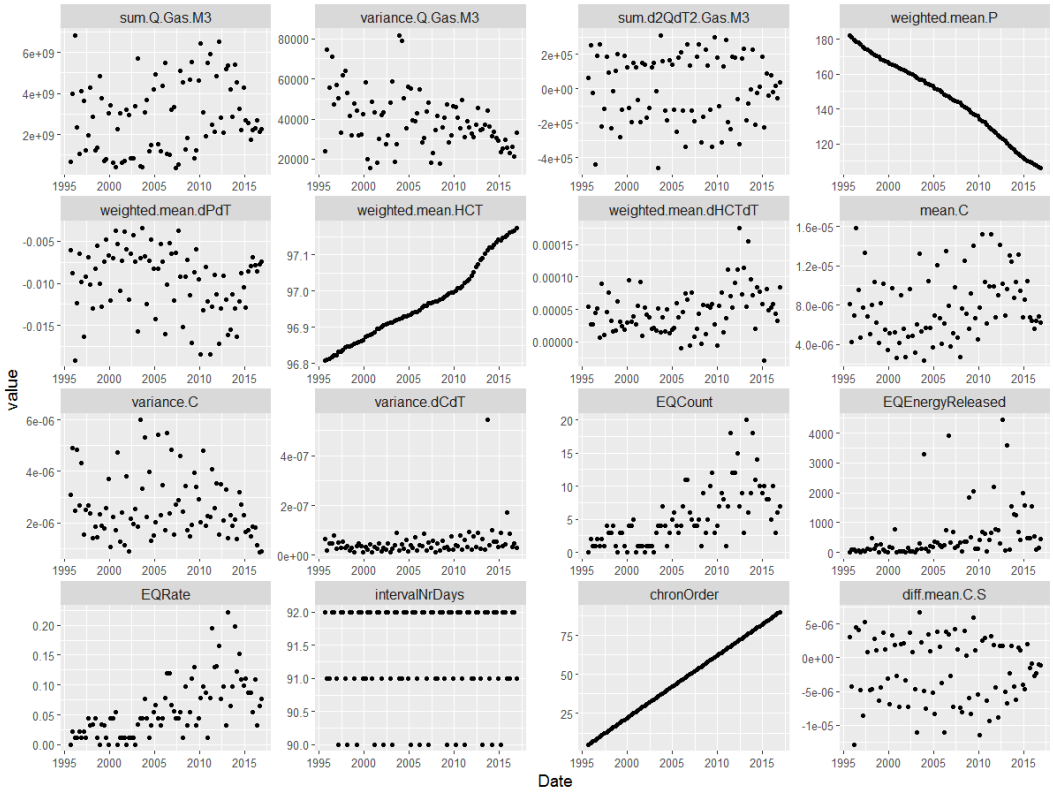


Figure 22: Feature time series of final set of features and targets for minimum magnitude 1.2 from 1995 to 2016 (excluding lagged features). No evident transformation might increase predictive performance.

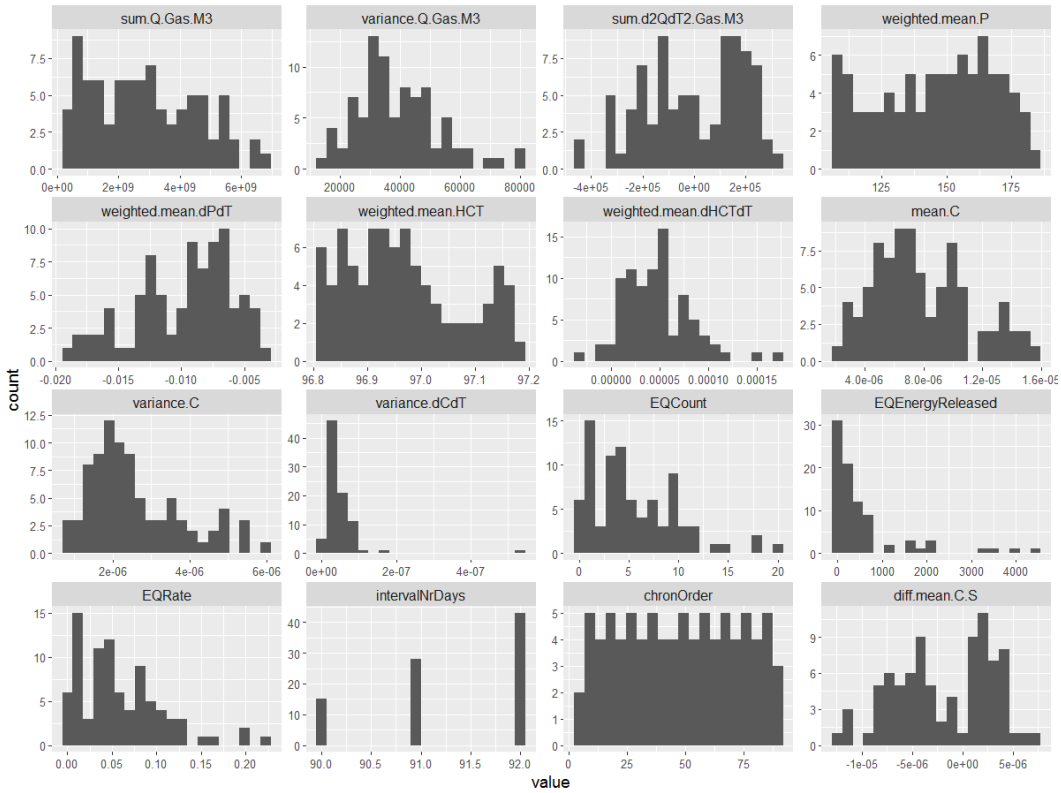


Figure 23: Feature distributions of final set of features and targets for minimum magnitude 1.2 from 1995 to 2016 (excluding lagged features). No evident transformation might increase predictive performance.

Third, some transformations might be helpful also in cases where no obvious transformation exists. We implemented two: Yeo-Johnson (YJ) and Principal Component Analysis (PCA). YJ is a monotonic transformation using power funtions, which e.g. makes the data more normal distribution-like and improves the validity of measures of association such as the Pearson correlation coefficient. It is a generalization of the Box-Cox transformation to the negative domain. PCA transforms a set of observations to a linearly independent (sub)set of variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (accounts for as much of the variability in the data as possible) and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. A disadvantage of PCA is that the principal components may not have any intuitive human interpretation.

4.6 Significance threshold

The features left over after correlation analysis contain limited redundant information but the information they contain might not be related to the target. The presence of features which are not relevant for the prediction task at hand might decrease predictive performance as coincidental correlations with the target could be seen as signal by the algorithms, whereas it is not. Therefore, as second feature reduction step only the features which cross a significance threshold are kept and the other discarded.

Not all theory required to explain how feature significance is calculated has been discussed at this point, we refer to section 7.2 for the methodology and Appendix 5 for the derivation. Here we suffice with the note that based on these sections the feature significance threshold is set to 0.4.

5 Methodology: Evaluating Model Performance

At the end of the previous chapter we ended up with self-consistent, tabulated data sets to which machine learning algorithms can be applied in order to obtain the corresponding models. Before discussing these algorithms in more detail in chapter 6, this chapter explains how we characterize the predictive performance of models.

It should be noted that different error metrics are in general not equivalent. Different metrics can provide different insights into the performance of the models and are therefore not redundant, but complementary to each other. Some metrics are in the units of the original data, and can therefore be quite easily interpreted, e.g. the mean absolute error (MAE) or the root mean square error (RMSE). However, the MAE and RMSE cannot be used to compare the performance of models across different prediction targets. Others, like the coefficient of determination R^2, are dimensionless and measure the amount of variance in the prediction target that can be explained by the model features enabling to some extent a comparison between different prediction targets. An overview of the error metrics that are considered in this report and the associated advantages and disadvantages is given in section 5.2.

Beyond the computation of the error metrics it is also imperative to estimate the standard error of the performance statistic, also known as its standard error, more will follow in section 5.3. It enables us to perform statistical hypothesis tests, that allow us to determine if model A outperforms model B in error metric m for a specified significance level. Section 5.4 will provide more details. To ensure models are chosen that perform well on historic and in expectation on future data, section 5.5 describes how we come to the minimum number of points needed to start predicting. Once a model is chosen and used for predictions, the uncertainty of the prediction needs to be quantified. In section 5.6 we outline how we approach this problem in the context of time series data. But first we explain in section 5.1 the basis of the above: a walk forward evaluation strategy.

The whole framework for model performance evaluation that is described in this chapter needs to be seen in the context of how the initially large set of models is benchmarked and reduced to a much smaller set of experiments to which eventually statistical hypothesis tests are applied. The iterative procedure based on model meta analysis, the underlying rationale and its potential pitfalls are explained in chapter 8.

5.1 Evaluation strategy - Walk Forward Testing

When testing the predictive power of a model it is important that the performance of the model is tested on data that has not been used to train the model. Due to overfitting to the given training data, model performance is in general always better in sample than out of sample. For this reason, the data set needs to be partitioned into training and test sets, where the model performance is only estimated on the test sets. If a given data set has sufficiently many data points (the exact number will depend on several factors like the complexity of the problem, the complexity of the model and the signal to noise ratio in the data) stable estimates of model performance can be obtained using only few training and test splits. Typically, around 70% of the data would be used for training and the rest would be reserved for testing, see (Friedman, Tibshirani, & Hastie, 2009) pp 222 for a more detailed discussion. In the case of a sufficiently large data set the estimated model performance is insensitive to the chosen partition. For smaller data set sizes and complex problems, like the case considered in this study, the estimated model performance is more uncertain and depends to a certain extent on the chosen partition. To minimize the effect of the chosen partition on the computed error metric a common approach is to repeat the modelling experiments on several training and test partitions of the data to obtain more stable error estimates and to be able to bound the uncertainty introduced through the different partitioning schemes. If the prediction target would be independently and identically distributed (i.i.d.) at different moments

in time several resampling schemes like k-fold cross-validation or several different non-blocked flavours of the bootstrap, would be available, see chapter 7 in (Friedman, Tibshirani, & Hastie, 2009) for further details. However, since we are dealing with time series data for which the i.i.d. assumption in general does not hold, those techniques bear the risk of overestimating the predictive performance of the models by leaking future information. Additionally, violation of the i.i.d. assumption can result in too small estimates of standard errors, which in turn could lead to Type 1 errors in hypothesis testing when testing two models for equivalence. The severity of the issues grows with increasing violation of i.i.d.-ness.

Therefore, we use a technique called Walk-Forward Testing, see p. 548 in (Kirkpatrick & Dahlquist, 2010), which is commonly used for back testing algorithms when dealing with time series data as it arises, for instance, in the financial industries. Back testing of models is also common in other disciplines that are concerned with forecasts like meteorology and climatology but are referred to as hindcasting. Models are conditioned to historical data available at an initial moment in time where data of sufficient quality is available, then a forecast is created over a specific time interval, after which the model is reconditioned and the procedure repeated. The quality of the model is assessed over the forecasting periods that have not been used to train the model. Depending on the actual application and the availability of the data the forecasting periods after which the models are updated can differentiate from hours (meteorology) to years or even decades (climatology), see for instance (Robert Fildes, 2011). The methodology of walk forward evaluation honours the time series nature of the data, such that no data in the test set is younger than any data point in the training set. Note that this would not be true for normal k-fold cross-validation or conventional bootstrap resampling schemes.

Let $n > 0$ be the number of data points in our data set. We denote the data point at time instance i by $d_i = (x_i, t_i)$, where $x_i \in \mathbb{R}^m$ are the m-covariate values that are available at time instance i and t_i is the prediction target at time instance i (e.g. the earthquake rate or count). Without loss of generality we assume that all categorical variables have been appropriately encoded as real numbers. Let $k \geq 0$ be the minimal number of samples that are required to train the (machine learning) model, see section 5.5 for a discussion on how k is found. Furthermore, let $1 \leq l \leq n$ be the forecast step size, i.e. the number of predictions that are generated before the model is updated. Then the Walk-Forward Testing approach works as follows (in pseudo code) for a given model f:

1. Let $i = k$.
2. Train model f on $(d_1, \dots, d_i)$, such that $f(x_j) \approx t_j$ for $1 \leq j \leq i$.
3. Let $p_{i+1} = f(x_{j+1}), \dots, p_{i+1+l} = f(x_{j+1+l})$ be the prediction of the model trained in step 2.
4. While $i \leq n - 1 - l$ let $i = i + l$ and goto 2.

An illustration of the approach is contained in Figure 24.

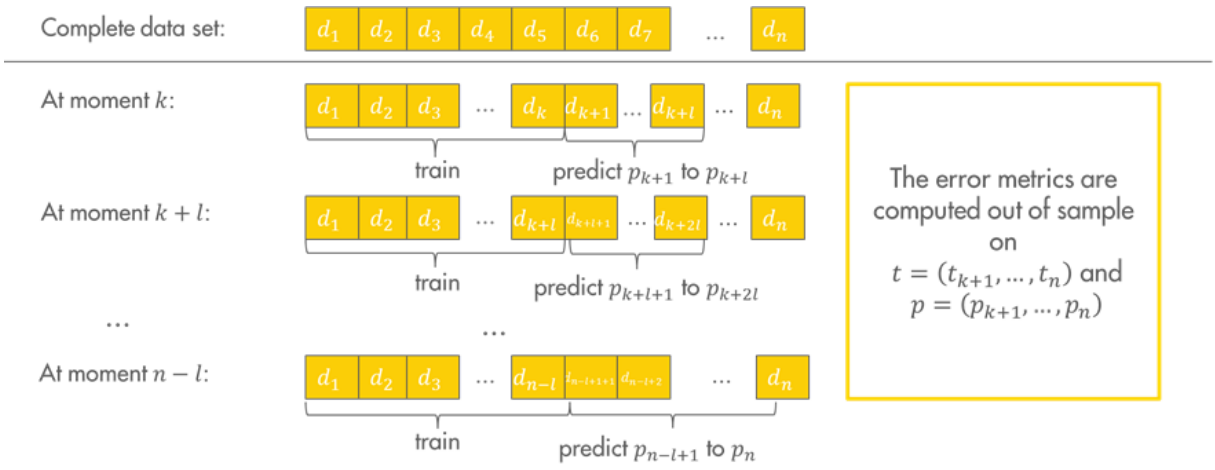


Figure 24: Illustration of Walk-Forward Testing for time series data when forecasting l time steps ahead

After the walk forward algorithm has terminated, we have a vector of predictions for time instances $k+1$ to n, namely p_{k+1} to p_n. Those, paired with the true values t_{k+1} to t_n can now be used to evaluate the performance of the algorithm f using one of the error metrics which we will introduce in section 5.2. We note that this approach implicitly assumes that models that perform better than others on short term forecasts also do so on longer term forecasts. In theory also longer term forecasts could be used for relative model performance evaluation, however as uncertainty increases with time longer term forecasts are in general more difficult to differentiate from each other. Using short term relative forecast differences increases the differentiative power.

5.2 Error Metrics/Measures

Once we have performed a prediction experiment and have predictions p_{k+1} to p_n we can quantify how different these predictions are from the true values t_{k+1} to t_n using some error metric m.

Note that error metrics, which are also called error measures, need not be metrics in the usual mathematical sense. Error metrics which take the form of means or expectations can be defined through a pointwise "loss function" $l: \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$, which is defined for individual pairs of prediction and truth, together with an aggregation function that combines the pointwise losses. For instance, the mean absolute error (MAE) and the Mean Poisson Loss (MPL) are the mean aggregates of the absolute and the Poisson loss. Other error metrics like the coefficient of determination R^2 exist which are not derived directly from a loss function.

The error measures that we have implemented in this study are defined and further explained in Table 10. For further details regarding these and other common error measures, see for instance (Bishop, 2007).

Err. Metric	Formula	Properties
MAE	$\frac{1}{n-k}\sum_{i=k+1}^{n}\lvert t_i - p_i\rvert$	• Result is in the unit of the original data • Uniform weighting of differences, thus less sensitive to outliers
RMSE	$\sqrt{\frac{1}{n-k}\sum_{i=k+1}^{n}(t_i - p_i)^2}$	• Smaller errors get less and larger errors get more weight • Result is in the unit of the original data • Potentially sensitive to outliers
R^2	$1 - \frac{(\sum_{i=k+1}^{n}(t_i - p_i)^2)}{(\sum_{i=k+1}^{n}(t_i - \bar{t})^2)}$	• Dimensionless error measure, ≤ 1. A value of 0 indicates a prediction that is equally good to predicting the mean. Value of 1 indicates perfect fit. • Allows comparison of predictability of different prediction targets. • Related to Pearson correlation coefficient r for linear models and in sample fit
MPL (Mean Poisson Loss)	$\frac{1}{n-k}\sum_{i=k+1}^{n}(p_i - \log(p_i)\, t_i + \log(t_i!))$	• Error metric specific to count data • Hard to compute in FP-arithmetic for large values of t_i if implemented naively • Special handling for the case $p_i = 0$ required
RMSLE	$\sqrt{\frac{1}{n-k}\sum_{i=k+1}^{n}(\log(t_i + 1) - \log(p_i + 1))^2}$	• If t_i is large, deviations from p_i have less weight than if t_i is small. • Commonly used for count data • Applicable to $p_i \geq -1$ and $t_i \geq -1$

Table 10: Overview of the implemented error metrics

Out of the error measures that we consider, both the MAE and RMSE are in the units of the actual data. Hence the results can be interpreted and compared in a rather straightforward way if only one prediction target is considered. MAE and RMSE are also complementary in the sense that the RMSE penalizes larger deviations in the residuals more heavily compared to the MAE, where all deviations receive uniform weight. Hence, the RMSE may be more sensitive to outliers. However, both error measures do not allow a comparison across different prediction targets, since by just looking at the individual RMSE and MAE it is unclear how much of the variance in the data is actually explained by the model. For this reason, we also include the dimensionless coefficient of determination R^2, which can be interpreted as the amount of variance explained by the model. In order to further improve the interpretability of results we do, on a case by case basis, complement the MAE and RMSE results by providing the respective error measures normalized by the mean of the prediction target over the whole test period under consideration. This as a substitute for error metrics like the mean absolute percentage error MAPE or similar error metrics that apply pointwise normalization by dividing through the value of the prediction target; such error metrics cannot be computed as our prediction target assumes the value of 0.

The coefficient of determination R^2 is in the range of $(-\infty, 1]$. It is a common misconception that R^2 only takes values in the interval $[0,1]$. This is only true if the coefficient of determination is computed in sample and for a linear model whose coefficients have for instance been fit using the ordinary least squares (OLS) approach. Both of these conditions are not satisfied in our setting.

In addition to these more traditional error measures mentioned above, we also investigate error measures that have been proposed in the literature specifically for count data, consider (Czando, Gneiting, & Held, 2009) for an overview. Currently we have implemented two of this particular type. The first one is the root mean square logarithmic error. It penalizes deviations of p_i from t_i more heavily if t_i is small and penalizes them less heavily if t_i is large. This is a sensible way of measuring model performance if the variance of the distribution is proportional to its mean. The second error measure of this type is derived from the Poisson loss, which is also in particular suitable for assessing model performance regarding count type predictions. While it is true that the assumption of independence of events is not true for earthquake data, the count data comes close to being Poisson distributed once aftershocks have been filtered, see for instance (Thalia Anagno, 1988). The mean Poisson loss is defined as

$$\frac{1}{n-k}\sum_{i=k+1}^{n}(p_i - \log(p_i)\,t_i + \log(t_i!)).$$

This equation is obtained by taking the mean of the negative log of the probability mass function of the Poisson distribution, $\frac{\lambda^{\alpha}e^{-\lambda}}{\alpha!}$, with $\lambda = p_i$ and $\alpha = t_i$. It should be noted that the out-of-sample Poisson loss is equivalent to the negative likelihood of a time inhomogeneous Poisson earthquake rate model, with piece-wise constant rate, where the constants are provided by the output of the ML - model(s). Furthermore, it should be noted that the computation of $\log(t_i!)$ can become numerically unstable if implemented in a naïve way. We have used the log factorial function provided by R which does not suffer from numerical instabilities. Additionally, we have made sure that the Poisson loss is always well defined by enforcing that the p_i are never below a small positive threshold of 0.0000001. The parameter was chosen through several experiments to make sure that the impact on the value of the error metric is minimal.

Since properly analysing the model results in all error metrics that we have implemented in the context of our factorial approach with subsequent downselection would have consumed a considerable amount of time we had to focus our analysis on a particular subset of error metrics, namely the MAE which is a typical and widely used error metric in machine learning and the RMSLE which is more commonly used for count type data. While we checked for a few cases that the ranking of algorithms with their respective settings is highly consistent between different error metrics a more detailed analysis should follow in a later follow up study. For reference we present for one of the targets all error metrics in Appendix 6.

5.3 Estimation of Standard Error for Error Metrics

Only having computed the values of the individual error functions for each experiment is not sufficient to assess if one method is significantly better than an alternative method with respect to a certain confidence level. For this reason, we also need to be able to estimate the standard deviation/standard error that is associated with each error measure. While explicit formulas are available for some error measures like the MAE, we chose a general approach, that can be applied to most computable error measure to assess the standard error (SE) that is associated with it.

In order to estimate the standard deviation of an error measure m, we make use of a technique called jackknife resampling. It is closely related to other resampling techniques like the bootstrap, but due to its deterministic nature, leads to reproducible results. Furthermore the bootstrap is a more general tool that can be used for both variance and distribution estimation. However, this versatility comes with significant computational cost compared to techniques like jackknife resampling. Since we are primarily interested in variance estimation and empirical studies also show that jackknife resampling outperforms the bootstrap for his particular purpose, see (Shao, 1995) pp. 281, it is our method of choice. In particular the jackknife estimate of variance is consistent for

sample means and correlation coefficients which covers the error metrics mentioned in Table 10, which we have been using in this study. Further details and references are contained for instance in (Efron & Stein, May 1981).

Let $t \in \mathbb{R}^n$ and $p \in \mathbb{R}^n$ again denote the true and predicted values and let m be an error measure. Let us further denote by $t_{-i} \in \mathbb{R}^{n-1}$ and $p_{-i} \in \mathbb{R}^{n-1}$, respectively t and p from which the i-th entry has been removed. I.e. $t_{-i} = (t_1, t_2, \dots, t_{i-1}, t_{i+1} \dots, t_n)$ and $p_{-i} = (p_1, p_2, \dots, p_{i-1}, p_{i+1} \dots, p_n)$. Then the standard error that is associated with m for a given instance of t and p is defined as

$$SE_m = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}\left(m(t_{-i}, p_{-i}) - \sum_{j=1}^{n}\frac{m(t_{-j}, p_{-j})}{n}\right)^2}.$$

In the presence of significant auto-correlation in the series $m(t, p)$ correlation correction needs to be applied to avoid underestimating uncertainties. Assuming stationarity, the adjusted formula to estimate the standard error is then given by

$$SE_m = \sqrt{\frac{1+\rho}{1-\rho}\frac{n-1}{n}\sum_{i=1}^{n}\left(m(t_{-i}, p_{-i}) - \sum_{j=1}^{n}\frac{m(t_{-j}, p_{-j})}{n}\right)^2},$$

where ρ is an estimate of the auto-correlation coefficient obtained for instance via the Praise Winsten estimation procedure, see (Bence, 1995). For mean based error measures this corresponds to the usual correlation adjustment of the sample error. Next, we will explain how we use the standard error for hypothesis testing and for grouping of models.

5.4 Comparing Model Performance via Hypothesis Testing

Once we have computed the values for the error measures with standard errors for the methods we want to investigate, we can perform hypothesis tests to determine if model A is either equivalent/better/worse to model B at a certain significance level for a chosen error measure. In this study, we will test whether the performance of selected machine learning models (sections 6.2 to 6.8) is statistically significantly better than simple "back-of-envelope" baseline models (section 6.9). Since several different hypothesis tests exist which differ in their requirements and specific properties we start with a brief general overview of the techniques that are available and commonly used. Then we proceed with matching the requirements of the tests with our situation to determine which tests are most suitable.

To choose the appropriate hypothesis test, several aspects of the underlying data set need to be considered which will be explained in the following. Based on the overview of the relevant aspects we try to match those requirements to our study and point out which tests need to be performed on the actual data to arrive at a suitable methodological decision. Since those additional tests are data specific they will only be described here and eventually be carried out in the experimental section 9.5. Still it should be noted that the sequence of tests carried out on different experimental setups and hence different datasets has led to a consistent choice of hypothesis tests across those experimental setups such that there can be no concern about an implicit bias:

(i) Parametric or non-parametric tests?

Parametric tests make distributional assumptions like normality on the statistic under consideration, non-parametric tests don't. For data sets where a priori the assumption cannot be made that the distribution is normal, tests like the Shapiro Wilk test should be

applied to verify that the assumption of normality cannot be rejected (at a given significance level). Based on several Monte Carlo simulation runs, see (Razali & Wah, 2011), this test has most statistical power for a given significance level compared to alternative tests like Anderson-Darling and Kolmogorov-Smirnov. In order for a test like Shapiro-Wilk to produce meaningful results a minimal sample size of around 30 is recommended, compare (Hogg & Tanis, 2005). Hence when the sample size is small and it cannot be safely assumed that the sample was drawn approximately from a normal distribution non-parametric hypothesis tests should be preferred. However, when the requirements for a parametric test are fulfilled the tests in general possesses more statistical power. Examples of parametric tests are variants of the t-test. Similarly, variants of the Wilcoxon test are examples of non-parametric tests. See (Dalgaard, 2008) for additional information.

Practical implication: in going forward we test if the error estimates are approximately normally distributed for both method A and B using the Shapiro-Wilk test, in conjunction with an analysis of significant outliers. In case outliers are detected and/or the null hypothesis of normality must be rejected we proceed with non-parametric tests.

(ii) Paired or unpaired tests?

Paired tests can be applied to data sets where a natural association between the two data sets exists to check if their location means/medians differ. Paired tests have higher statistical power compared to their unpaired counterparts due to reducing variance. However, they possess a potentially higher false positive (type 1 error) rate. Further details are contained in (Anderson, Kish, & Cornell, 1980). The requirements to apply a paired hypothesis test are satisfied since method A and method B are applied to exactly the same (input) data points – hence if different prediction strategies predict the same test label, they are naturally paired through which label they predict. Eventually we are trying to determine which of these techniques produces more accurate estimates. The choice then becomes a trade-off between gains in statistical power and keeping type 1 errors under control. It should be noted that an elevation in type 1 errors can potentially lead to falsely declaring method A to be statistically significantly better than method B. Nevertheless, the use of paired non-parametric tests for a pairwise comparison of two regression methods A and B is also suggested in literature in (Magdalena Graczyk, 2010). Pairwise tests in the context of comparing forecasts are both recommended in (Mariano, 1995) and (Timothy D., 2014). Furthermore, in the context of comparing machine learning models, paired tests are also discussed in (Bengio, 2003).

Practical implication: since we are interested in finding new leads in the relationships in the data that may have remained undetected till this moment in time we tend to favour paired sample tests in our interpretation. Also given that we are looking for subtle effects with small effect size that may have gone unnoticed so far and the fact that the sample size is small this choice is a deliberate one. Of course, proper scrutiny needs to be applied to the models which are obtained in this way to counter act type 1 errors. This could for instance take the form of assessing the partial dependence plots of the models to see if they are in conflict with first principles. For suitable model types like random forests, also the variable importance can be inspected to identify spurious results. We will perform and report on both paired and unpaired tests but focus on the interpretation of the paired test results for the reasons mentioned above.

(iii) Test samples independently and identically distributed (i.i.d.) or not?

Most of the common hypothesis tests have in common that the samples, i.e. the respective error estimates, to which the tests are applied need to be approximately i.i.d. Deviations

from the i.i.d. assumption may lead to an underestimation of uncertainty by overestimating the degrees of freedom and hence to an increase in Type 1 errors. One way to check if the i.i.d. assumption is violated is for instance via the non-parametric Wald–Wolfowitz runs test, consider (Magel & Wibowo, 1997) for more details. If a significant deviation is detected, for example in form of correlations between temporally adjacent samples, a correction to the conventional statistical tests should be applied (usually in form of a sample size adjustment). A detailed discussion of this particular topic is contained in (Zimmerman, 2012).

Practical implication: The Wald–Wolfowitz runs test is used to test for any major deviations from the i.i.d. assumption for the error estimates from both methods. The test expects as input a two valued data sequence to check that the elements of the sequence are mutually independent. We transform our time series of real valued predictions into a two valued one by computing the median of the data set and by assigning + to values larger than the median and – to values smaller than the median (values equal to the median are omitted). In addition to that we examine the auto-correlation of the series to assert that no statistically significant auto-correlation can be detected. If the hypothesis of i.i.d. ness needs to be rejected we proceed with correlation corrected versions of the respective hypothesis tests, if not we use the ordinary versions.

(iv) One-sided or two-sided test?

Two sided hypothesis tests check if there is evidence that method A and B are statistically significantly different from each other independently of the directionality of the relationship. A one sided test has more statistical power but explicitly excludes the possibility of an opposite relationship to the direction for which is tested. I.e. is method A statistically significantly better/worse than method B at a given significance level?

Practical implication: As stated above, the aim of this study is to be able to find potentially new leads based on subtle effects in the data. For that reason we again chose for the statistically more powerful version of the tests. The same caveats as mentioned under (ii) apply.

A comprehensive discussion of statistical two-sample location tests, including their specific requirements and underlying assumptions, is contained in (Fagerland & Sandvik, 2009). A high-level overview based on (Fagerland & Sandvik, 2009) is shown in Table 11.

		I.i.d.	**Non-i.i.d.**
Unpaired	**Normally distributed/ No outliers**	Welch's t-test	Welch's t-test with corr. Correction
	Non-parametric	Wilcoxon rank-sum test	Wilcoxon rank-sum test with corr. Correction
Paired	**Normally distributed/ No outliers**	Paired t-test	Paired t-test with corr. correction
	Non-parametric	Wilcoxon signed-rank test	Wilcoxon signed-rank test with corr. Correction

Table 11: Parametric and non-parametric location tests for two groups for parametric & non-parametric, paired & unpaired, i.i.d. & non-i.i.d.

We will now give a brief overview of how the tests mentioned above can be computed. Let A and B be two models for the same prediction task, p_A and p_B the corresponding predictions created by the model on the test data, m an error measure and l a lossction, than the associated tests can be described as follows:

- **Welch's t-test (unequal variance t-test):**
  We want to use Welch's t-test to test H_0 that A and B are equivalent with respect to m. First, we compute $m_A = m(t, p_A)$ and $m_B = m(t, p_B)$ together with the associated standard errors SE_{m_A} and SE_{m_B}. The test statistic t is defined as follows

$$t = \sqrt{n} \frac{m_A - m_B}{\sqrt{SE_{m_A}^2 + SE_{m_B}^2}}.$$

  In order to perform a hypothesis test, the distribution function of the test statistic t needs to be known or at least well approximated by a known distribution for sufficiently large sample sizes. For mean based error aggregations of normally distributed squared error or absolute error point measures, like the RMSE and the MAE, Student's t distribution is appropriate. In our setting, the effective degrees of freedom of Student's t-distribution can be estimated using the Welch-Satterthwaite equation:

$$\nu = \left\lfloor (n-1) \frac{\left(SE_{m_A}^2 + SE_{m_B}^2\right)^2}{\left(SE_{m_A}^4 + SE_{m_B}^4\right)} \right\rfloor.$$

  Now a one (larger/smaller) or two-tailed (unequal) t-test can be applied to test H_0 at the desired significance level. Note that we use the standard functionality that is available in R for this purpose, which can be called via the `t.test(…)` command. Further details can be found for instance in (Welch, 1947), (Dalgaard, 2008), (Fagerland & Sandvik, 2009) and (Ruxton, 2016)

- **Paired t-test:**
  By $r_A \in \mathbb{R}^n$ and respectively $r_B \in \mathbb{R}^n$ we denote the vectors of the evaluations of l for model A and model B. Now the paired t-test works as follows:

  Compute the difference between r_A and r_B, such that $d_i = e_{Ai} - e_{Bi}$.

  1. Compute the mean of d, such that $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$.
  2. Compute the standard error of $\bar{d}$ as $SE(\bar{d}) = \frac{SD(\bar{d})}{\sqrt{n}}$, where $SD(\bar{d})$ is the standard deviation of $\bar{d}$.
  3. Calculate $t = \frac{\bar{d}}{SE(\bar{d})}$.
  4. Under the null hypothesis of r_A and r_B having the same mean t will follow a Student's t distribution with $n - 1$ degrees of freedom.
  5. Use tabulated values of Student's t distribution vs. the value of t calculated in step 4 to obtain one or two tailed p-values.

  In order to perform the test we again use standard functionality provided by R in form of `t.test(…, paired = True)`.

- **Wilcoxon rank-sum test:**
  By $r_A \in \mathbb{R}^n$ and respectively $r_B \in \mathbb{R}^n$ we denote the vectors of the evaluations of l for model A and model B. Now the Wilcoxon rank-sum test works as follows:

  1. Combine r_A and r_B into one set and assign ranks (in ascending order) to the elements. Tied values are assigned to midpoint of the tied ranks.

2. Let R_A be the sum of the ranks for the samples that came from model A. Analogously we define R_B.
3. Then let $U_A = R_A - \frac{n(n-1)}{2}$ and let $U_B = R_B - \frac{n(n-1)}{2}$.
4. Let $U = \min(U_A, U_B)$.
5. Use tabulated values of U to obtain p-values for one or two sided hypothesis tests.

To perform the test we again rely on standard functionality in R in the form of the command `wilcox.test(…)`.

- **The Wilcoxon signed-rank test:**

  By $r_A \in \mathbb{R}^n$ and respectively $r_B \in \mathbb{R}^n$ we denote the vectors of the evaluations of the loss function for model A and model B. Now the Wilcoxon signed-rank test works as follows:

  H_0: The difference between the pairs follows a symmetric distribution around zero

  H_1: The difference between the pairs does not follow a symmetric distribution around zero

  1. For $i = 1$ to n calculate $|r_{A,i} - r_{B,i}|$ as well as $\mathrm{sgn}(r_{A,i} - r_{B,i})$
  2. Exclude pairs for which $|r_{A,i} - r_{B,i}| = 0$ and let n_r denote the reduced sample size.
  3. Order the remaining n_r samples in increasing order by $|r_{A,i} - r_{B,i}|$.
  4. Rank the pairs starting with 1. Ties receives equal rank as the average of the ranks they entail. Denote the ranks by R_i.
  5. Calculate $W = \sum_{i=1}^{n_r} (\mathrm{sgn}(r_{A,i} - r_{B,i}) R_i)$.
  6. Under H_0, W follows a specific distribution with expected value 0 and variance $\frac{n_r(n_r+1)(n_r+2)}{6}$.
  7. H_0 is rejected if $|z| > W_{\mathrm{crit},n_r}$, where the values of W_{crit,n_r} can be obtained from a reference table.

  Further details about the implementation in R, that we are using, and some additional theoretical considerations can be found in (Dalgaard, 2008).

  The function call in R is given by `wilcox.test(…, paired=True)`.

We employ hypothesis testing to establish if a specific machine learning model A, that has been selected via the procedure described in subsection 8.3, is statistically significantly better than a given baseline (model B). This means that we are not performing multiple hypothesis testing, as in "is at least one of the models better than the baseline", where appropriate corrections like the Holm–Bonferroni method would need to be applied. A detailed discussion of how to perform multiple hypothesis testing in a machine learning context for the purpose of model comparison of regression models is contained in (Magdalena Graczyk, 2010).

In expectation, predictions further in the future will have a larger forecast uncertainty and consequently larger prediction errors, making it harder to distinguish useful models from the naïve baselines due to an increase in overall variance. Hence, for the purpose of model selection we have made the deliberate choice to focus on forecasting one time interval ahead. We tacitly assume that a model that has competitive short term predictive performance will also be performing competitive to its peer models in longer term forecasts. The validity of this assumption will need to be further investigated in a future study once more data becomes available. We note that our workflow does supports forecasting more time intervals ahead, a capability we'll use in section 5.6.

5.5 Minimum Number of Training Points

Comparisons as made in section 5.4 can be biased in favour of algorithms that are "quick learners", and already provide reasonable estimates based on a smaller number of data points. It can be shown

that if algorithm A would outperform algorithm B in e.g. 60% of the cases when given enough data (as will be the case in future deployment of such an algorithm), but suffers a severe degradation of performance when trained on insufficient data points, algorithm B would be preferred according to our decision criteria, despite being the suboptimal choice going forward. On the other hand, increasing the minimal number of training points k that are required before walk forward validation starts will lead to a smaller test set size and to more uncertainty in the associated error estimates. Since it can be expected that the performance of any sensible machine learning technique will improve or at least not deteriorate with increasing k, choosing a small k will lead to conservative estimates of model performance.

To compromise between the aspects mentioned above we inspected the relationship between number of training points and model performance for each algorithm type, as measured by the median of the squared error, as well as its 10th and 90th percentile (to cater for algorithms that performed badly only for a few specific combinations of parameters). A cut-off was determined empirically at 8 points to ensure stable performance and a fair comparison across the board for various algorithms.

5.6 Forecast Uncertainty Quantification

As explained in the last paragraph of section 5.4, by default our forecasts are one time interval ahead. Consequently, the uncertainty estimates in the form of standard errors are by default only applicable for one time interval ahead – hence not multiple time intervals ahead as required for long term seismicity forecasting. Given the non-parametric nature of most of our algorithms there is no analytical derivation from which we can obtain longer term uncertainty estimates so we proceed to obtain such estimates using an empirical approach.

Let $h > 0$ be the number of points in the historical data set, let $1 \leq l$ be the forecast step size (i.e. the number of predictions that are generated before the model is retrained) and let k be the minimum number of points used for training the model. Furthermore, we denote the predictions of a walk forward run with forecast step size l by $p_i^{(l)}$ for $i \in \{k+1, \dots, h\}$. The associated prediction errors are denoted by $\delta_i^{(l)} = m\left(t_i, p_i^l\right)$ for a pointwise error metric m and t_i the true value at time interval i for $i \in \{k+1, \dots, h\}$. Since these estimates are all highly dependent on i we are stabilizing the results by estimating the forecast uncertainty z time intervals ahead $\bar{\delta}^{(z)}$ as the 10th/90th percentiles of the set of all $\delta_i^{(l)}$ for which the time interval between (re)-training the model and the actual prediction is equal to z. In order to obtain the required $\delta_i^{(l)}$ the calculations are performed in a block wise fashion for increasing forecast window sizes.

To quantify e.g. the uncertainty $\bar{\delta}^{(3)}$ of forecasting three steps ahead:

- We generate walk forward runs calculating iteratively:
  - three steps forward $\delta_i^{(3)}$ for $i \in \{4, \dots, h\}$;
  - four steps forward $\delta_i^{(4)}$ for $i \in \{5, \dots, h\}$;
  - five steps forward $\delta_i^{(5)}$ for $i \in \{6, \dots, h\}$;
  - etc.
- From these runs we select:
  - $\delta_{k+3}^{(3)}, \delta_{k+6}^{(3)}, \delta_{k+9}^{(3)}, \dots$
  - $\delta_{k+3}^{(4)}, \delta_{k+7}^{(4)}, \delta_{k+11}^{(4)}, \dots$
  - $\delta_{k+3}^{(5)}, \delta_{k+8}^{(5)}, \delta_{k+13}^{(5)}, \dots$
  - etc.
- The 10th/90th percentiles of the elements listed above are used to obtain $\bar{\delta}^{(3)}$.

An illustrative example is contained in Figure 25.

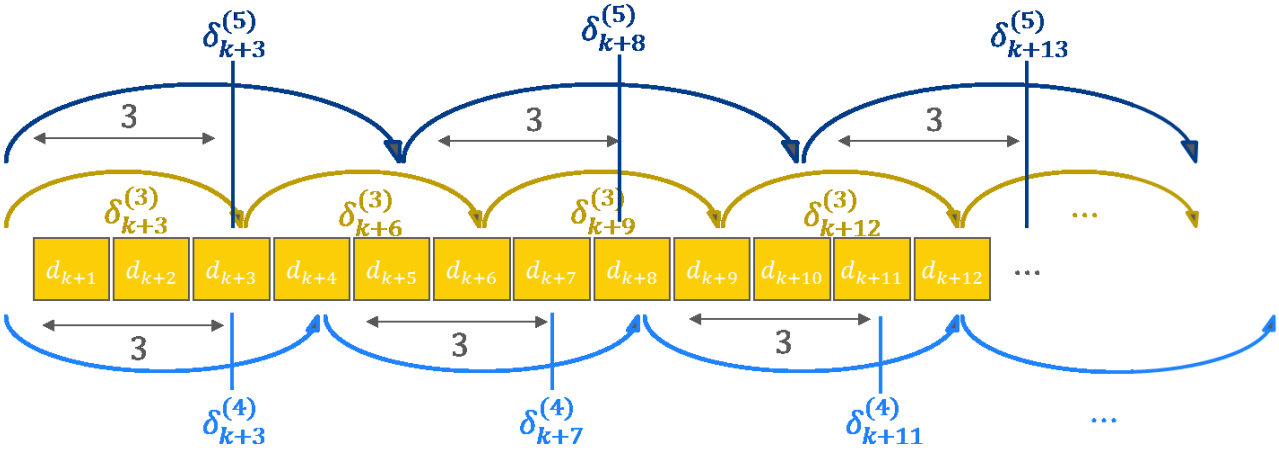


Figure 25: Illustrative example of how the uncertainty estimate for forecasting three steps ahead $\overline{\boldsymbol{\delta}}^{(3)}$ is derived from the 10th/90th percentiles of the set of all three step ahead forecasts.

Due to the variance of the estimation for the forecast errors, the empirical estimates can violate our theoretical assertion of a smooth, monotonically increasing error function with time. In view of that, we implemented an isotonic regression to ensure a monotonically increasing error. The lower confidence interval is bounded by 0 since we only consider non-negative prediction targets.

An alternative approach that could be considered in a future iteration would be to use the estimated standard errors instead of the bootstrapped percentiles which potentially exhibit high variance proportional to $1/(percentile\_density)^2$.

Due to the large number of experiments that are necessary we also note that obtaining empirical uncertainty estimates as described above is computationally demanding.

6 Methodology: Machine Learning Models

With the stage set in chapter 4 in the form of self-consistent, tabulated data sets to which machine learning algorithms can be applied, this chapter provides a high-level description of the key machine learning models (or algorithms) used in this study: Generalized Linear Models and variants (GLMs, section 6.2), K-Nearest Neighbours (KNNs, section 6.3), Random Forests (RFs, section 6.4), Support Vector Machines (SVMs, section 6.5), ARIMA models (section 6.6), Neural Nets (section 6.7) and GBM (section 6.8). The respective model hyperparameters, advantages and disadvantages can be found in Appendix 4.

Furthermore, to assess whether the beforementioned algorithms possess non-trivial predictive power regarding the target we do a statistical comparison of the predictive performance of our models against naïve baselines, described in section 6.9. Post-processing strategies of prediction results are described in section 6.10.

Please note that this overview does not intend to be exhaustive but merely introduce in broad strokes the main methods to those who have a cursory understanding or who are unfamiliar with machine learning but wish to know the basics. The interested and more advanced reader is encouraged to look into the available literature for a more complete and mathematical overview of the presented methods, e.g. "An Introduction to Statistical Learning" (James, Witten, Hastie, & Tibshirani, 2017) and "Applied Predictive Modelling" (Kuhn & Johnson, 2018).

6.1 Model Overview & Selection

Given the comparative scarcity of research using datasets of the type we have in this study, we did not commit to a specific algorithm or algorithm family a priori. Instead, as will be explained in chapter 8, we opt to empirically test and rank several types of algorithms to determine which model families work well in the given context through a benchmarking study. The candidate algorithms are loosely based on the work of (Delgado M.F, 2014), who have tested 179 different algorithms on datasets from the UCI Machine Learning repository (Bache & Lichman, 2013).

Rank	Acc.	κ	Classifier	Rank	Acc.	κ	Classifier
32.9	82.0	63.5	parRF_t (RF)	67.3	77.7	55.6	pda_t (DA)
33.1	**82.3**	**63.6**	rf_t (RF)	67.6	78.7	55.2	elm_m (NNET)
36.8	81.8	62.2	svm_C (SVM)	67.6	77.8	54.2	SimpleLogistic_w (LMR)
38.0	81.2	60.1	svmPoly_t (SVM)	69.2	78.3	57.4	MAB_J48_w (BST)
39.4	81.9	62.5	rforest_R (RF)	69.8	78.8	56.7	BG_REPTree_w (BAG)
39.6	82.0	62.0	elm_kernel_m (NNET)	69.8	78.1	55.4	SMO_w (SVM)
40.3	81.4	61.1	svmRadialCost_t (SVM)	70.6	78.3	58.0	MLP_w (NNET)
42.5	81.0	60.0	svmRadial_t (SVM)	71.0	78.8	58.23	BG_RandomTree_w (BAG)
42.9	80.6	61.0	C5.0_t (BST)	71.0	77.1	55.1	mlm_R (GLM)
44.1	79.4	60.5	avNNet_t (NNET)	71.0	77.8	56.2	BG_J48_w (BAG)
45.5	79.5	61.0	nnet_t (NNET)	72.0	75.7	52.6	rbf_t (NNET)
47.0	78.7	59.4	pcaNNet_t (NNET)	72.1	77.1	54.8	fda_R (DA)
47.1	80.8	53.0	BG_LibSVM_w (BAG)	72.4	77.0	54.7	lda_R (DA)
47.3	80.3	62.0	mlp_t (NNET)	72.4	79.1	55.6	svmlight_C (NNET)
47.6	80.6	60.0	RotationForest_w (RF)	72.6	78.4	57.9	AdaBoostM1_J48_w (BST)
50.1	80.9	61.6	RRF_t (RF)	72.7	78.4	56.2	BG_IBk_w (BAG)
51.6	80.7	61.4	RRFglobal_t (RF)	72.9	77.1	54.6	ldaBag_R (BAG)
52.5	80.6	58.0	MAB_LibSVM_w (BST)	73.2	78.3	56.2	BG_LWL_w (BAG)
52.6	79.9	56.9	LibSVM_w (SVM)	73.7	77.9	56.0	MAB_REPTree_w (BST)
57.6	79.1	59.3	adaboost_R (BST)	74.0	77.4	52.6	RandomSubSpace_w (DT)
58.5	79.7	57.2	pnn_m (NNET)	74.4	76.9	54.2	lda2_t (DA)
58.9	78.5	54.7	cforest_t (RF)	74.6	74.1	51.8	svmBag_R (BAG)
59.9	79.7	42.6	dkp_C (NNET)	74.6	77.5	55.2	LibLINEAR_w (SVM)
60.4	80.1	55.8	gausspr Radial_R (OM)	75.9	77.2	55.6	rbfDDA_t (NNET)
60.5	80.0	57.4	RandomForest_w (RF)	76.5	76.9	53.8	sda_t (DA)
62.1	78.7	56.0	svmLinear_t (SVM)	76.6	78.1	56.5	END_w (OEN)
62.5	78.4	57.5	fda_t (DA)	76.6	77.3	54.8	LogitBoost_w (BST)
62.6	78.6	56.0	knn_t (NN)	76.6	78.2	57.3	MAB_RandomTree_w (BST)
62.8	78.5	58.1	mlp_C (NNET)	77.1	78.4	54.0	BG_RandomForest_w (BAG)
63.0	79.9	59.4	RandomCommittee_w (OEN)	78.5	76.5	53.7	Logistic_w (LMR)
63.4	78.7	58.4	Decorate_w (OEN)	78.7	76.6	50.5	ctreeBag_R (BAG)
63.6	76.9	56.0	mlpWeightDecay_t (NNET)	79.0	76.8	53.5	BG_Logistic_w (BAG)
63.8	78.7	56.7	rda_R (DA)	79.1	77.4	53.0	lvq_t (NNET)
64.0	79.0	58.6	MAB_MLP_w (BST)	79.1	74.4	50.7	pls_t (PLSR)
64.1	79.9	56.9	MAB_RandomForest_w (BST)	79.8	76.9	54.7	hdda_R (DA)
65.0	79.0	56.8	knn_R (NN)	80.6	75.9	53.3	MCC_w (OEN)
65.2	77.9	56.2	multinom_t (LMR)	80.9	76.9	54.5	mda_R (DA)
65.5	77.4	56.6	gcvEarth_t (MARS)	81.4	76.7	55.2	C5.0Rules_t (RL)
65.5	77.8	55.7	glmnet_R (GLM)	81.6	78.3	55.8	lssvmRadial_t (SVM)
65.6	78.6	58.4	MAB_PART_w (BST)	81.7	75.6	50.9	JRip_t (RL)
66.0	78.5	56.5	CVR_w (OM)	82.0	76.1	53.3	MAB_Logistic_w (BST)
66.4	79.2	58.9	treebag_t (BAG)	84.2	75.8	53.9	C5.0Tree_t (DT)
66.6	78.2	56.8	BG_PART_w (BAG)	84.6	75.7	50.8	BG_DecisionTable_w (BAG)
66.7	75.5	55.2	mda_t (DA)	84.9	76.5	53.4	NBTree_w (DT)

Figure 26: Overview showing model rank on multiple datasets, reproduced from (Delgado M.F, 2014)

It is important to clarify however that the (Delgado M.F, 2014) study focused on the use of algorithms for classification which differs from the setup of the seismicity study described in these pages. A more recent study by (Makridakis, 2018) focused on benchmarking machine learning models against classic statistical methods for the task of forecasting, which is in line with the usage of machine learning in this study. We have expanded and elaborated on the benchmarking study in a number of ways both in how the validation of the algorithms is concerned and in the comparison with statistical baselines.

6.2 Generalized Linear Models

Generalized Linear Models (GLM) are an extension of "classical" ordinary least squares regression (OLS) (Nelder & Wedderburn, 1972). OLS tries to fit the parameter weights for the linear relationship between the features and the target. GLMs extend on this concept by allowing the target to exhibit error distributions that are not normally distributed. In this study we use GLMS and two GLM variants: (i) GLMnet, a GLM with elastic net regularization and (ii) GLMtop, a GLM model that has been trained using the top 5 most significant features (as per the variable significance analysis in section 7).

The GLMnet (elastic net) algorithm deals with the multicollinearity problem in the original feature space by applying dimensionality reduction. As the ratio between number of fitted coefficients and number of observations increases, the estimates for the coefficients incur more variance. By

applying the bias-variance trade-off, we can choose to introduce a controlled bias in our algorithm, to drastically reduce the variance of the estimates. We can do this by applying a technique called *regularization* (Friedman, Hastie, & Tibshiranie, 2010). Simply put, we can regularize the coefficients of the GLMs by applying a penalty for large components. Whereas a traditional GLM would seek to find the component weights such that a loss-function is minimized, regularized GLM allows for the optimization with regards to a loss function that is a weighted version of the sum of the absolute values of the coefficient weights (L1 norm), or the sum of the squared values of the coefficient weights (L2 norm). The latter technique is called *ridge regression*, while the former is referred to as *LASSO*. Elastic net regularization effectively combines both L1 and L2 types by using an additional alpha parameter to gauge the degree to which either should be implemented.

A GLM Top model with default hyperparameters has been trained only using a selection of the top 5 features. This model might has as advantage that is performance is not potentially degraded by less well performing features. No regularization has been implemented in this case.

6.3 K-Nearest Neighbours

K-Nearest Neighbours (KNN) (Hu, Huang, Ke, & Tsai, 2016) is a simple, yet effective, machine learning algorithm that makes use of the distance (by some chosen metric) between different observations in the dataset. Intuitively, observations that are more similar to each other regarding a subset of the features (the predictors), are increasingly likely to be more similar regarding another subset of the features (the target variable). This idea is formalised in the KNN algorithm, where classifications or predictions for a new (unseen) instance are based on a committee or aggregation of k most similar examples.

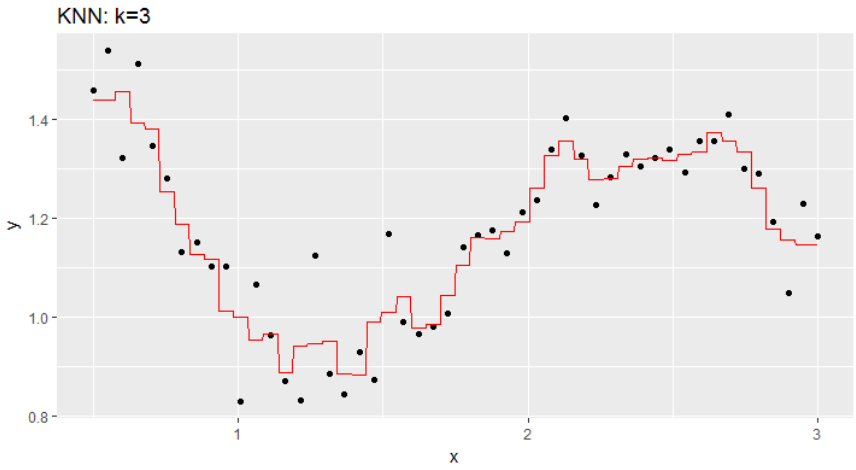


Figure 27: Example of KNN used for regression, showing the KNN predictions (red), the actual measurements (black dots). Adapted from (Kim, Kim, & Namkoong, 2016)

6.4 Random Forests

Random Forests (Breiman, Random Forests, 2001) have been used to very good results across a wide variety of tasks. Random Forests, at their core, represent an extension of decision tree algorithms using ensembles. Ensembles refer to a modelling technique where a decision or prediction is not produced by a single algorithm, but rather by a collection of them (Schapire, 1990). The use of this meta-modelling technique is not specifically limited to Random Forests, but can be applied to any base algorithm, or even collection of diverse algorithms.

The units within a Random Forest are known as Decision trees (Breiman, Friedman, Olshen, & Stone, 1984), renowned for their simplicity, clarity, and speed. These trees perform a recursive

partitioning of the data space with the objective of making the data partitions as homogeneous as possible. In study we use binary trees though other partitions are possible.

Decision trees do suffer from high variance. It is for this reason that we aim to decrease the variance of the solution by creating an ensemble of trees (Breiman, Friedman, Olshen, & Stone, 1984), rather than look at a single tree. Random Forests achieve this decorrelation in two ways: bootstrap sampling and restricting the set of candidates features to split on.

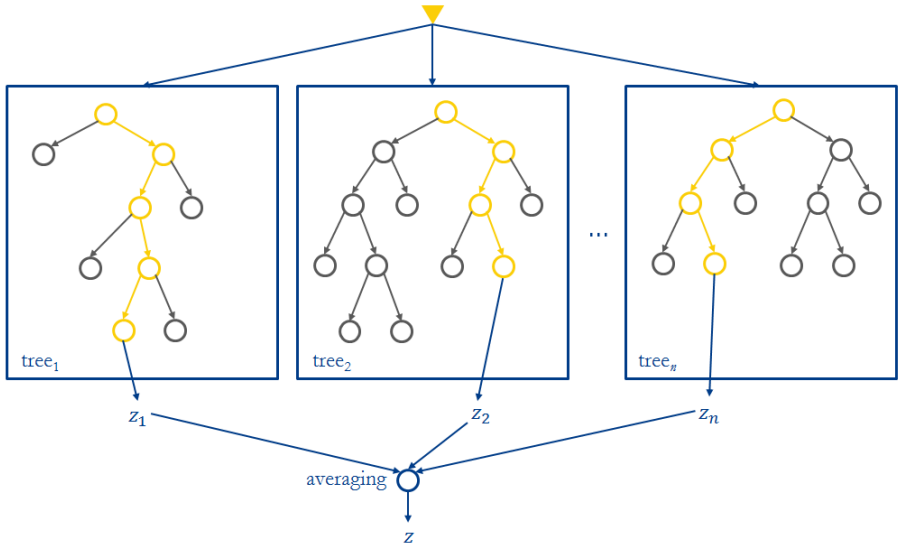


Figure 28: Illustrative example of how regression predictions of individual trees combine in a random forest through averaging of the prediction results of individual trees.

6.5 SVR

SVR's are non-probabilistic algorithms which can be considered extensions and generalizations of optimal separating hyperplanes that get defined by the data points closest to the decision boundary, which are referred to as support vectors (Cortes & Vapnik, 1995). SVR's extend on the concept of optimal separating hyperplanes in two ways:

- By accommodating the case of overlapping classes.
- By allowing nonlinear decision boundaries in the original feature space by employing the kernel trick.

In SVR's, the input is implicitly mapped onto a m-dimensional (where m can in fact be infinite, and in those cases not computable) feature space using some fixed (nonlinear) mapping (kernel trick), and then a linear model is constructed in this feature space (Friedman, Tibshirani, & Hastie, 2009). The main motivation is to seek and optimize the generalization bounds given for the regression. These bounds rely on defining the loss function that ignores errors situated within a certain distance of the true value. In other words, the goal is to find a function whose prediction deviates from the target value by an amount no more than ε.

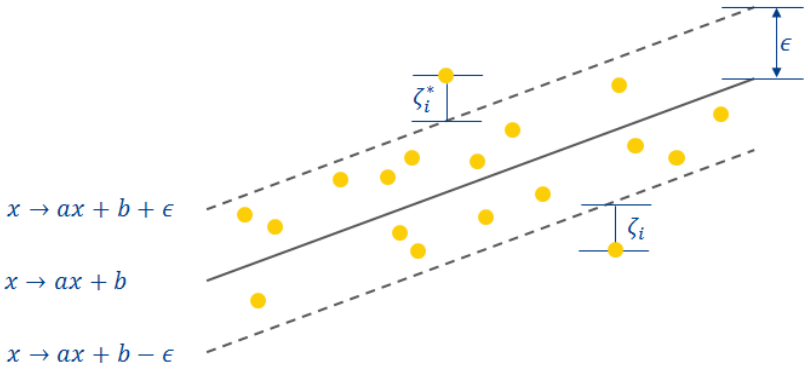


Figure 29: Example of SVM regression. Points within the pink band, where the prediction error < epsilon, don't contribute to the total loss of the function. Outside of this band are the support points that determine the parameters of the functions.

6.6 ARIMA

ARIMA (AutoRegressive Integrated Moving Average) models aim to integrate a time-dependent structure into the modelling process. It does so in three ways: *autoregression, integration* and *moving averages*. The AutoRegression component of ARIMA models refers to the inclusion of lagged input values in the model. The Moving Average component refers to the inclusion of past forecasting errors (multiplied by a coefficient weight). The Integration component indicates that this model does not necessarily have to be built on the original input values of the time series, but can also be trained on a series to which we have applied one or more differencing steps, often to remove non-stationarity in the series.

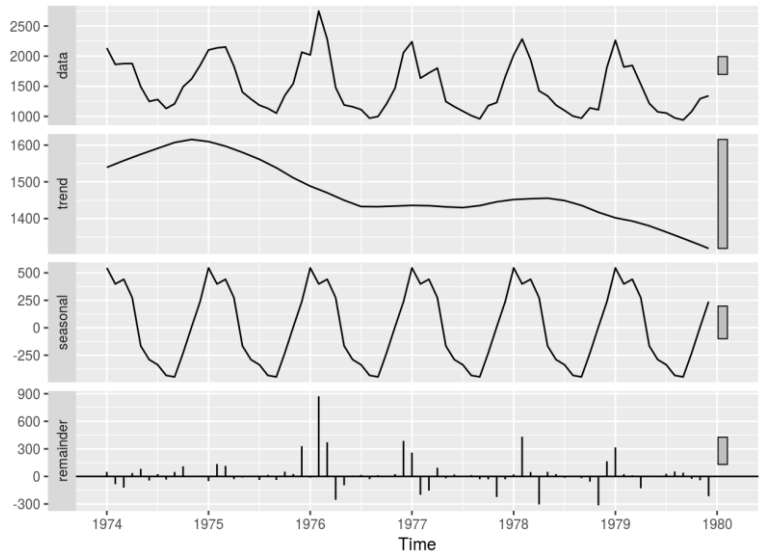


Figure 30: Example ARIMA output showing trend and seasonal decomposition

6.7 Neural Networks

An Artificial Neural Network (ANN) (Hopfield, 1988) is a machine learning technique that draws inspiration from neurobiology and computational neuroscience, where a learning system is loosely modelled on the inner workings of the (human) brain. The model that is being trained is comprised out of several interconnected nodes; the values of individual nodes gets determined by the value of their preceding connected nodes, the weights of these connections, and a nonlinear activation

function that transforms the sum of these incoming values. On a mathematical level the model would "learn" by updating the weights of the connections between the nodes.

This process of updating the connection weights generally happens using a technique called *backpropagation,* which calculates the gradient of the error function with respect to the network's weights. This happens "backwards", starting at the output layer, gradually working its way back to the input layer, hence the name backpropagation. These gradients are then used to proportionally update the weights of the network.

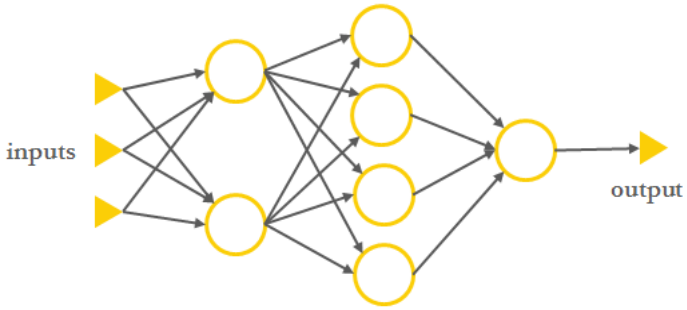


Figure 31: Visual representation of a simple Feedforward Neural Network with 2 hidden layers.

6.8 Gradient Boosting Machines (GBM)

Gradient Boosting Machines (GBM) (Hastie, 2007) much like Random Forests are an example of constructing a potentially strong learner out of a collection of weak learners (that is, learners that are only initially only marginally better than random chance) these weak learners are usually decision trees, as is the case in random forests, but they could also be simple linear models like GLMs. The main characteristic of GBM's is that GBMs make use of a boosting paradigm (as their name implies), primarily aimed at reducing bias. Boosting means that subsequent learners give extra attention to previously wrongly estimated examples. This means that learners are added iteratively and the weights for the wrongly estimated observations are used by the new learner to improve the prediction.

6.9 Simple Baselines

To assess whether machine learning algorithms as described in previous sections possess non-trivial predictive power in their application to induced seismicity in Groningen, we do a statistical comparison of the predictive performance of our models against naïve baseline models. By a simple model we mean a "back-of-envelope" model that only use time series data of past measurements and/or very simple physics. While also ARIMA type models (see section 6.6) only use prediction target time series data, we don't classify these as simple baselines since the underlying structure of the model and the techniques to determine a suitable form and fit the corresponding parameters are rather involved.

If the best machine learning based model would fail to beat the best simple baseline in a statistically significant way that would imply that there is no evidence that the advanced machinery described above adds value when it comes to Groningen induced seismicity predictions, given the data currently available.

In total, we have implemented five naïve baselines, which we will now detail. Let t_i be the true value of the respective prediction target, e.g. mean seismicity rate, at time instance i. By p_i we denote the prediction of a model f at time instance i.

1. **Last observation**: at time instance $i \geq 2$, the observed value at time instance $i-1$ is predicted, i.e. $p_i = t_{i-1}$.

2. **Training mean**: predicting the mean over all past observations. At time instance i, we predict the mean over all past observations such that $p_i = \frac{1}{i-1}\sum_{j=1}^{i-1} x_j$.
3. **Auto moving average**: predicting the mean over a fixed window $i > w_i \geq 1$, where the optimal value of w_i is automatically estimated using out of sample walk forward validation (see subsection 5.1). At time instance i we predict $p_i = \frac{1}{w_1}\sum_{j=1}^{w_i} t_{i-j}$. Note that 1. and 2. above are special cases of this, with *w* = 1 and *n* -1 respectively.
4. **Depletion based moving average**: assumes that the activity rate scales with depletion, i.e. $p_i = c_i \Delta P_i$, with ΔP_i the pressure depletion on time interval i and c the scaling relation between activity rate and depletion given by $c_i = T_{i-1,i-a}/(P_{i-1} - P_{i-a})$, with $T_{i,j} = \sum_{z=i}^{j} t_z$ the cumulative number of earthquakes between time intervals i and $j > i$ and a the moving average lookback window. We have set to use the same window selection strategy as for the auto moving average in point 3 above.
5. **Depletion based historical average**: the same as the depletion moving average, but then with $a = i + 1$, i.e. the full history is taken into account.

In principle, the baselines based on activity rate time series are inspired by the observation that observed earthquake counts above the magnitude of completeness in the whole Groningen region (and thus the rates derived from them) are, when aftershocks are not considered, roughly approximated by a Poisson point process with rate parameter $\lambda(t)$. The various forms of averages that we are computing are providing estimates of the $\lambda(t)$ parameter and are also the mean of the Poisson distribution with rate parameter $\lambda(t)$. The physics depletion baselines are inspired by the knowledge that activity rate scales with reservoir pressure depletion.

6.10 Post-Processing of Prediction Results

Once the predictions have been created using any of the algorithms mentioned above, we want to make sure that some generic boundary conditions of the prediction targets are always met. For this purpose, we have implemented generic post-processing routines that can take any algorithm that is available in MLR like the ones mentioned previously and apply certain post-processing transformations to the results.

Currently we have implemented two post-processing routines. The first one enables us to bound the predictions of the algorithms within a specific range. In this way, we can enforce that for prediction targets like count and earthquake rate data only non-negative predictions are created by the "fused" algorithm. Overall, this improves the performance of certain model types that might otherwise produce negative (and thus un-physical) predictions. Additionally, it ensures that if an algorithm should run into numerical issues during model training, especially if the number of training points is still small compared to the number of available features, we can assure that also no unreasonably large predictions are being produced.

The second one makes sure that predictions for count data are integer and not double floating-point values. Similarly, we can enforce for count rate data that if the count rate prediction is multiplied by the number of days over which the count rate was computed that an integer is obtained. We acknowledge that ensuring integer predictions could make numeric predictions worse in most practical settings if measured by a metric like MAE or MSE, never the less it is important to consider that without this integer correction we would be using an unphysical prediction e.g. we would predict that next month there will be ¾ of an earthquake which does not make sense from a purely physical point of view. These additions help to mildly improve predictive performance for some methods while also making sure that some physical boundary conditions, which are fundamental to the problem, are honoured.

7 Methodology: Machine Learning Analysis Tools

Many of the models discussed in chapter 6 are rather complicated in terms of how they internally capture the relationship between the inputs and outputs which essentially turns them into black boxes. However, several techniques exist that allow to extract some human interpretable information. The first analysis steps towards building understanding of black box input-output system are (i) insights in the degree to which features drive model behaviour (as a proxy of the actual physical system), (ii) consequently which features are significant and (iii) how significant features influence predictions. Tools for these analysis steps are discussed in the three sections below. These tools will be applied to individual machine learning models but also play a role in meta-analysis as detailed in chapter 8.

7.1 Variable Importance

Insights into which variables are important drivers for model behaviour can be used as means to start understanding the behaviour of so called black-box models. In the case of our study, variable importance could e.g. provide cues for what drivers might be missing in physics based models. In principle, variable importance is model dependent – unfortunately for many of the model classes discussed previously in section 6 it is not straight forward to estimate variable importance within the model.

For this reason, we limit ourselves to estimating variable importance only for random forest models, for which proven and efficient methods are readily available. While the use of linear models could also provide an estimate of variable importance, the random forest way of measuring variable importance has some key advantages over variable importance assessment via the former:

- non-linear effects and interactions are considered.
- less sensitive to monotonous transformations of the input data.
- still works reasonably well in the presence of several "noise" variables.

Still, they also share some common drawbacks, which we will address individually:

- sensitive to association between the features (issue of association).
- breaks down if the ratio of noise covariates to informative covariates gets too high (issue of multiplicity).

Next, we will briefly describe the underlying methodology that is used to estimate variable importance in a random forest but a more in depth explanation can be found in (Breiman, Random Forests, 2001) and (Friedman, Tibshirani, & Hastie, 2009).

Random Forests are ensembles of decision trees where each tree is trained on a random sub-sample of the available training data using a random subset of features. For each of the data points that were used for training, an internal mean squared error (MSE) score is kept, which measures the predictive performance of the trees that did not use those data points for training. Hence, an out-of-bag (mostly equivalent to an out-of-sample) error estimate is available to judge model performance.

To determine the importance of each individual feature, random permutations are introduced into the data and the performance of the model is assessed again with these randomly permuted data points. An intuitive interpretation is that if the model performance gets substantially worse, a feature must have been important, however if the model performance remains virtually unchanged (or even improves the model) a feature was not important.

An illustrative example of a Random Forest based variable importance plot is shown in Figure 32, where the bars indicate the overall importance of the variable and the whiskers the estimated standard error.

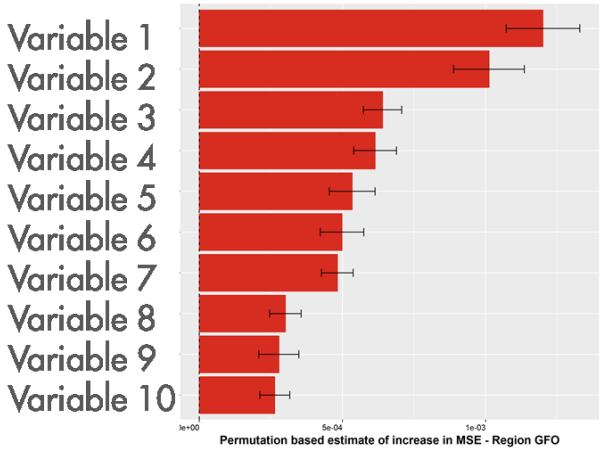


Figure 32: Illustrative example of Random Forest based Variable Importance Analysis

It should be noted that variable importance only works in one direction: if removal of a feature makes the model worse, it was important. The other direction is however is not true, i.e., if removal does not make it worse, it still could have been important, but the encoded information could still be captured through other correlated/associated variables.

7.2 Relevant Variables

We mentioned that the Random Forest based variable importance assessment (just as other methods) suffers from strong association (both linear and non-linear) in the features and from too many noise variables.

For instance, the issue with correlated data is that the contribution of the correlated features gets spread out, mostly uniformly, over the group of features. This may lead to significant effects not being detected. To mitigate this particular issue, we de-correlate the features by determining groups of correlated features (see section 4.4) and by selecting one representative from these groups (see Appendix 2). A record is kept of the group members for future reference and further analysis. Note that effects of non-linear association are potentially more delicate and are not handled via this approach.

To address the issue of multiplicity we determine relevant features using a heuristic approach implemented in the R package Boruta, see (Kursa, 2010) and Appendix 10. On a high level, it works as follows:

1. A random forest model is built from all the features which are either deemed important or for which no decision has been made yet.
2. Random permutations of existing data are introduced and evaluated in terms of the MSE metric, using the trees that were not trained with the particular data points.
3. Features which are significantly worse (two-sided t-test, see Subsection 5.4) than the best random permutation are dropped.
4. Features which are significantly better (two-sided t-test, see Subsection 5.4) than the best random permutation are labelled as important.
5. The unimportant features are dropped and the procedure repeated until all features are either deemed important or a maximal number of iterations (defaults to 100) has been reached.

7.3 Individual Conditional Expectations

Variable importance plots like Figure 32 inform us which features drive model behaviour but not how. To shed some light on this question we are using Individual Conditional Expectation (ICE) plots that look at the average impact of a variable on a model response. ICE plots were introduced in (Goldstein, Kapelner, Bleich, & Pitkin, 2014) as an extension to partial dependence plots which are described in detail in (Friedman, Tibshirani, & Hastie, 2009). ICE plots show the marginal response of the model with respect to changes in one variable. In addition to that they show actual data points and the model response conditioned to all other variables except the one shown on the x-axis of the plot assuming the values of the actual data point. For a more formal definition the reader is referred to (Goldstein, Kapelner, Bleich, & Pitkin, 2014). We will only give a conceptual overview and explain how to interpret the plots.

A stylized example of such a plot is contained in Figure 33. On the x-axis the variable that is investigated is being shown, the range is the range of the actual data such that no extrapolation takes place. The y-axis is the model response. The black dots indicate actual data points. In general only a fixed fraction of actual data points is shown to keep the plots interpretable. The thin black lines going through those data points show the model response conditioned to those data points. That means that for a particular data point the underlying model is evaluated with variable X varying in the observed range and all other variables being fixed to the values of the respective data point. The black line with the yellow border shows the average effect that changes in the variable under investigation have on the model response and is identical to a partial dependence plot. It is computed as the average of the thin black lines. How far the model response deviates from a linear response can be estimated by how much a line deviates from a straight line. If the individual curves are not approximately parallel and show significantly different behaviour this hints at interaction effects, which can subsequently be analysed by creating 2D partial dependence plots.

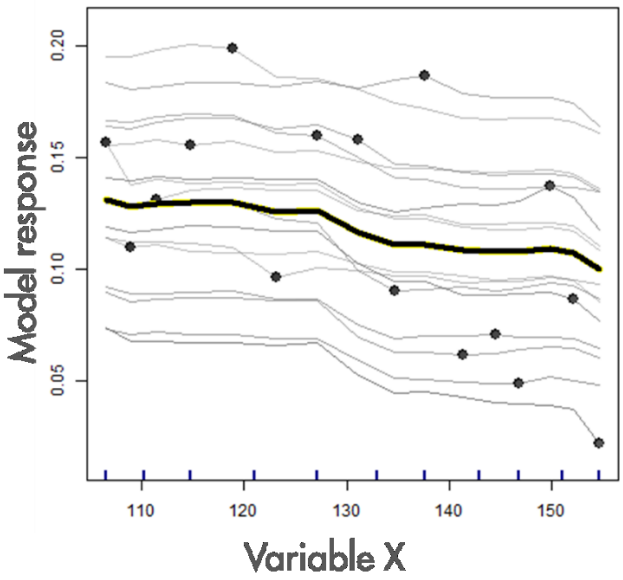


Figure 33: Illustrative example of an Individual Conditional Expectation Plot which shows the average effect of one variable on the model response.

8 Methodology: A Factorial Approach in Combination with Model Meta-Analysis

Previous chapters described a plethora of processing and modelling choices (model meta parameters). For example:

- Feature generation (chapter 3) requires decisions on which features to include, geospatial and temporal aggregation intervals and ranges, aggregation functions (e.g. mean, min, max, …), etc.
- Several meta parameters related to the target (e.g. minimum magnitude, target quantity, etc. – see section 3.2) and data processing (time delays, lags, etc. – see chapter 4) have been defined.
- Potentially suitable machine learning models have been described in chapter 6.

Model performance and model response may be significantly impacted by those choices. As such, understanding the effect of these choices on model performance is important for two reasons. First, it allows us to assess if observed performance effects given certain choices are coincidental or show up persistently, which might provide additional insights in underlying mechanisms. Moreover, it allows us to optimize model meta parameters consistently such that the predictive performance of the model can be maximized, given the available meta parameter ranges. What we mean here by consistently is that we pick meta parameters that fall into a range of meta parameter values that are performing consistently better than other ranges. In this way we ensure that we don't just pick an outlier model that coincidentally performed well in the chosen test metric. It should be noted that just picking the best performing model out of the set of all experiments would lead to overly optimistic estimates of model performance and hence is not a sensible strategy. If the check for consistent performance would not be taken into consideration the performance estimates could be significantly biased and the estimated generalization error could be too optimistic. Still it needs to be noted that the model and meta parameter selection strategy that we apply in this case is fairly novel and it has not been shown that it will lead to un-biased model performance estimates since the optimization procedure does take out-of-sample model performance into account and hence cannot claim to be fully out of sample. Reserving an additional hold-out data set was not considered feasible at the time of writing of this study due to the overall small data set size. Hence total clarity on this point can be reached at a later moment in time when a sufficient amount of new unseen data will be available to validate if the performance estimates of the model(s) are in line with the estimates obtained through our selection procedure. Alternatively, one could train/test the models up to 2012 and use the remaining data as a hold-out set but this will decrease statistical power for model selection and therefore in not the approach taken here. An often followed approach for studying meta parameters and their impact on performance is to analyse their impact individually, however this does not take interaction effects between meta parameters into account. Instead, this study takes a factorial approach and runs experiments on all combinations of plausible parameter combinations of the meta parameter space in an iterative fashion; the details are described in section 8.1. The meta data resulting from factorial runs is extensive. Section 8.2 elaborates on the meta-analysis setup to gain insights and iteratively downselect the model and meta parameter space in three ways: machine learning model selection (section 8.3), meta parameter range reduction (further detailed in section 8.4) and feature down-selection (section 8.5). In several of these sections we employ the machine learning analysis tools explained in chapter 7. In summary the model meta learning approach is used to select a robust model and meta parameter combination and to make (to a lesser) extent an inference about certain parameter choices and their impact on model performance.

While this chapter covers the underlying methodology its application and the respective results will be covered in chapter 9.

8.1 A Factorial Approach

The main model meta parameters are related to the machine learning models that are used, the exact target definition (what we aim to predict), data integration choices and feature selection thresholds. When considering the values probed for each of the meta parameters, we face the familiar trade-off between range and resolution in order to keep the number of model training/evaluation runs manageable. We approach this via iterative downselection: starting with the broad range with coarse resolution and iteratively "zoom in" on the best and robust performing meta parameter subset. An exception to this are meta parameter choices which cannot easily be compared, e.g. different target definitions. Here we make a decision based on which choices best fit the scope of this study. The parameter range of our factorial experimental design is shown in Table 12. The obvious meta parameter set missing is the set of machine learning model hyperparameters – as hyperparameter tuning is computationally intensive these are not part of the factorial setup but analysed later on in a downselected meta parameter space.

An experiment is defined as a unique combination of meta parameters. As can be seen in the table below, we start with a factorial experimental design of around 175,000 experiments. Earlier versions of the experimental setup were exploring a larger parameter space that lead to around 4 million experiments. The number of experiments has been reduced to focus the study on the (expected) most relevant meta-parameters. Meta data is recorded for each experiment, including:

- Meta-parameter choices;
- Model performance for each of the error metrics (section 5.2) with the associated standard errors (section 5.3);
- Random Forest based variable importance (increase in MSE together with SE) for each feature (section 7.1);
- List of significant and potentially significant features (section 7.2) per region as determined by the Boruta test.

The following section elaborates on the analysis performed on this meta data.

Meta parameter	Value range	# Val.	Ref.
ML Model (excl. baselines)	RF, KNN, SVM, NN, GLM, GLM Net, GLM Top, ARIMA, GBM	7	Chapter 6
Target quantity	EQ rate (also implemented EQ Count, EQ Energy Released but not investigated in this study)	1	Section 3.2
Targets	(a) $M_{min} = 1.5, T_{start} = 1995, T_{agg} = 3$ months (b) $M_{min} = 1.2, T_{start} = 1995, T_{agg} = 3$ months (c) $M_{min} = 1.2, T_{start} = 2004, T_{agg} = 3$ months (d) $M_{min} = 1.0, T_{start} = 2004, T_{agg} = 1$ months	4	Section 3.2
Geospatial agg.	GFO	1	Section 3.2
Aftershock processing	None (also implemented: windowGK but not investigated in this study)	1	Section 3.2
Time delay Q	0, 2, 4, 6, 8, 10, 12	7	Section 4.1
Time delay P	0, 2, 4, 6, 8, 10, 12	7	Section 4.1
Time delay S	0, 2, 4, 6, 8, 10, 12	7	Section 4.1
Time delay C	0, 2, 4, 6, 8, 10, 12	7	Section 4.1
Smoothing	None	1	Section 4.2
Max. nr. Lags	0, 2	2	Section 4.3
Feature correlation threshold	0.9	1	Section 4.4
Feature transformations	None (also implemented: PCA, JY but not investigated in this study)	1	Section 4.5
Feature significance treshold	0.4	1	Section 4.6 and 8.4
Nr. of experiments		**172,872**	

Table 12: The factorial experimental design of the meta parameters probed in the initial runs. Iterative downselection will decrease the ranges of the meta parameters used. One combination of meta parameter choices can be regarded as an experiment.

8.2 Meta Analysis Setup

Given the number of experiments, systematic analysis of experiment outcomes is required, i.e. model meta analysis. It will not provide any direct insight on seismicity itself, but rather on meta parameter choices that in turn should improve seismicity predictions. We use the same tools for meta analysis as in the main analysis itself: machine learning models as explored in chapter 6 and the machine learning analysis tools described in chapter 7. For meta analysis we restrict ourselves to establish a relationship between the meta parameter choices made and model performance using random forests. We choose to use random forest models because of their ease of use (no data pre-processing required), proven performance in a lot of other use cases, and the amount of available diagnostic tools and plots.

First, we assess the performance of the overall meta model using the reported out-of-bag estimate of explained variance R^2, using the random forest package in R. Only if the model manages to explain a fair fraction of the variance in the data, we do proceed with further analysis. This analysis includes again the usual variable importance analysis, testing for significant variables and partial dependence plots that illustrate the marginal effects of the individual features on the prediction target.

Depending on how we partition the experiments, we can address different questions, which are relevant in the context of this study. In Table 13 we give a non-exhaustive overview of questions that could be addressed with our setup. Note that only the questions related to the first two groupings have been investigated in detail in this study.

Data subsets/groupings:	**Questions addressed:**
Best set of equivalent models for metric $\boldsymbol{m}$ as determined by standard error margins/ hypothesis testing for a specific prediction target	• How consistent are the effects of the features and meta-parameters across the different models? • Can we identify optimal parameters ranges for the significant meta-parameters?
Full data set	• What ranges of model meta parameters lead to better model performance? • How sensitive is model performance to the choice of model meta parameters?
Grouped by Region	• In which regions are which features significant (based on the Boruta test)? • Are some regions easier to predict than others?
Grouped by minimum magnitude	• In which magnitude ranges are which features significant (based on the Boruta test)? • Are some magnitude ranges easier to predict than others?

Table 13: Different partitions of the experiments can be used to address different questions. Only the first two have been investigated in this study.

In order to determine the set of best equivalent models for a given metric m from the pool of experiments, we currently use the following approach. It is visual in nature and based on the heuristic of determining the (frequentist) confidence bands as indicated by the standard error around the error measure m for each experiment. The models for the same prediction target whose

confidence bands intersect with the error bands around the best model w.r.t. to m are considered to be equivalent. At its core this is very similar to a more formal unpaired hypothesis test which have been discussed earlier in this report.

Based on meta analysis, we proceed to downselect machine learning models (section 8.3), constrain the relevant feature value ranges (section 8.4) and reduce the number of features (section 8.5).

8.3 Machine learning model down-selection and hyperparameter tuning

As mentioned in section 6.1 it is hard to reliably predict a priori which machine learning algorithm will perform best for a dataset like the one that we are dealing with here without performing actual prediction experiments on the data. The reason being that no representative meta-studies are available on this topic in literature that would have established which algorithm (or group of algorithms) seem to be performing well on average. Hence performing a significant number of experiments will allow us to get insights in the average performance of each algorithm that can be expected when applied to new data. Given the usual relative performance spread and the possible effects of tuning we don't apply very strict criteria to exclude below-average performing models but base exclusion on visual inspection of performance distribution.

Models have hyperparameters (see section 6) which can be tuned to potentially improve predictive performance. An individual benchmark experiment takes around 10 minutes to run if the machine learning algorithms are run with default hyperparameters. The execution time goes up to about 1 hour if model hyperparameter tuning is enabled. Hence there are limits to the number of meta parameter combinations that we can explore in a factorial setup when also model hyper parameter tuning is enabled. For this reason, we use an iterative process in which we reduce the number of features and meta parameters in a stepwise fashion to a more manageable set in which we observe average model performance to be robust and good. Only then, we do switch on full model hyperparameter tuning. For model hyper parameter tuning the usual concept of splitting the data into a train, test and validation data set is used. Practically this means that the training set defined in our walk-forward validation approach which is described in section 5.1 is again sub-partitioned into a training and test set to which the same walk-forward approach, with the same step length l, is applied to tune model hyperparameters. In other the words the re-sampling setup that is used for model hyper-parameter tuning is consistent with the setup that is used for estimating the out-of-sample performance of the model. An illustration of the process can be found in Figure 34.

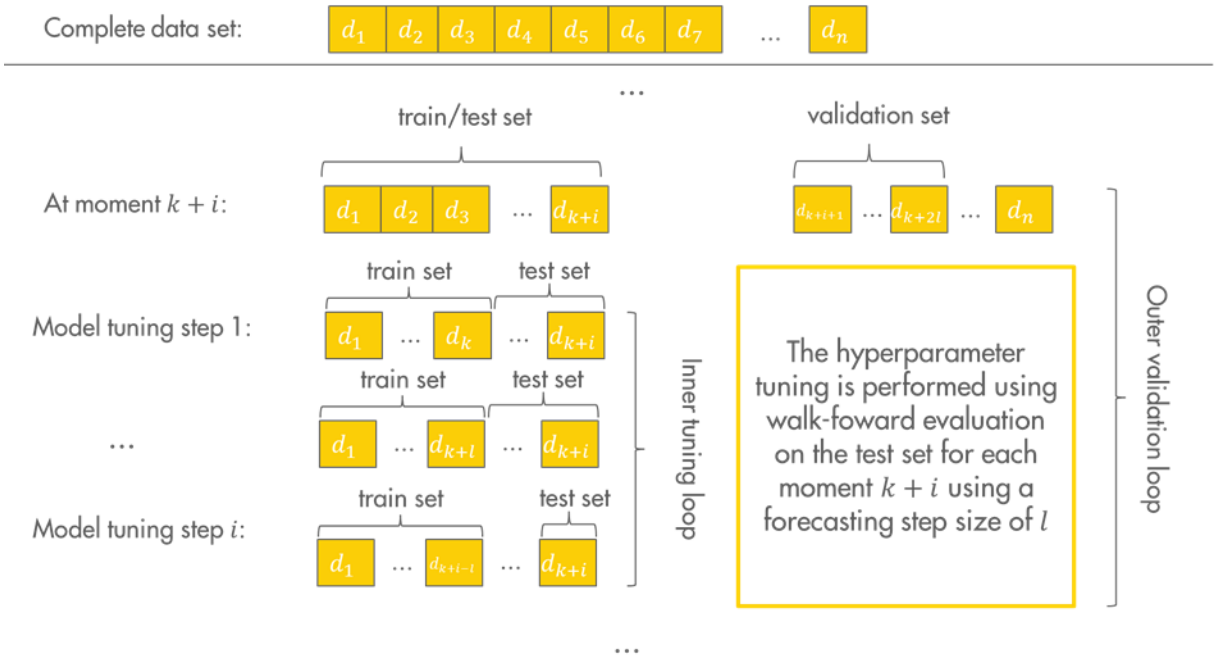


Figure 34: Illustration of inner resampling for model tuning using walk forward approach at training and prediction time step $k + i$. The procedure is repeated for each forecasting step as part of the outer validation loop.

The initial exploratory tuning for the individual machine learning models had been determined in a separate exercise in which a possibly large parameter space was investigated using the walk-forward validation approach described in section 5.1 applied in an inner loop to the training data. Hence there can be no leakage of information. In the exploratory stage, the i-race algorithm as implemented in the i-race package was used (M. López-Ibáñez, 2016), see also Appendix 10. It allows to automatically tune the parameters of an algorithm subject to an objective function (e.g. the prediction error in a given error metric) and a computational budget.

8.4 Constraining relevant meta parameter value range

To constrain meta parameter ranges to an interval on which they give more predictive performance we use Individual Conditional Expectation (ICE) plots, see section 7.3. The application of ICE plots to constraining and refining meta-parameter ranges is fairly straightforward. As model response we take an error metric and as variables we take the meta parameter of interest, then a stable (with respect to small changes in the parameter) local minimum in the ICE plots will correspond with on average improved model performance for the parameter under investigation.

8.5 Feature down-selection

Selecting a possibly minimal set of features that achieve peak model performance is desirable for several reasons. It makes models more easily interpretable if "noise" variables are removed. Additionally, some models, e.g. plain linear models without regularization, tend to perform poorly if too many (noise) terms are used to fit the model. To a lesser extent this also applies to more advanced machine learning based models. For that reason, it is crucial to assess the importance of individual features on model performance. Furthermore, if the model is reduced to a limited number of features this might help to provide more easily accessible physical insights. The most relevant features are selected in two steps: (i) correlation based down-selection and (ii) significance (with respect to a model) based down-selection.

First, highly correlated features are grouped and only one representative of the group is taken forward. The representative is chosen based on discussions with domain experts. A record of the dropped features is created since this might help with the interpretation of model results. Details of correlation based feature removal can be found in section 4.4.

Model based feature significance testing as a second step is slightly more involved. We start by removing all variables which are never tested as relevant by the Boruta algorithm, which is explained in more detail in Appendix A10.2. However, since we are deriving several new features like first and second order difference quotients from several time delayed versions of raw data sources, some of them may by pure chance be correlated to the prediction target. In order to minimize this effect, we also exclude variables that are being tested as significant (by the Boruta significance test) from the workflow whose detection ratio is outside of a safe margin of the false-positive detection rate that we have established for a random permutation of the original data. More details of how this cut-off is determined can be found in Appendix 5.

9 Results Meta-Analysis: A Robust Model & Meta Parameter Combination

In chapter 8 we describe the methodology for the meta analysis, namely the downselection from the factorial setup to chosen robust Model & Meta Parameter combinations (MMPs) for each target. In this chapter the results are presented. In a nutshell, starting with the factorial setup as shown in Table 12 we downselect machine learning models as explained in section 9.1, reduce the meta parameter ranges as described in section 9.2 and downselect features as explained in section 9.3. On the resulting smaller meta parameter space we tune the machine learning model hyperparameters in section 9.4 and present the robust MMPs used for seismicity predictions in section 9.5. A schematic illustration of the flow of this chapter is shown in Figure 35. Given the factorial approach in our experimental setup we also build in some guards against spurious detections. These are described in Appendix 5.

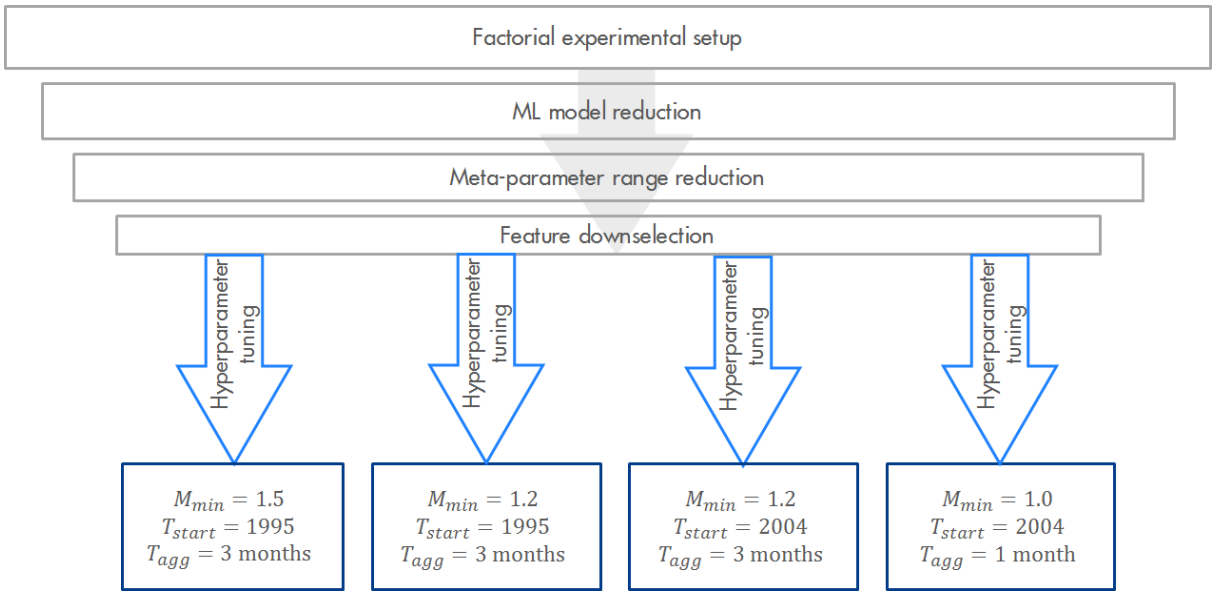


Figure 35: Sketch of the downselection process from the factorial experiments to the robust MMPs for the various target choices. The three arrows (yellow, red, and blue) indicate the hyperparameter tuning for each of the minimum magnitudes used in this study (1.0, 1.2, and 1.5, respectively).

9.1 ML model reduction

As described in chapter 6, RF, KNN, SVM, NN, GLM and variants (GLM Net, GLM Top), ARIMA and GBMs have been selected as initial models. Given consistent underperformance compared to other models, GBM and pure GLM models are not progressed. A visual overview of model performance spread of the progressed models as well as our baselines is shown in Figure 36. Neural Nets also perform relatively poorly but we hope that hyperparameter tuning in section 9.4 significantly improves their performance.

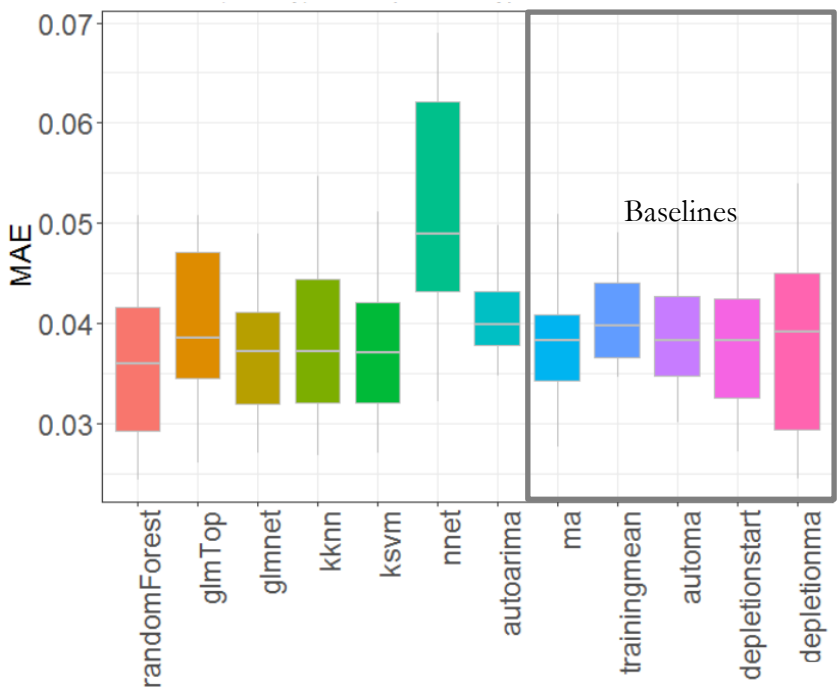


Figure 36: Illustrative example of a Relative Performance Plot based on MAE, the boxplots show the spread in the prediction of the models and the mean performance.

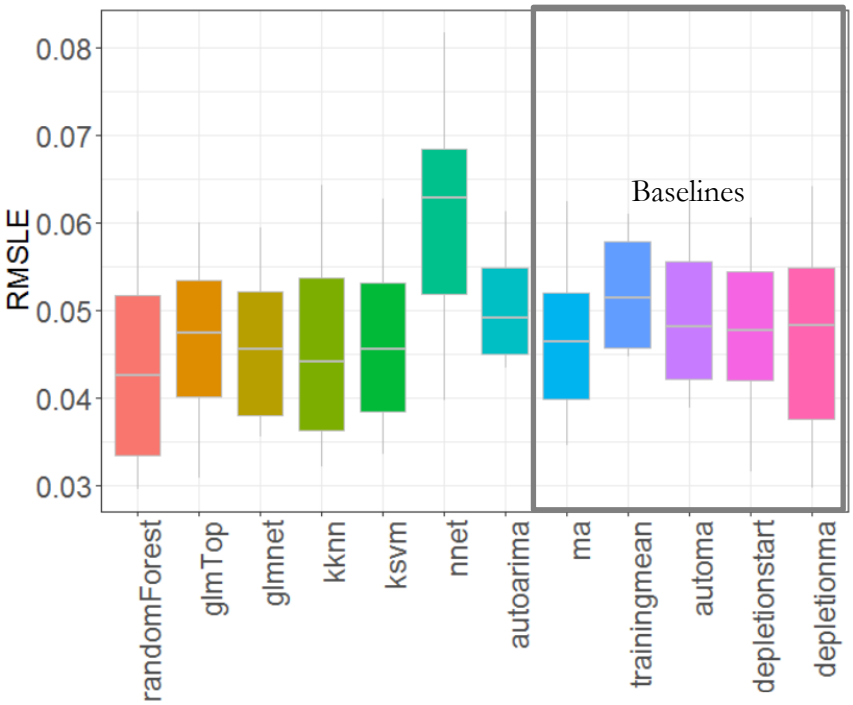


Figure 37: Illustrative Example of a Relative Performance Plot based on RMSLE, the boxplots show the spread in the prediction of the models and the mean performance.

9.2 Meta-parameter range reduction

Following the downselection of machine learning models, we further reduce the number of experiments by decreasing meta-parameters ranges. The largest contribution to the number of experiments comes from the time delays, contributing in total with a factor of $2401 = 7^4$. Variable importance assessment followed by ICE plots are used to downselect this to a smaller number.

The variable importance assessment of the time delay parameters is shown in Figure 38 for $M_{min} = 1.5$ (3-month aggregation and period 1995-2016), $M_{min} = 1.2$ (3-month aggregation and periods 1995-2016 and 2004-2016) and $M_{min} = 1.0$ (1-month aggregation and period 2004-2016).

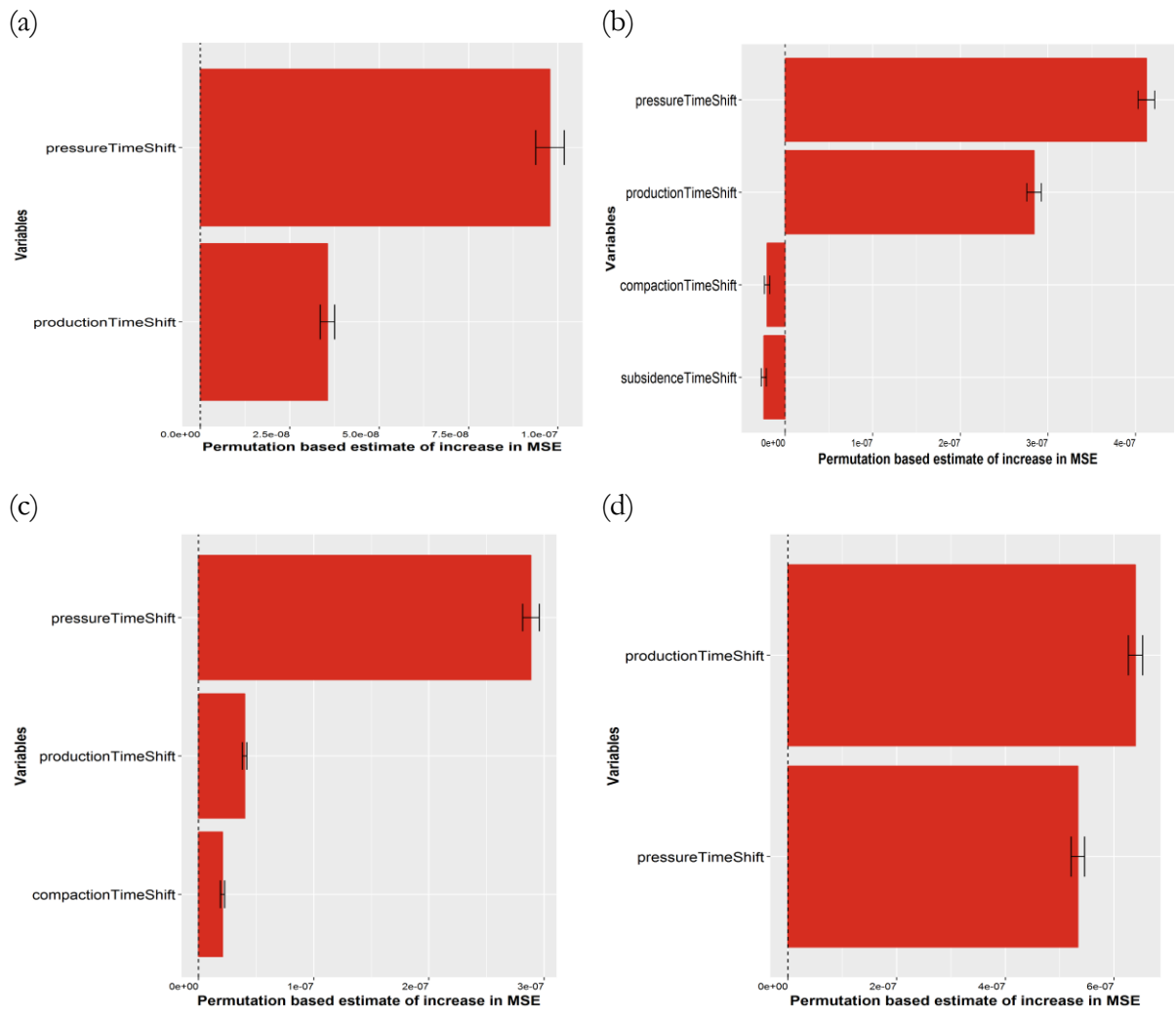


Figure 38: Variable importance assessment for the relationship between time delays and predictive performance for the selected experiments in Table 15. Importance is expressed as the meta-analysis random forest MAE – a larger MAE increase means the meta-parameter is more important. (a) $M_{min} = 1.5$ and 3-month aggregation period from 1995-2016, (b) $M_{min} = 1.0$ and 1-month aggregation period from 2004-2016, (c) $M_{min} = 1.2$ and 3-month aggregation period from 1995 – 2016, (d) $M_{min} = 1.2$ and 3-month aggregation period from 2004 – 2016.

Note that for all magnitudes of completion and aggregation periods only the pressure time delay and production time delay seem to have a significant impact on our ability to predict seismicity. The associated ICE plots for Figure 38b are shown in Figure 39.

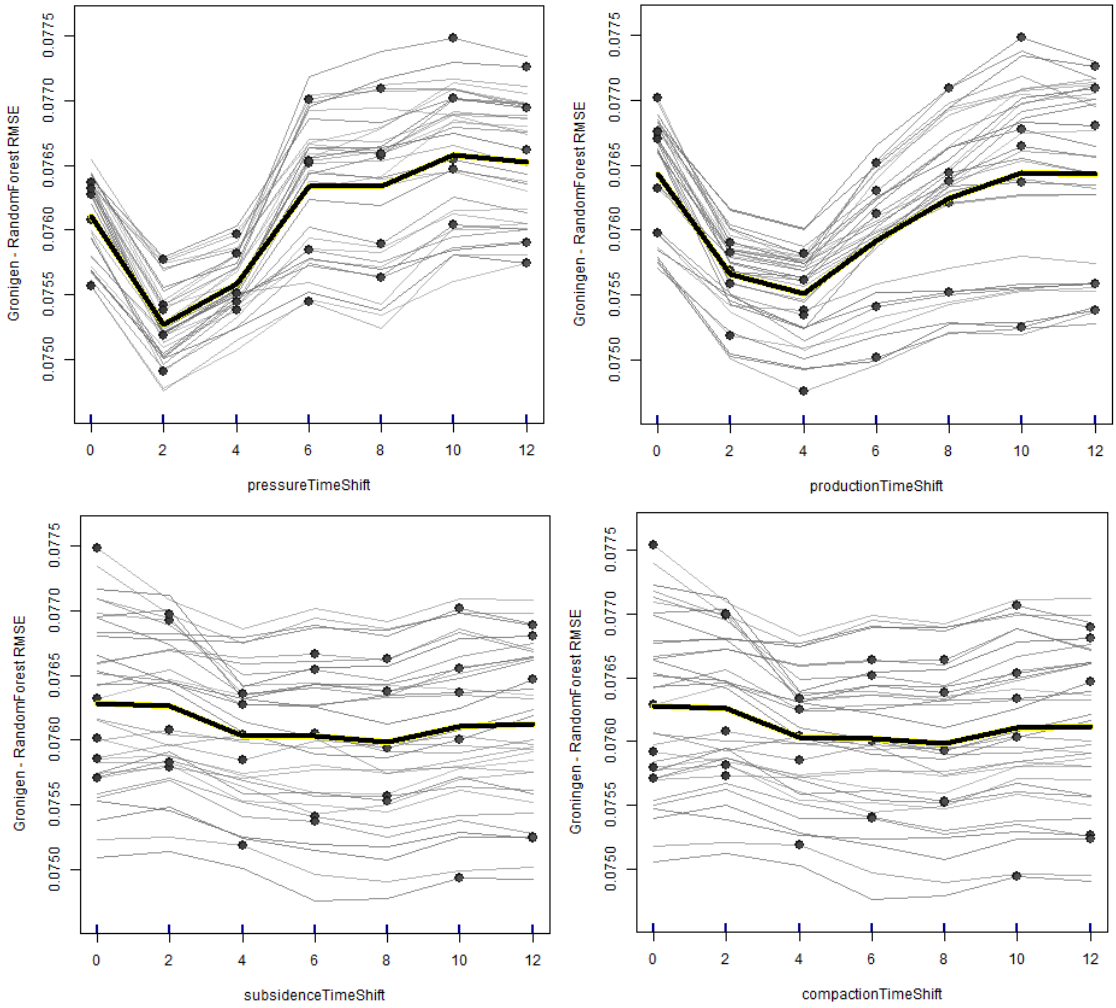


Figure 39: ICE plots for different time delays for $M_{min} = 1.0$ and aggregation period of 1 month, indicating the average impact on predictive performance w.r.t. experiments described in Table 15.

Based on (iterative refinements of) ICE plots as above we can refine time delay parameter ranges. For pressure P the minimum seems to be around 2 months, so we take as corresponding time delay range 1, 2 and 3 months. Similarly, for production Q we keep a time delay range of 3,4 and 5 months. For compaction C and subsidence S a clear minimum is less evident, so we keep a larger range of values. We note that all time shifts have a rather minimal impact on RMSE, less than $\pm 2\%$. As such, it is not evident at this point that time shifts significantly improve model performance. To test for this, we also include time shift 0 for all variables.

Additionally, based on exploratory analysis it seems that lags help with predictive performance so we discard the 0 lags option. Combining the above we reduce the number of options available in our factorial experimental design with nearly 95% to around 10,700 experiments as shown in Table 14.

Meta parameter	Value range	# Val.
ML Model (excl. baselines)	RF, KNN, SVM, NN, GLMnet, GLMtop, Arima	7
Target quantity	EQ rate	1
Targets	(a) $M_{min} = 1.5, T_{start} = 1995, T_{agg} = 3$ months (b) $M_{min} = 1.2, T_{start} = 1995, T_{agg} = 3$ months (c) $M_{min} = 1.2, T_{start} = 2004, T_{agg} = 3$ months (d) $M_{min} = 1.0, T_{start} = 2004, T_{agg} = 1$ months	4
Geospatial agg.	GFO	1
Aftershock proc.	None	1
Time delay Q	0, 3, 4, 5	4
Time delay P	0, 1, 2, 3	4
Time delay S	0, 2, 3, 4, 5, 6	6
Time delay C	0, 2, 3, 4	4
Max. nr. lags	2	1
Smoothing	0 months (none)	1
Transformations	None	1
Feature correlation threshold	0.9	1
Feature significance treshold	0.4	1
Nr. of experiments		**10,752**

Table 14: Experiments which are taken forward to the model hyperparameter tuning experiment

9.3 Feature down-selection

The objective of down-selecting features is to make ML models more easily interpretable in addition to increasing their overall performance. We refer the reader to section 8.4 for the full explanation on the down-selection methodology. As noted there, feature down-selection happens in two steps: (i) correlation based feature selection; (ii) significance based feature selection. The correlation based selection is straightforward and the full list of correlation groups for each final variable can be found in Appendix 1. We highlight that the resulting subset of representative features all bear a certain amount of unique information.

Feature significance testing is based on the Random Forest based variable importance assessment, which is underlying the significance test that is implemented in the Boruta algorithm. Following the approach described in Section 8.4, we found that the false-discovery rate of at least one feature derived from a randomly permuted source data set to be tested as significant is around $25\% \pm 15\%$. Therefore, we use a cut-off threshold of 40% under which a variable is excluded from further experiments if either that variable or one of its lagged variants is tested significant in fewer instances. Features and the percentage of experiments where they are significant are shown in Figure 40 for the target $M_{min} = 1.5$, $T_{start} = 1995$ and $T_{agg} = 3\,\text{months}$. The feature significance plots for the other targets are shown in Appendix 3.

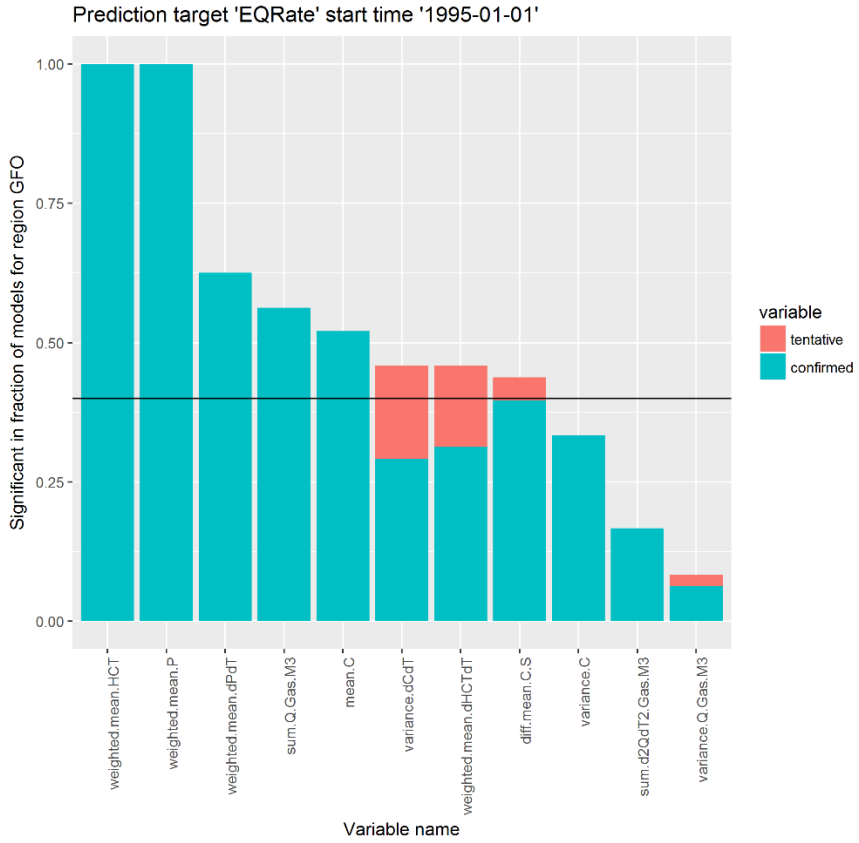


Figure 40: Fractions of GFO models for which covariates were tested as significant. The black line indicates survival threshold, all features below this threshold are discarded.

A summary of excluded features is shown in Table 15. Note that tests for variable significance in a model are always conditional to the prediction target and would in theory need to be repeated each time a different prediction target is used. However, we have empirically determined that there is little variability between the prediction targets investigated here.

Region	Excluded covariates
GFO	variance.d2QdT2.Gas.M3, weighted.mean.d.FF.HCTdT, variance.S, weighted.mean.d2.FF.HCTdT2, weighted.mean.d2HCTdT2, variance.dQdT.Gas.M3 ,mean.dCdT, weighted.mean.d2PdT2, sum.dQdT.Gas.M3, variance.dSdT

Table 15: List of covariates that were excluded for the target EQRate

9.4 Model Hyper Parameter Tuning

Following the large reduction in experiments via model downselection, meta parameter range reduction and feature downselection we are left with a relatively small set of model and meta parameters combinations, allowing the relatively computationally expensive hyper parameter tuning. Below we describe hyperparameter tuning for Random Forests (Figure 41), KNN (Figure 42), SVM (Figure 43), Neural Nets (Figure 44) and GLM Net (Figure 45). For the sake of brevity and given the similar results for all the minimum magnitudes under consideration, we focus here on the results obtained for $M_{min} = 1.0$. While the ranges and combinations are by no means exhaustive, they represent a sensible compromise between expected improvement in model performance and computational cost. The reader interested in more details regarding the hyperparameters is referred to Appendix 4.

Random Forest		
Parameter	**Default**	**Tuning Range**
nTree	500	[100,1000]
mTry	$\lfloor m/3 \rfloor$	$[1, \lfloor m/2 \rfloor]$
nodeSize	5	[1,10]

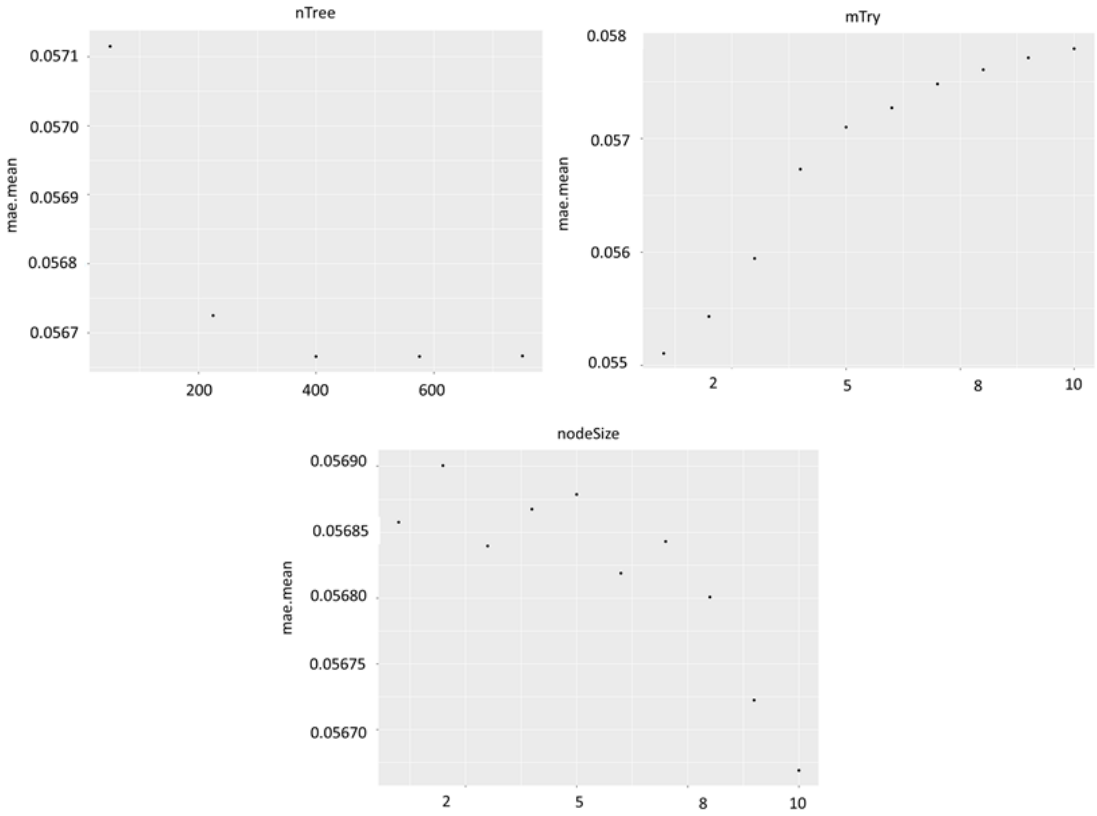


Figure 41: Average effect of tuning Random Forest hyperparameters on the MAE error measure. From left to right: hyperparameters nTree, mTry and nodeSize.

K-Nearest Neighbours		
Parameter	**Default**	**Tuning Range**
K	7	1-12
Distance	2	[0.5,2.5]
Kernel	Optimal	rectangular, triangular, epanechnikov, optimal

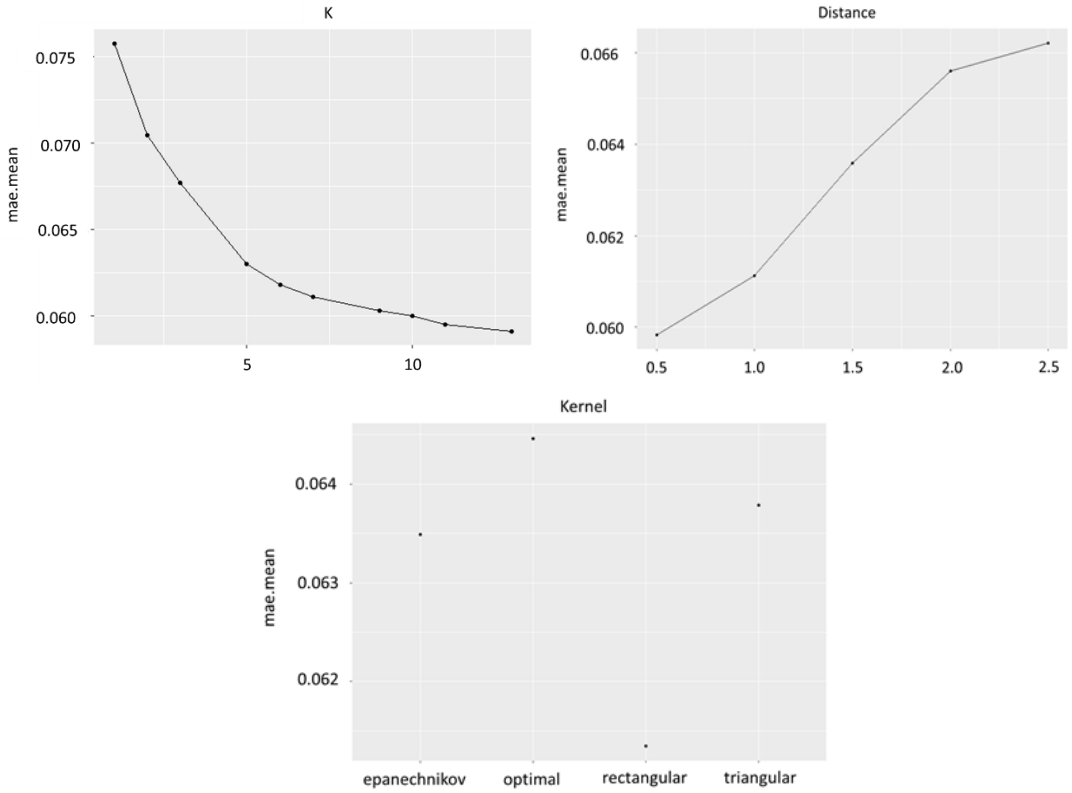


Figure 42: Average effect of tuning KNN hyperparameters on the MAE error measure. From left to right: hyperparameters K, Distance and Kernel.

Kernel SVM		
Parameter	**Default**	**Tuning Range**
Kernel function	Gauss kernel	Gauss kernel
SVM-Type	ϵ-SVR	ϵ -svr, ν-SVM, ϵ -bsvr
C	1	$[2^{-5}, 2^{3}]$
ϵ	0.1	[0,1]
ν	0.2	[0,1]
σ	Estimated from data	Estimated from data

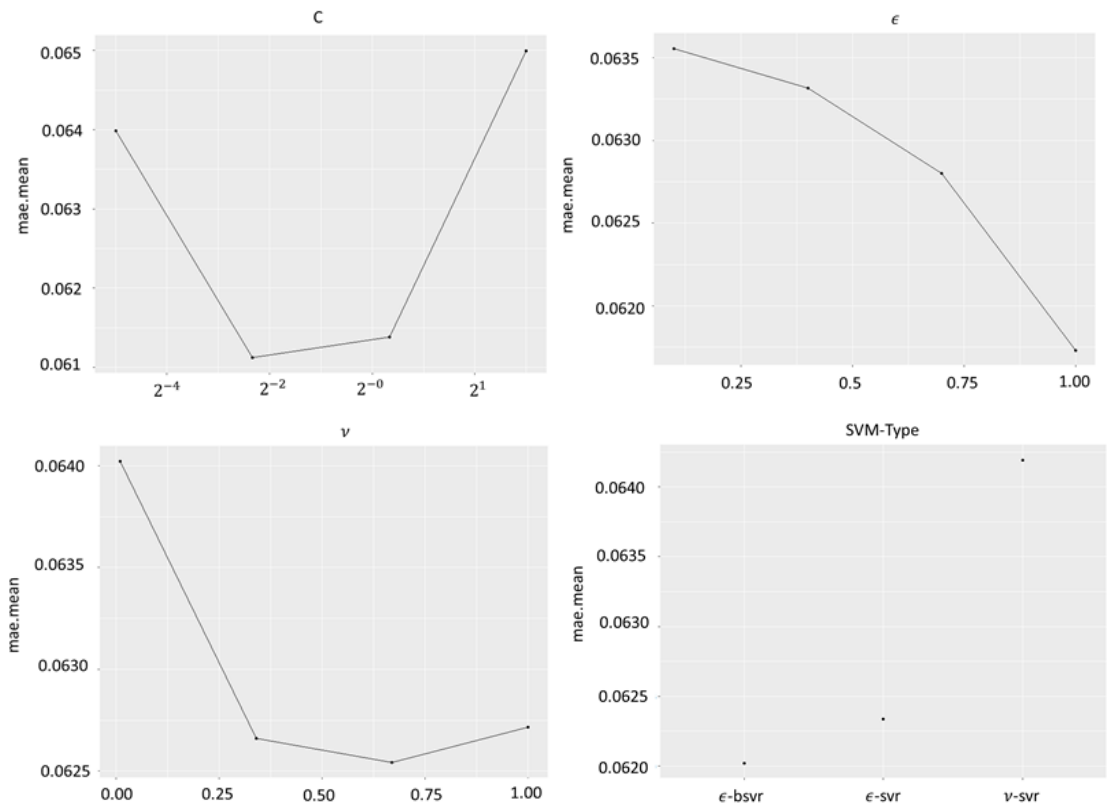


Figure 43: Average effect of tuning SVM hyperparameters on the MAE error measure. From top left to bottom right: hyperparameter SVM-type, C, ϵ, and ν.

Shallow Neural Network		
Parameter	**Default**	**Tuning Range**
Size	2	2:10
Max iterations	100	[100, 200, 300]
Abs. tol.	10^{-4}	[0.0001, 0.01]

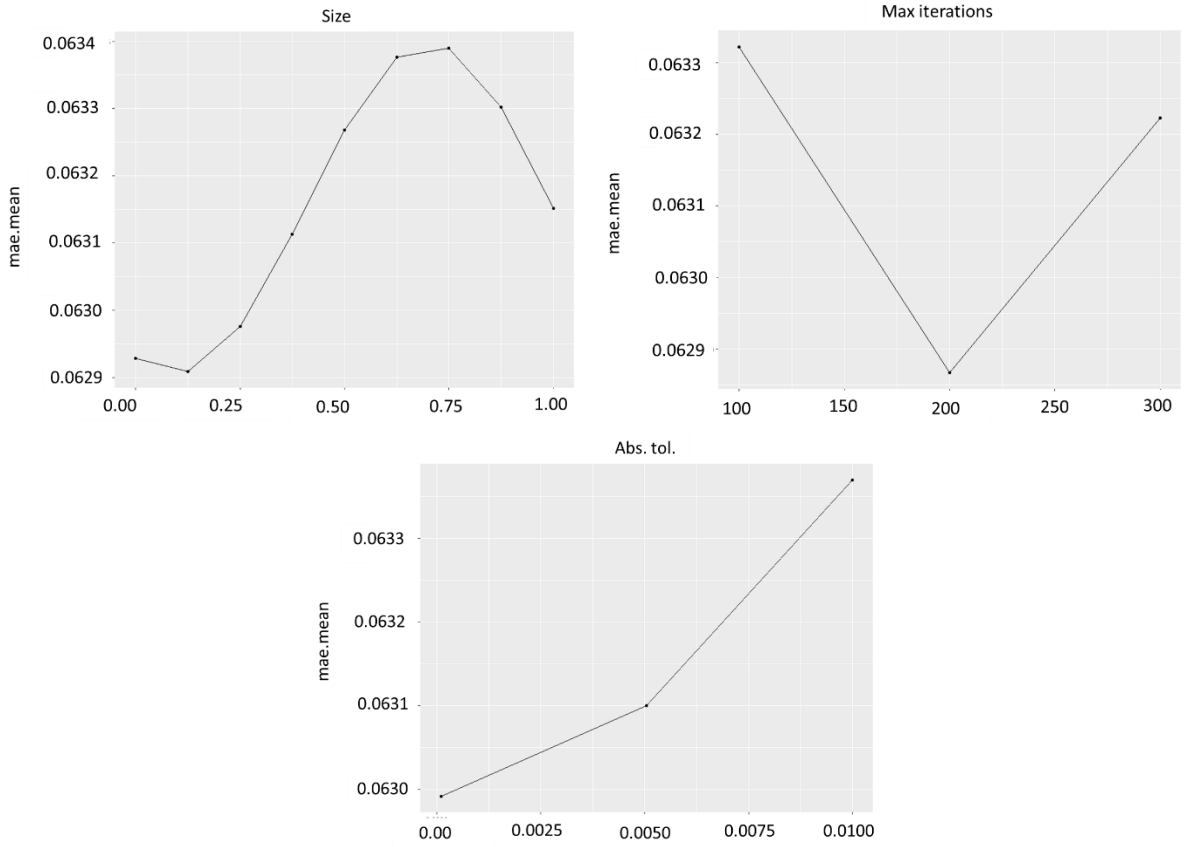


Figure 44: Average effect of tuning NN hyperparameters on the MAE error measure. From left to right: hyperparameters size, maximum number of iterations and absolute tolerance.

GLM Net		
Parameter	**Default**	**Tuning Range**
Family	Gaussian	Gaussian, Poisson
Alpha	1	[0,1]
nLambda	100	[100,250]

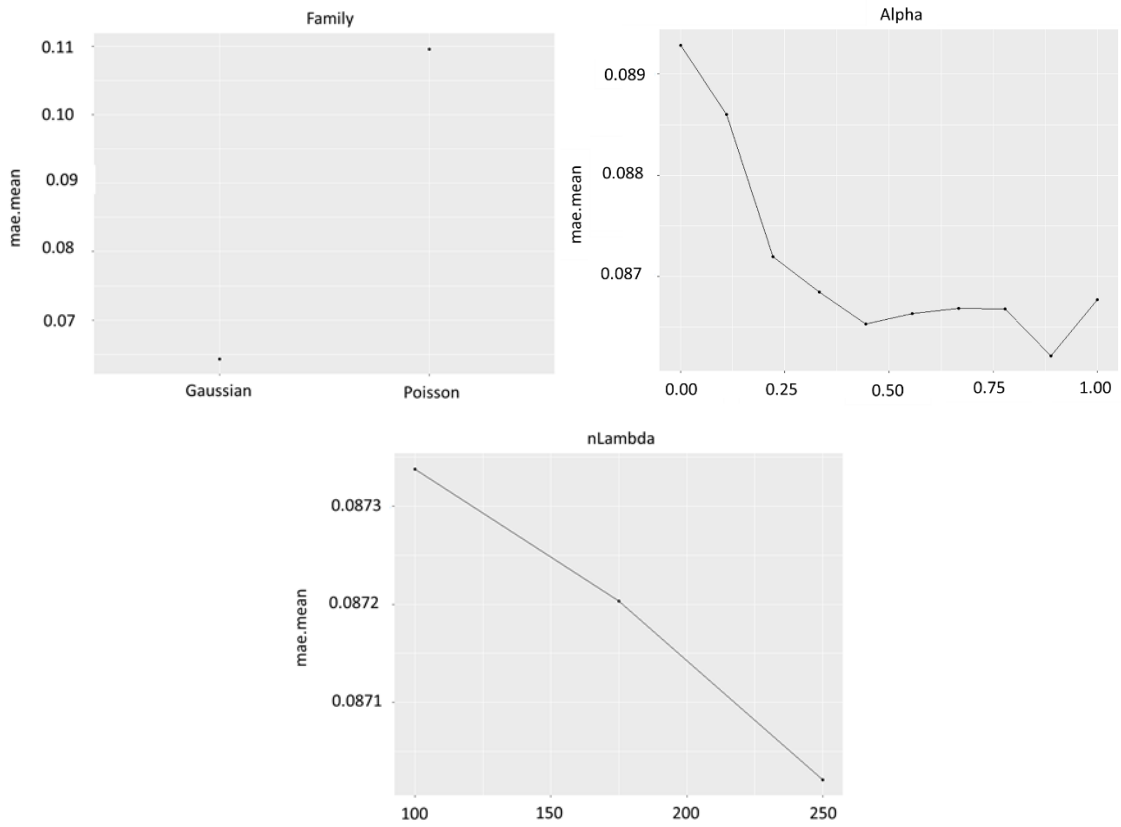


Figure 45: Average effect of tuning GLM Net hyperparameters on the MAE error measure. From left to right: Family, Alpha and nLambda.

The average improvement in model earthquake rate predictability for the GFO region, in terms of MAE and RMSLE, is shown in Table 16. The performance of Random Forests, KNNs and SVRs doesn't change in a meaningful way, GLM Nets improve reasonably (~14%) within the RMSLE error metric but only relatively little in the MAE error metric (~2.5%), while Neural Nets achieve improvement in both error metrics (7.5%).

Algorithm	Mean Relative Reduction in MAE	Mean Relative Reduction in RMSLE
KNN	0.02%	0.18%
Random Forest	0.41%	0.71%
Kernel SVR	-0.01%	0.02%
NNet	7.81%	7.28%
GLM Net	2.45%	13.83%

Table 16: Overview of observed mean relative improvement in model performance. Negative numbers indicate an increase in contrast to a reduction.

9.5 Final model and meta parameter selection

With hyperparameters tuned we make a final run of experiments with the meta parameter ranges as given in Table 14. From this final run for each target a robust MMP is chosen using the following criteria:

1. Robustness: small changes in meta-parameter values should result in only small changes in model MAE;
2. Explanatory power: the MMP should have an $R^2 \geq 0.1$;
3. Minimal model error: for those MMPs satisfying the criteria above we search for those with a minimum MAE and RMSLE.

When differences between models with time delays and without time delays are very small, we prefer MMPs without time delays (0 time delays) in favour of physical interpretability. Of course this is only required when the MMP itself has a straightforward physical interpretation, which is not the case for $M_{min} = 1.0$ as this minimum magnitude combines changes in seismicity rate with changes in detection sensitivity. The final MMP choices for each target are documented in Table 17, throughout this study we will refer to these MMPs by their ID as shown in the second row. Given that error estimates between physically straightforward interpretable MMPs with and without time delays are less than 1% and are not statistically significant, we use 0 time delays in most cases.

Target choices	$M_{min} = 1.5$ $T_{start} = '95$ $T_{agg} = 3\,\mathrm{m}$	$M_{min} = 1.2$ $T_{start} = '95$ $T_{agg} = 3\,\mathrm{m}$	$M_{min} = 1.2$ $T_{start} = '04$ $T_{agg} = 3\,\mathrm{m}$	$M_{min} = 1.0$ $T_{start} = '04$ $T_{agg} = 1\,\mathrm{m}$
MMP ID	**RF-FC-01-1.5**	**RF-FC35-1.2**	**RF-FC36-1.2**	**RF-FC107**
ML Models	RF, SVM, KNN, GLM Top	RF, KNN, SVM, GLM Top	RF, KNN, SVM, GLM Net,	RF, GLM Net, SVM, GLM Top
Geospatial agg.	GFO	GFO	GFO	GFO
Temporal ranges	1995-2016	1995-2016	2004-2016	2004-2016
Temporal agg.	3 months	3 months	3 months	1 month
Min EQ mag.	1.5	1.2	1.2	1.0
Time delay Q	0	0	0	5
Time delay P	0	0	0	2
Time delay S	0	0	0	3
Time delay C	0	0	0	0
Max. nr. Lags	2	2	2	2

Table 17: Final choice of MMPs for the targets. These MMPs will be used for seismicity predictions in chapter 10.

10 Evaluation of Machine Learning based Seismicity Event Rate Forecasts

The meta-analysis in chapter 9 provides us with robust and relatively well-performing Model & Meta-Parameter Combinations (MMPs) for each of the targets. This chapter evaluates the seismicity event rate forecasts these MMPs generate. Within a Machine Learning context, forecast performance is a fundamental measure on which models get evaluated (Breiman, Statistical Modeling: The Two Cultures, 2001), we will quantify this in section 10.1. For usability within the PSHRA framework, seismicity event rate forecasts for at least 1 to 5 years ahead are required. We investigate the qualitative behaviour of the models over years in section 10.2. Subsequently, we comment on the range of validity of the methodology as reported on in this study in section 10.3. The focus of this chapter will be on the event rate forecasts for the targets $M_{min} = 1.5$ and $M_{min} = 1.2$ with $T_{start} = 1995$. For an overview of the quantitative evaluation of the other targets we refer to Appendix 7.

10.1 Quantitative Evaluation: Forecast Performance

Following the approach developed in chapter 5, a quantitative evaluation of the forecast performance is based on the ability of models to beat simple baselines, followed by model performance in our selected error metrics (MAE, RMSLE, R^2). With the final MMPs selected, to determine whether these MMPs manage to beat the baseline in a statistically significant way we apply the test procedure as outlined in section 5.4. This test procedure allows us to determine the right test statistic to be applied, by testing for normality, randomness and autocorrelation in the residuals. In the lines below we will go over the results for the target of $M_{min} = 1.5$ in some detail, for the other targets the procedure is analogous and only the results are shown.

As mentioned the first check is to identify if there are any outliers in the absolute value of the residuals – if outliers are present, this would make the case for a non-parametric test. We test for outliers by visual inspection of a QQ and box plot for both models to be compared, see Figure 46 below for the QQ plot for the Random Forest with $M_{min} = 1.5$. The QQ plot doesn't show a straight line, from which we conclude a non-normal distribution and hence a non-parametric test is the more appropriate choice.

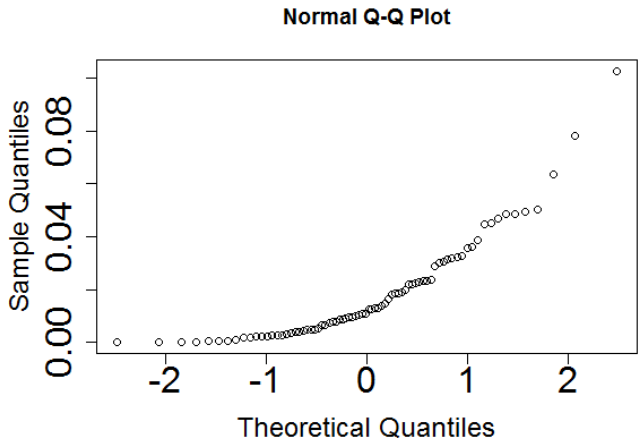


Figure 46: Outlier detection on the residuals of the Random Forest model with $M_{min} = 1.5$. Using Q-Q plot; We conclude a non-parametric test is appropriate.

Next we perform an autocorrelation check on the residuals, which together with the Wald-Wolfowitz test for randomness allows us to check if the MMPs are approximately i.i.d.. If that were not the case a correlation correction is required. Figure 47 below shows the residuals autocorrelation for the Random Forest with $M_{min} = 1.5$. As could be seen in the autocorrelation

plot there is no major correlation between the residuals or indication of any apparent trends or seasonal biases.

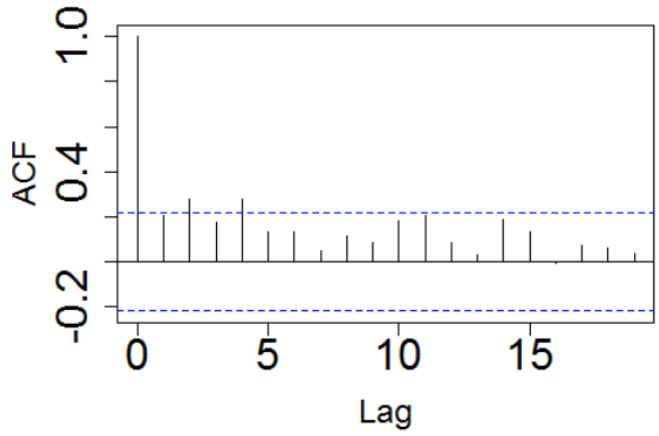


Figure 47: Autocorrelation plot of the residuals of the Random Forest model with $M_{min} = 1.5$.

We proceed to the more formal Wald-Wolfowitz test which has been described in section 5.4, allowing us to accept the null-hypothesis that the residuals of the Random Forest with $M_{min} = 1.5$ are random (p-value of 0.2544).

The same outlier, autocorrelation and randomness tests are also applied to the model-to-be-compared-with (here: the best baseline). If either of the models is tested as non-random we test additionally whether the difference between model residuals is random. The results for all targets is shown in Table 18 below – for all targets the Wilcoxon signed rank test is appropriate.

M_{min}	Outliers	ACF RF	Random Residuals RF	Random Residuals RF-Baseline	Decision
1.5	Yes	< 0.45 correlation with lags	Yes (p-value = 0.2544)	Yes (p-value = 1)	Wilcoxon signed rank test
1.2 (1995)	Yes	< 0.2 correlation with lags	Yes (p-value = 0.3619)	Yes (p-value = 1)	Wilcoxon signed rank test
1.2 (2004)	Yes	< 0.2 correlation with lags	Yes (p-value = 0.7603)	Yes (p-value = 0.5418)	Wilcoxon signed rank test
1.0	Yes	< 0.2 correlation with lags	Yes (p-value = 0.4095)	Yes (p-value = 1)	Wilcoxon signed rank test

Table 18: Forecast performance test selection overview for each of the targets (rows). Decision on which test to use is shown in the rightmost column.

Having selected the Wilcoxon signed rank test, we proceed to the key error metrics of the MMPs as shown in Table 19 for $M_{min} = 1.5$ and Table 20 for $M_{min} = 1.2$. For each target the top four

robust machine learning models, the best statistical baseline and the best physical baseline are shown. In case different metrics result in different rankings the MAE has been used as the guiding metric. For reference we provide the results for all error metrics in Appendix 6.

$M_{min} = 1.5$ $T_{start} = '95$ $T_{agg} = 3\,\mathrm{m}$	Model: Random Forest	Model: SVM	Model: KNN	Model: GLM Top	Baseline: Moving Average	Baseline: Depletion Moving Average
MAE	0.018±0.002	0.019±0.002	0.019±0.002	0.020±0.002	0.019±0.002	0.020±0.002
RMSLE	0.025±0.003	0.025±0.003	0.026±0.003	0.026±0.003	0.025±0.002	0.026±0.002
R^2	0.209±0.183	0.180±0.187	0.158±0.192	0.117±0.212	0.214±0.158	0.159±0.153

Table 19: Error metrics of the best four models for meta parameter setting FC01-1.5 for the target $M_{min} = 1.5$, $T_{start} = '95$ and $T_{agg} = 3$ months. Error metrics for the best statistical and best physical baseline are also shown. In case rankings differed for various metrics the MAE has been used as guiding metric.

$M_{min} = 1.2$ $T_{start} = '95$ $T_{agg} = 3\,\mathrm{m}$	Model: Random Forest	Model: KNN	Model: SVM	Model: GLM Top	Baseline: Moving Average	Baseline: Depletion Moving Average
MAE	0.025±0.003	0.028±0.003	0.028±0.003	0.028±0.003	0.028±0.003	0.029±0.003
RMSLE	0.032±0.003	0.034±0.003	0.034±0.003	0.033±0.003	0.034±0.003	0.038±0.004
R^2	0.469±0.109	0.411±0.115	0.381±0.119	0.436±0.103	0.384±0.116	0.259±0.147

Table 20: Error metrics of the best four models for meta parameter setting FC35-1.2 for the target $M_{min} = 1.2$, $T_{start} = '95$ and $T_{agg} = 3$ months. Error metrics for the best statistical and best physical baseline are also shown. In case rankings differed for various metrics the MAE has been used as guiding metric.

In light of the different typical event rates for $M_{min} = 1.5$ (Table 19) and $M_{min} = 1.2$ (Table 20) the MAE and RMSLE results between both tables cannot be straightforwardly compared. The R^2 metric can be compared and doing so shows an increase in explanatory power for all models and all baselines from $M_{min} = 1.5$ to $M_{min} = 1.2$, as expected given the increase in the number of events available between $M_{min} = 1.5$ and $M_{min} = 1.2$.

For illustrative purposes, using an absolute order on the error metrics and not taking uncertainty estimates into account, for $M_{min} = 1.5$ we note that the Random Forest is slightly better than the best baseline (the moving average) in the MAE metric but the opposite in the R^2 metric. All other machine learning models perform similar or slightly worse than the best baseline in the MAE metric. For $M_{min} = 1.2$ the Random Forest outperforms the best baseline (moving average with automatically tuned window) in all metrics. The other machine learning models perform again similar to the best baseline in the MAE metric.

For $M_{min} = 1.5$, a one-sided Wilcoxon signed rank test shows that the Random Forest is ever so slightly not statistically significantly better ($p = 0.058$, test statistic $V = 1292$) than the best baseline at a significance level of 0.05. We note the unpaired Wilcoxon rank-sum test is not significant, with $p = 0.218$ (test statistic $W = 2971$). Furthermore, we note that the exact value of these statistics depend on the chosen experimental setup, a slight change in this setup will impact model-baseline differences. Since the MAE difference between the Random Forest and the Moving Average is smaller than the standard error of each, a slight change in the experimental setup will directly impact the p-value and the associated hypothesis test, for better or worse. With the statistical significance of the test depending on the experimental setup, we conclude that more events would be required to reach a more definite conclusion. Visual inspection of Table 19 and Table 20 shows that the other machine learning models have a near identical MAE as that of the best baseline, with the MAE standard error being much larger than the MAE differences. Without formal Wilcoxon test we can therefore conclude that the other machine learning models are not statistically significantly better than the baseline.

For $M_{min} = 1.2$ we note that the Random Forest is statistically significantly better than the Moving Average ($p = 0.005$, test statistic $V = 1082$) in the experimental setup used for this report but based on analogous reasoning as above, we note that more events would be required to reach a more definite conclusion. The unpaired test doesn't show statistically significant results ($p = 0.180$, $W = 2931$). For the other machine learning models we observe they are not statistically significantly better than the baseline without further formal tests.

10.2 Qualitative Evaluation: Forecast Behaviour over Years

The previous section showed that the selected models in general have similar quantitative performance, despite their very different functional forms. Each individual model has been trained out-of-sample via the walk-forward approach explained in section 5.1. This involved iterative forecasting and retraining in quarterly or monthly periods – hence the models have certified forecast performance for forecasting smaller periods ahead. For PSHRA forecasts for at least five years ahead are required – hence it is important to understand the qualitative behaviour of seismic event rates as forecasted by the various functional forms the various models represent. In particular, it is interesting to understand how the forecasts of various selected models qualitatively compare on the longer term, to each other and to the default PSHRA forecast. Here, we consider forecasts for two future production scenarios: (i) the Production Plan 2016 production policy scenario (NAM, 2016) and (ii) the post-March 2018 production scenario (Ministry of Economic Affairs and Climate, 2018).

For the Production Plan 2016, all selected machine learning models forecast relatively stable seismicity event rates for the coming five years, illustrative examples are shown in Figure 48 and Figure 49. By visual inspection, the qualitative behaviour seems relatively in line with the default PSHRA statistical physics based forecast (Bourne & Oates, Development of statistical geomechanical models for forecasting seismicity induced by gas production from the Groningen field, 2017), although the default PSHRA forecast seems to be at the higher end of the confidence interval of this forecasts of this study.

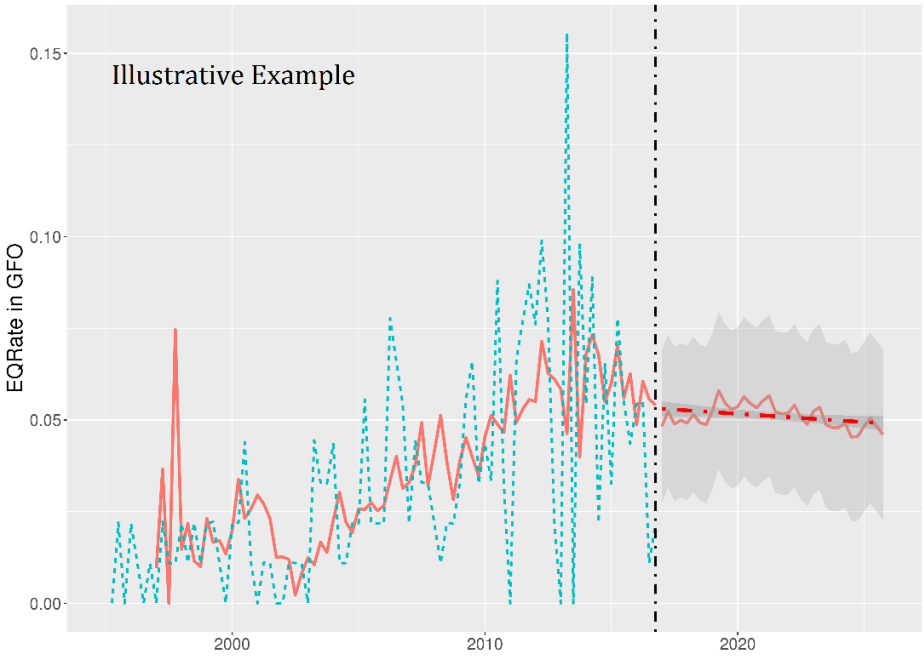


Figure 48: Illustrative example of a Machine Learning based seismicity event rate forecast for the Groningen field, being the GLM Top forecast for $M \geq 1.5$ for the Production Plan 2016 default production scenario. The figure shows the expected daily seismicity rates per quarter. The vertical dotted-dashed line is at December 31st 2016, marking the end of the dataset used for training and testing the models. The historical seismicity rates are shown by the blue dotted line and the algorithm forecasts are shown by the red solid line as of the minimum number of points. Left of the vertical line the algorithm is retrained after every forecast, right of the vertical line no retraining is done. The shaded grey area shows the 0.9 confidence interval – we note that the limitations as outlined in section 5.6 apply. The dashed red line is a forecast trendline with its uncertainties shown by the shaded dark grey area.

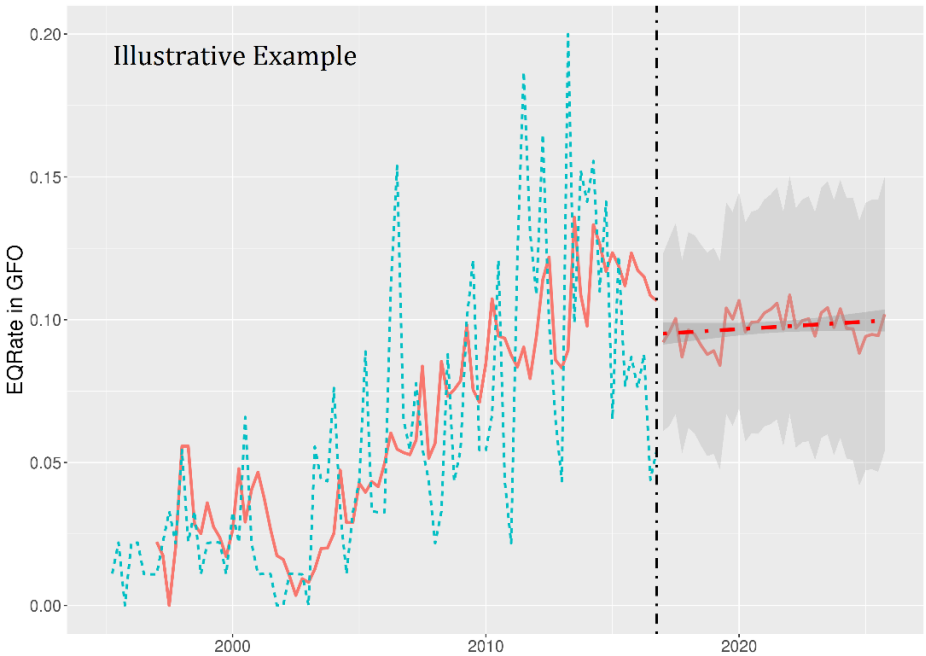


Figure 49: Illustrative example of a Machine Learning based seismicity event rate forecast for the Groningen field, being the GLM Top forecast for $M \geq 1.2$ for the Production Plan 2016 default production scenario.

For the post-March 2018 average policy scenario the forecasts for the selected machine learning models qualitatively diverge. Models which cannot extrapolate like the Random Forest, KNN and Moving Average baseline all forecast a relatively stable (for KNN even an increasing) event rate for the coming five years, contrary to expectations as production reduces with nearly 40% in this period. More on this in the next section, for now we suffice with the note we do not consider these forecasts in line with the boundary conditions imposed by physics. Based on visual inspection, models which can extrapolate like the SVM, GLM Top and Depletion Moving Average (DMA) baseline do forecast a (modest) decline, as illustrated for the same illustrative example model as for the Production Plan 2016 in Figure 50. For this scenario, the work-in-progress PSHRA seismicity event rate forecasts available at the time of writing (June 2018) are in apparent qualitative agreement with the extrapolating machine learning models for the period 2017-2021, but PSHRA event rate forecasts decline substantially faster after 2021.

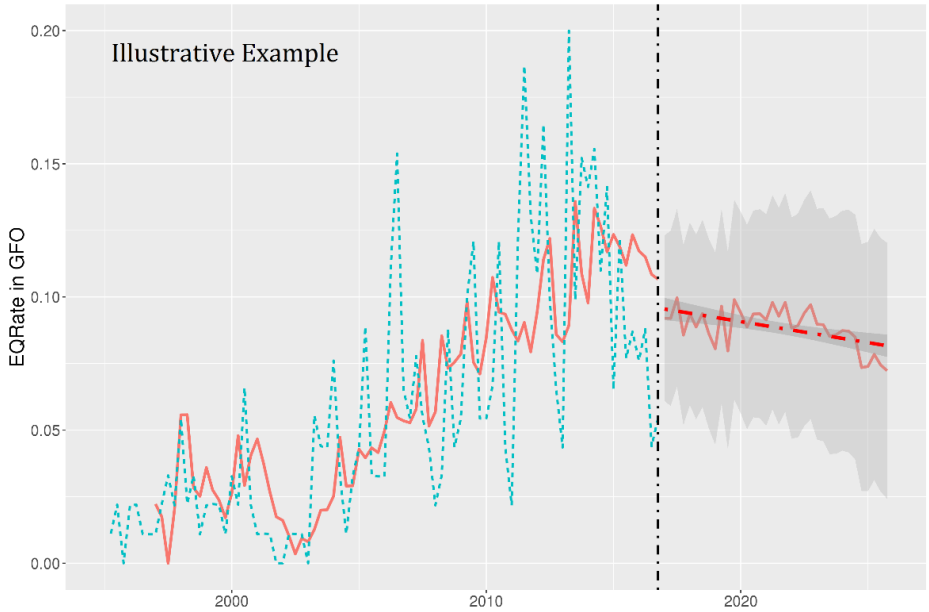


Figure 50: Illustrative example of a Machine Learning based seismicity event rate forecast for the Groningen field, being the GLM Top forecast for $M \geq 1.2$ for the average post-March 2018 production scenario.

For completeness, we note that the confidence bands are generated using the approach outlined in section 5.6. This comes with two potential caveats: first, the use of quantiles may lead to relatively high variability in the uncertainty estimates. Second, the confidence band estimates become increasingly uncertain the further we move in the future due to the decreasing number of experiments that can be carried out for the respective forecast window size.

10.3 Range of Validity

Every theory or methodology has its range of validity – the methodology described in this report is no exception. In this section we describe the conditions under which the methodology is valid and the type of problems we can expect if the conditions certifying validity are no longer fully met. We identify three key aspects influencing the range of validity: (i) the interval length after which models are evaluated and retrained; (ii) out-of-bound feature handling; (iii) model selection rules applied. We will discuss each of these aspects in more detail.

- **The interval length after which models are evaluated and retrained** is determined by a trade-off between two factors: (i) predictive power and (ii) the assumed similarity between shorter term and longer term forecasts. As uncertainty generally increases with time,

evaluating and retraining models on shorter time scales increases predictive power and hence, the ability to statistically distinguish between models. The evaluation strategy as reported on here is geared towards maximum statistical power and uses an evaluation and retraining period of 1 to 3 months, when uncertainties are still relatively small. Models which perform well for these short term (1-3 month) forecasts are downselected. An implied (although not a technical fundamental) assumption is that short term forecast performance is indicative for longer term forecast performance, i.e. for PSHRA between one to five years. This assumption might hold if past feature values are relatively similar to future feature values but doesn't hold otherwise, and in particular it doesn't seem to hold for the post March 2018 production scenario.

- **Out-of-bound feature handling** is relevant specifically for non-extrapolating models, for the selected models this concerns the RF and KNN. Within the physical context of this study, forecasts for non-extrapolating models are only valid when the future values of the feature set are contained in the convex hull of the past feature set values. In practice and in the spirit of (Breiman, Statistical Modeling: The Two Cultures, 2001) mild out-of-boundness is acceptable as long as it doesn't impact forecast performance. The degree by which out-of-boundness impacts forecast performance depends on (i) the fraction of future time series which are out-of-bound; (ii) the amount by which they are out-of-bound; (iii) the importance of the feature within the overall model. We know that several features in our setup are monotonically evolving features (e.g. features like P, C and HCT), so they will always be outside the convex hull of past observations. As shown in Appendix 8 for the RF model, it turns out that monotonically evolving features are amongst the key drivers of the model – hence within the current setup non-extrapolating models might be strongly challenged. It might also contribute to the qualitatively diverging forecast behaviour for the post-March 2018 production scenario – where in particular the non-extrapolating models don't forecast a declining seismicity trend.
- **The model evaluation and selection criteria** are mathematical criteria: (i) minimum forecast error; (ii) maximum variance explained and (iii) robustness of results under small changes of meta parameter values. None of these criteria guarantee forecast behaviour in line with (high-level generally agreed upon) physical expectations. As an illustrative example, there is a maximum amount of energy contained in the physical system, which places a (magnitude dependent) maximum on the number of earthquakes possible. The models will not be aware of this unless this information is explicitly provided.

11 Conclusions and Discussion

This study is part of NAM's Study and Data Acquisition Plan in the context of the Measure and Control Protocol. The goal of this study is to develop a machine learning based methodology to forecast production induced seismicity event rates for the Groningen Field. The methodology allows probing of a wide variety of possible linear and non-linear combinations and interaction terms between physical variables without assuming a priori knowledge on the nature of the relationships between these variables. A two-step approach is employed: a factorial experimental setup followed by meta analysis (analysis of the effectiveness of the experimental setup) is used to select robust and relatively well performing models and meta parameters. The selected models and meta parameters are used for seismicity event rate forecasts.

The event rate forecasts are evaluated in three ways: (i) quantitatively; (ii) qualitatively and (iii) the range of validity. Quantitatively, we note that with the data used in this setup, in general the machine learning models are not statistically significantly better than baseline models. Qualitatively, we observe that for the Winningsplan 2016 [Production Plan 2016] the selected models and meta parameters forecast a relatively stable seismicity event rate for the coming five years, in line with the default PSHRA statistical physics based forecasts. For the average production scenario announced by the Ministry of Economic Affairs and Climate in March 2018 [hereafter the average post-March 2018 production scenario] model behaviour diverges qualitatively. Models which can extrapolate (SVMs, GLM variants) forecast a modest decline, though the default PSHRA event rate forecasts decline substantially faster after 2021. Models which cannot extrapolate (RFs, KNNs) have difficulty with this future scenario and forecast a stable or even increasing event rate – we consider these forecasts to be unphysical.

The range of validity of the methodology described in this study is influenced by three key aspects, which all might play a role in the unphysical forecasts for the post-March 2018 production scenario. First, the model evaluation strategy is geared towards maximum statistical power, thereby decreasing uncertainties and hence improving the ability to statistically distinguish between the forecasts of various models. In particular, models are evaluated and retrained after each step forward when uncertainties are still relatively small. An implied (although not technically fundamental) assumption is that one step forward (1-3 months) forecast performance is indicative for many steps forward (1-5 years) forecast performance. This assumption is not always satisfied and may lead to selecting models which perform well on the short term but not on the long term. Second, physical variables like P, C and HCT are monotonically evolving features and thus are guaranteed to go out-of-bounds of the convex hull of the past values of the feature set. Given the fact that some of these features are expected (and for RFs are shown) to play a major role, this poses an elevated challenge for non-extrapolating models which might result in unphysical behaviour.

Third, the model and evaluation selection criteria are mathematical criteria and as such are not bound to (high-level generally agreed upon) physical expectations.

Three concrete steps which could mitigate the limitations on the range of validity are:

- Investigate usage of longer term (1-5 years) forecasts to validate model performance, instead of the short term (1-3 months) forecast performance evaluations used in this study.
- Update the feature set of non-extrapolating models such that only features whose future values won't exceed the convex hull of historical feature set values are used.
- Extend the model evaluation and selection criteria with rules encoding high-level physics based boundary conditions.

Regarding the definiteness of the conclusions reported above, we note that in light of the (from a machine learning perspective) relatively limited number of events all data available at the start of this study is used for model meta analysis (and thus model selection). Although various safeguards have been placed, forecast performance estimates might be on the optimistic side and a hold-out set is required to validate these estimates. Ideally, the training and testing period of the hold-out set ends around a moment that the production strategy changes. Models which forecast an appropriate change in seismicity following the change in production strategy probably captured underlying mechanisms driving seismicity better than models which don't. Two approaches are available. First, a validation set will be obtained naturally over time. An appropriate cut-off moment between training/testing and validation might be before the post-March 2018 production scenario is enacted. As this approach has the disadvantage it will take quite some years to obtain a large enough validation set, second, training/testing the model up to the production shut-ins following the Huizinge earthquake in 2012 and using the remaining years for validation might work as well. A disadvantage of this second approach is that it roughly halves the number of events, which might impact our ability to discriminate between models.

In light of the above, the authors advise that pending more definite conclusions on (longer term) forecast performance the models should not be used for business decisions.

Next steps:

To further improve machine learning based seismicity forecasts for the Groningen field and to follow up on the leads from this study three suggestions are presented:

1. Extend the range of validity of the methodology and the definiteness of conclusions by progressing the suggestions mentioned above.
2. Investigate the forecast performance gain which hybrid models combining physics and machine learning models could provide.
3. Extend the event rate methodology developed in this study to include areal and magnitude forecast capabilities.

On several aspects of our approach further discussions are insightful, we address these per chapter:

Chapter 3: Data: Sources and Features

- **Feature selection**: the features used in this study are simple physical quantities (e.g. P, C, …) or their temporal or geospatial derivatives. More geomechanics informed features representing e.g. visco-elastic stress relaxation, visco-elastic deformation, rate friction, state friction etc. might help improve forecast performance and quantify the relevance of the underlying physical process with respect to seismicity.
- **Aggregation functions**: for most features either the average (e.g. reservoir pressure) or the sum (e.g. production) were used as spatial aggregation functions. Different aggregation functions (e.g. the max reservoir pressure) could change forecast performance – this has not been explored.
- **Target definition**: the study investigates event rates defined as the number of earthquakes per unit of time above a minimum magnitude. Different definitions of event rates could be used, e.g. the number of earthquakes per unit of production or unit of subsidence instead of time. Alternatively, quantities like the earthquake energy released (which combines counts and magnitudes) could be investigated.

Chapter 4: Methodology: Defining Meta Parameters

- **Meta parameter time delay**: analogously to the references mentioned in section 2.2 we find evidence for time delays for production Q and reservoir pressure P, but the differences in forecast performance between models with and without time delay are very small and statistically insignificant for all targets we investigated. We note that this study uses one time delay for each physical quantity and all its derivatives whereas e.g. (Pijpers, Trend changes in tremor rates Groningen - update Nov. 2016, 2016) found different time delays for a physical quantity and its temporal derivatives. Furthermore, we note that in the current setup one time delay is assumed for the entire field, whereas changes in some quantities (e.g. reservoir pressure P) propagate through the field in the order of months. A location dependent time delay might increase the impact time delays have on forecast performance, although with the typical delay of e.g. reservoir pressure of around 3 months (the same period as the aggregation period used in this study) the effect is probably limited.
- **Meta parameter lag**: the usage of lags has been explored and lags seem to add predictive power – hence we use non-zero lags as part of the final meta-parameter setup shown in Table 14. A more detailed analysis of the impact of lags, including multi-segment lags, might improve predictive performance.

Chapter 5: Methodology: Evaluating Model Performance

- **Errors exogeneous variables**: calculating historical forecast errors does not take into account the variability and forecast errors of some of the model's exogeneous variables. The different features used (e.g. P, Q, C) don't necessarily have the same precision for future forecasts as they had for the historical part of our dataset. Consequently, we are underestimating the true forecast error. To take these exogeneous uncertainties into account, one would need to take forward uncertainties per data sources. E.g. for the dynamic reservoir data one could take the P10, P50 and P90 realizations of the reservoir model MoReS and use these for forecasts. We note though that only a limited set of machine learning models supports this type of fully probabilistic modelling.
- **Test statistic choice**: the hypothesis testing procedure used to select robust models is based on the paired Wilcoxon signed rank test. As discussed in 5.4, paired tests have higher statistical power compared to their unpaired counterparts due to reducing variance. However, they possess a potentially higher false positive (type 1 error) rate. The choice then becomes a trade-off between gains in statistical power and keeping type 1 errors under control. Since we are interested in finding new leads in the relationships in the data that may have remained undetected till this moment in time we tend to favour paired tests but note that an elevation in type 1 errors can potentially lead to falsely declaring method A to be statistically significantly better than method B.
- **Forecast uncertainty estimates**: our procedure for enforcing the monotonically increasing forecast errors has limited accuracy in terms of fit on the forecast errors and is computationally expensive – improvements are possible on both points.
- **Forecast uncertainty estimates**: on top, as our dataset is finite the estimation of larger forecast horizons is increasingly associated with models that were built on fewer data points, and might include some models where the parameters still suffer from instability. This might lead to overestimating the true forecast error compared to the actual implementations of algorithms of this type.

Chapter 6: Methodology: Machine Learning Models

- **Model tuning**: limited time was available for model tuning. Spending more time here might improve machine learning model performance, in particular for models which are known to be sensitivity to tuning, e.g. Neural Networks and Gradient Boosting Machines.
- **Neural network variants**: even though we have covered a basic neural network (to be precise a Feed Forward Neural Network, FFNN), we have not investigated a specific form of neural networks that has a form of built-in memory and is thus unlike feedforward neural networks not stateless. These so-called recurrent neural networks (RNNs) are potentially more suitable to handle time series data since they can capture time dependent interaction between covariates. One form of RNNs, which have revolutionized areas like speech recognition and robot control, are so-called long short-term (LSTM) memory networks. Conceptually, LSTMs mimic short term memory in biological neural networks. Unlike FFNNs they have memory cells that can store previously computed results in earlier time steps for a variable amount of time. Different architectures for those memory cells exist. They could warrant further investigation. Further details can for instance be found in (Hochreiter & Schmidhuber, 1997).
- **Extrapolating Random Forests**: Random Forests can be extended with extrapolation capabilities, for example by adding GLMs to leaf nodes. We didn't include such models so far given their technical complexity combined with non-proven performance, but such models might provide useful for the post-March 2018 average policy production scenario.

Chapter 10: Evaluation of Machine Learning based Seismicity Event Rate Forecasts

- **Predictive power**: as discussed in section 10.1, whether or not probabilistic models (like Random Forests) are significantly better than the best baseline in the experimental setup of this study depends on experimental setup details, e.g. the aggregation starting month (January, February or March). This reflects the relatively large model uncertainties, which in turn are related to amongst others the (statistically speaking) limited number of events. Additional events are required to reach a more definite conclusion on the statistical significance of the results of the probabilistic models.

12 Next Steps

In previous chapter three key steps have been identified to improve the methodology developed in this study. The first of these, extending the range of validity and the definiteness of conclusions, is already discussed in the previous chapter. In this chapter we further detail the next steps:

2. Develop hybrid physics + machine learning models.
3. Extend the event rate methodology developed in this study to a full PSHRA compliant methodology.

12.1 Developing hybrid machine learning + physics models

Physics based models and machine learning models have different starting points: physics based models are rooted in empirical physics theory, whereas machine learning models are based on functional approximations. These different starting points give both type of models some complementary strengths and weaknesses: e.g. physics based models provide insights in why certain effects happen, but don't handle unknown factors well – the opposite is true for machine learning. Additionally, machine learning based models may be poor at extrapolating. Hybrid Physics+ML models combine both methodologies to potentially get "the best of both worlds", see e.g. Table 21.

Model type	Physics model	Machine learning model	Hybrid Physics+ML model
Completeness	Physics theory provides several key factors, but unknown factors can remain	Observed data contains known and unknown factors, but no separation known and unknown	Physics theory provides several key factors, remaining unknown factors modelled with ML
Input-output relation	Outputs are directly understandable given inputs	Outputs are not directly understandable given inputs	Outputs are more interpretable than in a pure ML setup
Target units	Physics based units	Normalization removes feature units	Physics based units
Error handling	Errors are intrinsic due to uncertainty of subsurface	Errors are algorithmically taken care of without adding to understanding	Errors illustrate physics theory or measurement anomalies that can add to overall understanding
Error accumulation	In case of multi-stage models, each stage needs to be accurate or errors accumulate	Errors are summed and taken care of in the model	Errors are handled in the model with assumption they are significant

Table 21: overview of some complementary properties of physics models (left), machine learning models (middle) and hybrid Physics+ML models (right).

An overview of physical seismicity analysis has been given in section 0. Two of the approaches discussed in that section are:

1. Statistical physics models, which have forecasting capabilities, most prominently the default PSHRA model (Bourne & Oates, Extreme Threshold Failures Within a Heterogeneous Elastic Thin Sheet and the Spatial-Temporal Development of Induced Seismicity Within the Groningen Gas Field, 2017);
2. Deterministic geomechanical models, e.g. 2D rupture models as described in (Van den Bogert P. A., 2015).

Both type of physics models could be combined with machine learning to generate Hybrid Physics+ML models. To concretize this, below we sketch some ideas on how machine learning could potentially contribute in the references mentioned above.

With respect to the default PSHRA model, machine learning can potentially contribute as follows:

1. Provide insights in the elasticity of the thin sheet models for reservoir deformation. Reservoir stresses & strains, the thin sheet topography and fault properties can be related with seismicity. This relation might provide insights in a possible non-linear relation between Coulumb stress and seismicity. In turn, this would provide insights in the elasticity of the earth (e.g. whether the earth behaves in approximation like rubber or foam).
2. Investigate different functional forms for the probability to exceed an extreme threshold (e.g. Extreme Value Theory). This would provide additional insights in the initial distribution of Coulumb stresses.
3. Reveal missing information in the default PSHRA model via residual analysis. Analysis of the prediction residuals of the default PSHRA model with Machine Learning might indicate missing seismicity drivers and thus provide additional insights to improve the default PSHRA model.

With respect to the deterministic geomechanical rupture model, machine learning might contribute by:

1. Constraining fault and formation properties across the Groningen field. This could be done using seismic event catalogue and geomechanical relationships derived from dynamic rupture simulations, integrate it with experimental results and work towards cumulative probability curves.
2. Work towards a fault-based seismological model with forecasting capability for frequency and magnitude, accounting for uncertainties and variability across the field.

12.2 Develop the machine learning event rate forecast methodology to a full PSHRA compliant methodology

The event rate forecast methodology as developed in this study cannot serve as a full alternative model within the context of PSHRA. To do so, the following extensions are required:

1. **Addition of geospatial resolution**: for hazard and risk information on the spatial distribution of seismicity is a must. Furthermore, inclusion of geospatial resolution will enable (more effective) addition of several data sources not (fully) accessible so far, including fault data and some of the data mentioned above. The trade-off of an increased geospatial resolution will be a lower temporal resolution.
2. **Addition of magnitude resolution:** for hazard and risk information on the magnitude of seismicity is a must, which so far beyond a minimum magnitude iss not available in the current approach. Forecasting various magnitude bins via e.g. probability mass functions might be possible but that effectively means making more forecasts. Given the relatively limited amount of earthquakes available it is unclear at the time of writing whether

probability mass function forecasts will be able to reach meaningful levels of forecast performance. An alternative more straightforward approach might be to impose a Gutenberg-Richter law based on an observed b-value on the forecasted number of earthquakes.

3. **End-2-end integration:** chapter 3 details the data sources used for this analysis. Except for historical production all data sources used in this study concern processed data, often in the form of output from geophysical or statistical models. For example, the reservoir pressures come from MoReS, which uses advanced reservoir simulation techniques to extend the pressure measurements at the wells to the entire field. Similarly, the subsidence and compaction models are interpolated from measurements to the entire field using models from Shell Statistics. Building models on models has the risk of suboptimal end-2-end results, as each model optimizes for its specific purpose not for the end result. An overarching end-2-end integration framework could help to ensure optimization for the end result.

Appendix 1. Data source exploration

Chapter 3 described the data source measurements, uncertainties and features. As extension to that chapter, this appendix offers a high level data exploration for the temporally varying data.

A1.1 Earthquake Data

The first earthquake was detected in 1986 nearby the city of Assen. Since then, the number of earthquakes increased both in frequency and intensity, as can be seen in Figure 51. In total, our data set (up to December 2016) contains 1387 earthquakes, of which 973 are within the outline of the Groningen field. Of these, respectively 270, 479 and 634 have a magnitude equal or larger than 1.0, 1.2 and 1.5.

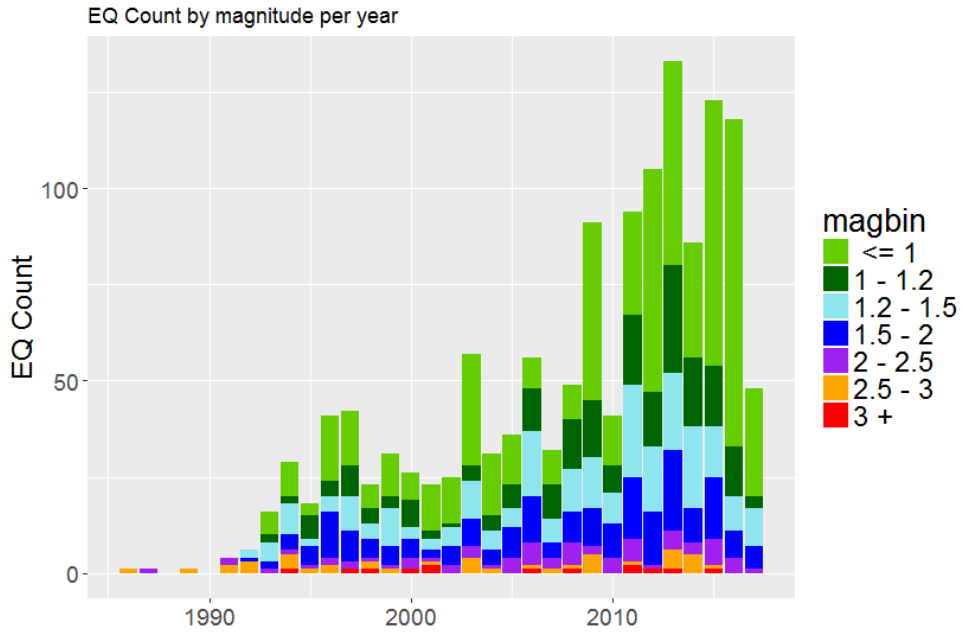


Figure 51: Earthquakes over the years by magnitude bins

In terms of geographical location a high earthquake frequency is visible in and around Loppersum and Slochteren, see Figure 52.

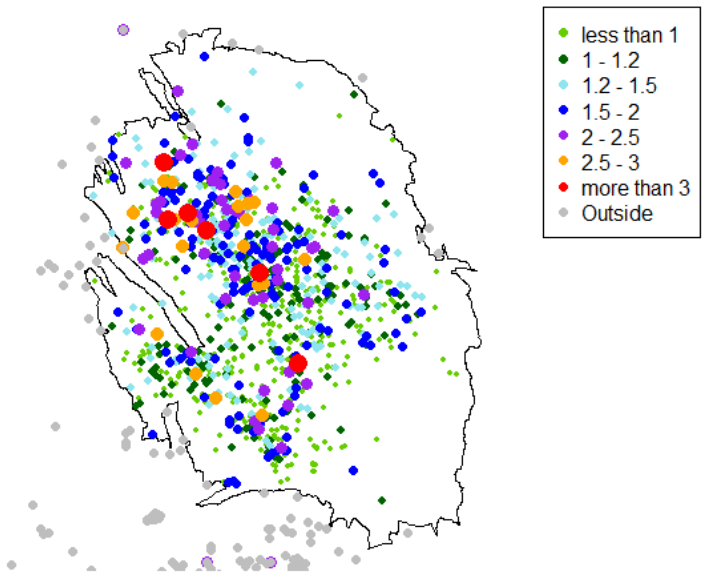


Figure 52: Geo-Location of Earthquakes by magnitude

A1.2 Production Data

Gas production commenced in 1956 and peaked in the 1970s. Recent years have seen a steep decline in gas production due to production caps. Figure 53 below shows the year to year gas production amounts since 1960 until 2017.

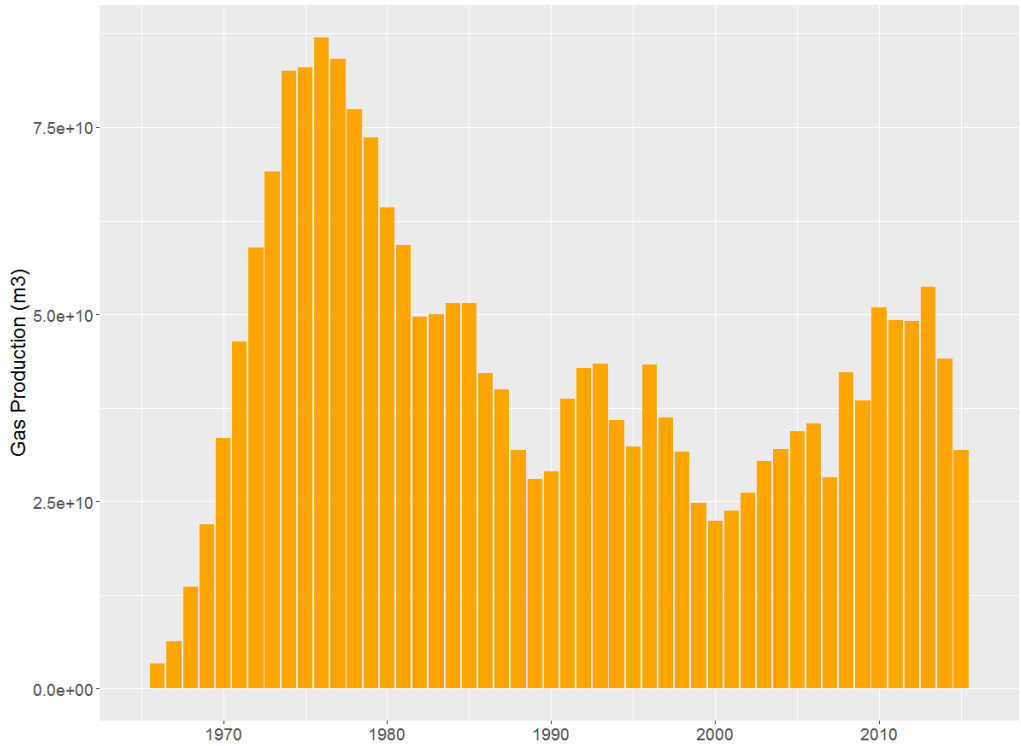


Figure 53: Yearly Gas production in the Groningen field, 1960-2017.

A zoom-in on the period under consideration in this study shown in Figure 53. The month to month variations highlight the historical demand driven seasonal production pattern. In recent years NAM begun to explore alternative production strategies with less seasonal variation.

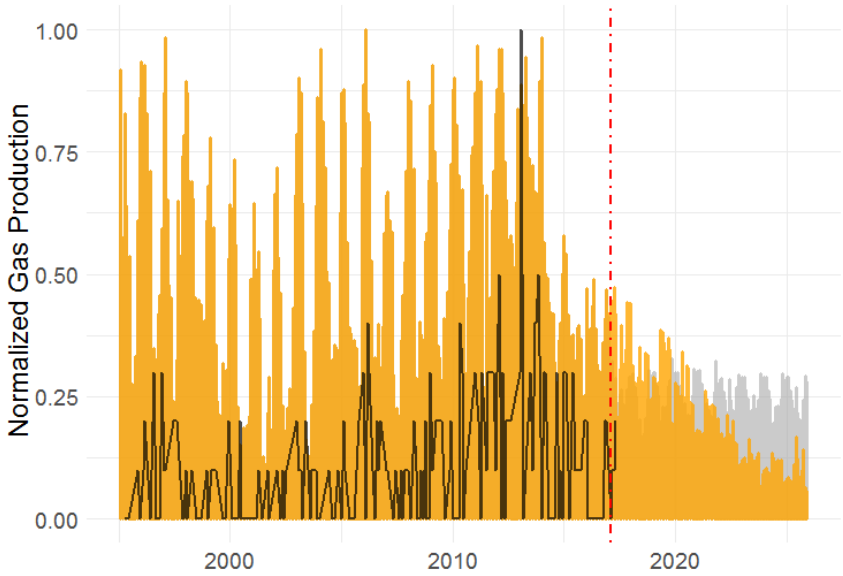


Figure 54: Normalized gas production (yellow) and earthquake rate ($M \geq 1.5$, black) per month in the Groningen field. Left from the red vertical line the historical values (1995-2016), right the production according to the post-March 2018 policy average production scenario (2017-2025) in yellow and the pre-March 2018 default production scenario in light grey.

A1.3 Dynamic Reservoir Data

The dynamic reservoir data concerns in particular the reservoir pressure, the reservoir hydrocarbon column mass (HCM) and hydrocarbon column thickness (HCT). Figure 55 shows the development of the dynamic properties during the period under consideration in this study.

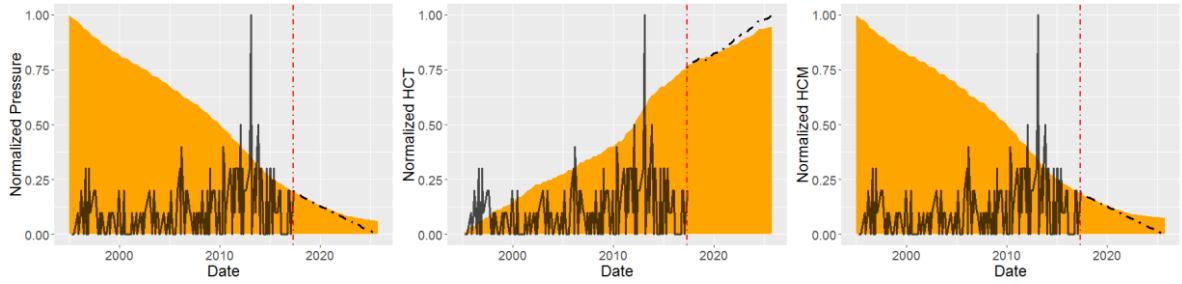


Figure 55: Normalized aggregated dynamic features in orange P (left) , HCT (middle) and HCM (bottom) and earthquake rate ($M \geq 1.5$, black) per month in the Groningen field. Left from the red vertical line the historical values (1995-2016), right the values as would result from the post-March 2018 policy average production scenario (2017-2025). The black dotted line right of the red vertical line shows the value under the former BP17 scenario.

The dynamic properties change in a smooth way over time, in line with expectations. The first order temporal differences of the dynamic properties are shown in Figure 56.

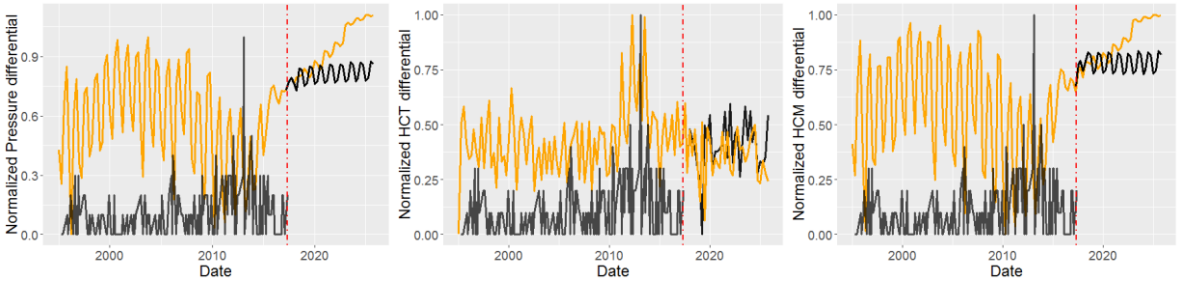


Figure 56: Normalized aggregated temporal difference dynamic features in orange $dPdT$ (left), $dHCTdT$ (middle) and $dHCMdT$ (right) and earthquake rate ($M \geq 1.5$, black) per month in the Groningen field. Left from the red vertical line the historical values (1995-2016), right the values as would result from the post-March 2018 policy average production scenario (2017-2025). The black line right of the red vertical line shows the value under the former BP17 scenario

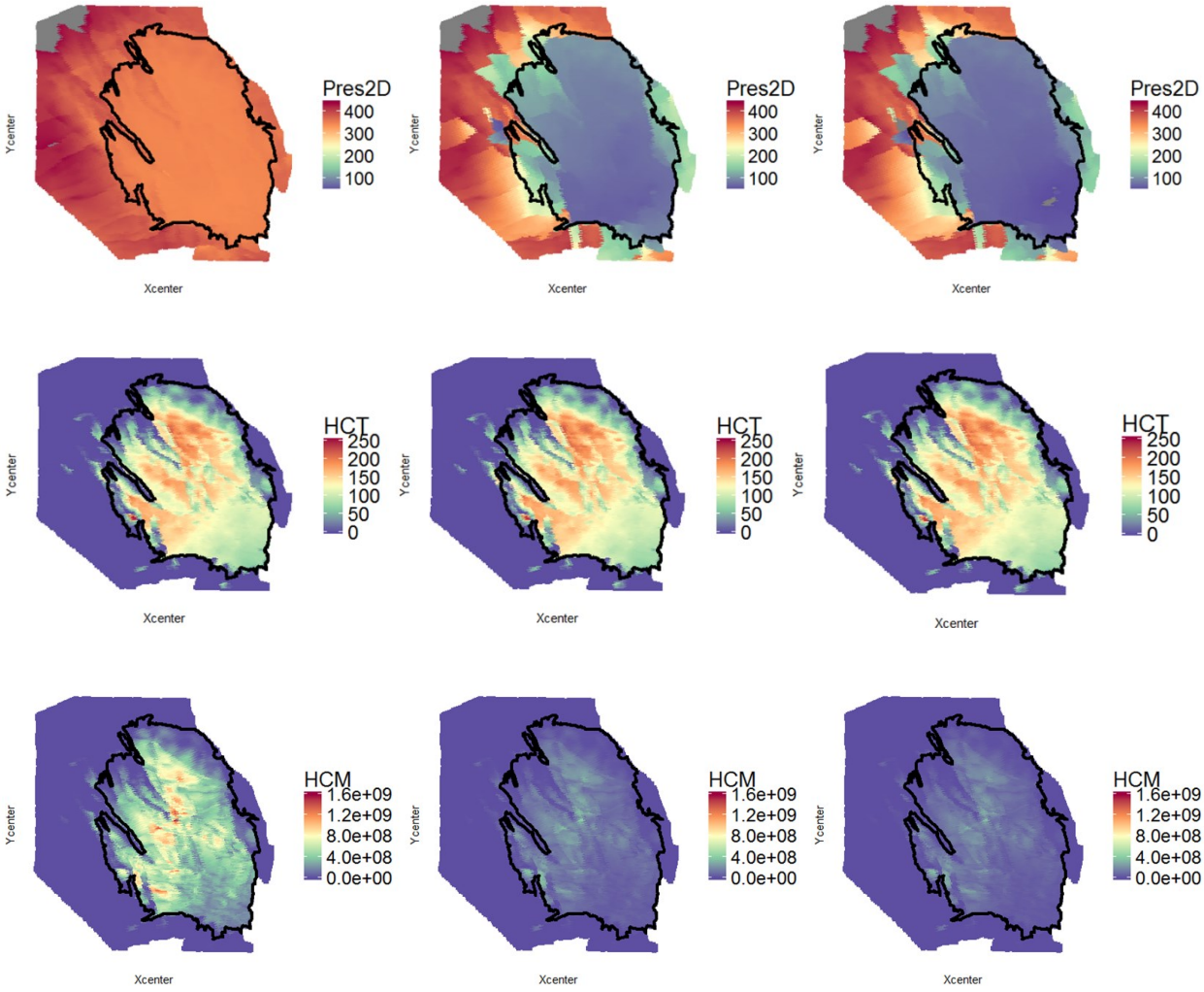


Figure 57: Geospatial overview of dynamic features, with P (top row), HCT (middle row) and HCM (bottom row) on January 1st 1995 (left column), January 1st 2017 (middle column) and December 31st 2025 (right column, based on the post-March 2018 policy average production scenario).

A1.4 Compaction Data

The compaction in the reservoir has been steadily increasing since January 30, 1958 from an average compaction across the reservoir of 0 m to an average reservoir-wide compaction of 0.1680 m on the 1st of January 2017. Compaction during the period under consideration is shown in Figure 58.

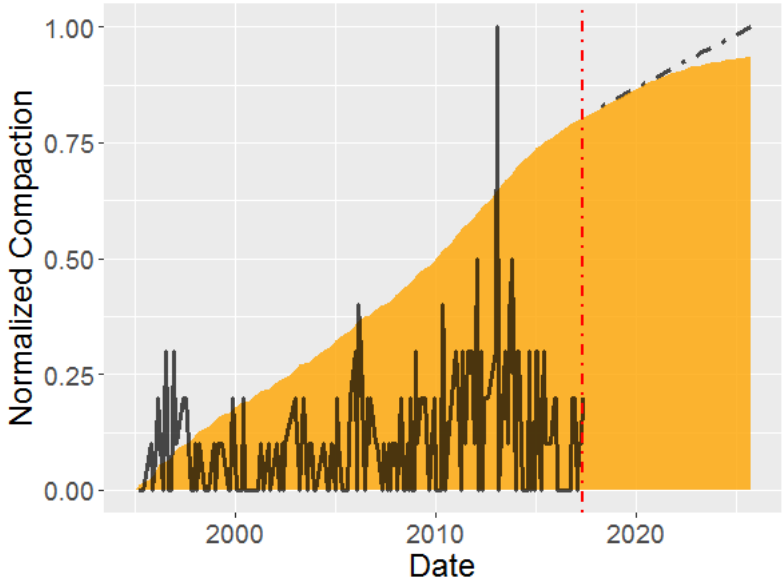


Figure 58: Normalized aggregated compaction (yellow) and earthquake rate ($M \geq 1.5$, black) per month in the Groningen field. Left from the red vertical line the historical values (1995-2016), right the values as would result from the post-March policy average production scenario (2017-2025). The black dotted line right of the red line indicates the values under the pre-March 2018 default production scenario.

We can also appreciate how the compaction pattern has progressed over time by looking at the gridded graphical representation of the data as seen in Figure 59. The main areas of compaction are in the central and northern areas of the reservoir with the latter becoming increasingly compacted also during the simulated future period between 2017 and 2025.

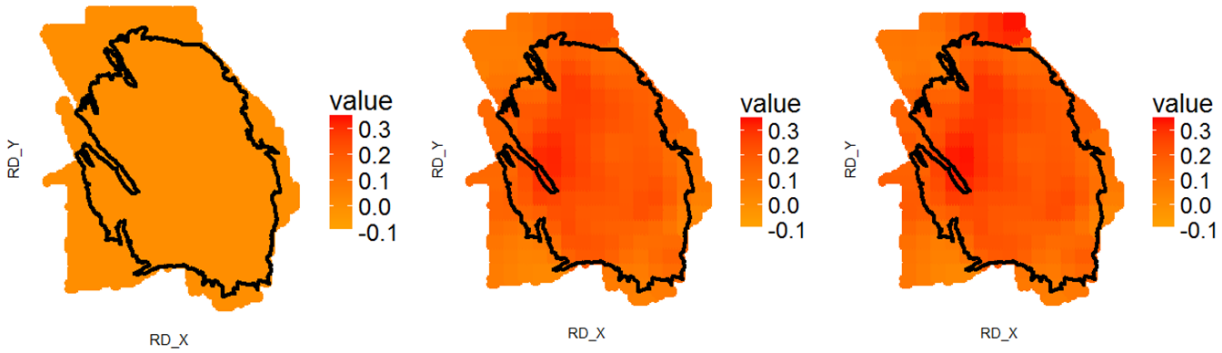


Figure 59: Geospatial subsidence patterns in 1958 (left), 2017 (middle) and 2025 (right, predictions as would result from the post-March 2018 policy average production scenario).

A1.5 Subsidence Data

Subsidence has been steadily increasing as pressure in the reservoir has decreased over the years due to production. In 1958 the mean subsidence above the reservoir was 0 m, which has increased to 0.21 m in 2017.The subsidence over the period under consideration is shown in Figure 60.

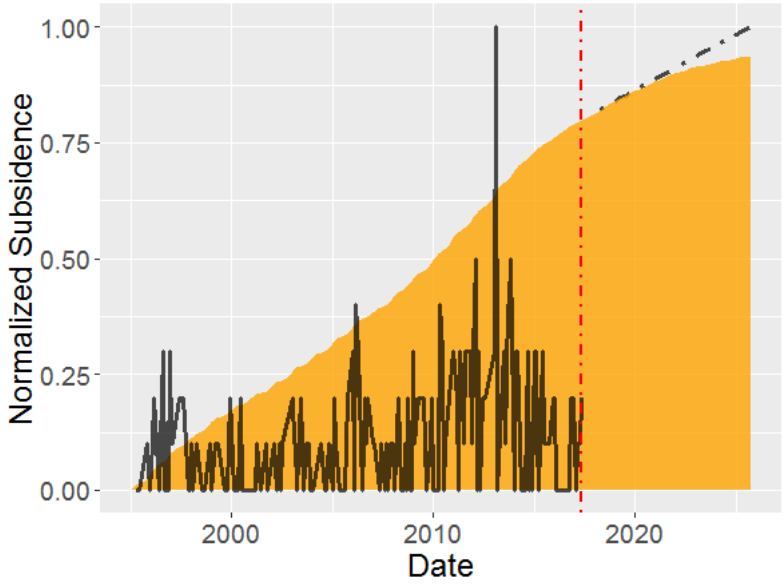


Figure 60: Normalized aggregated subsidence (yellow) and earthquake rate ($M \geq 1.5$, black) per month in the Groningen field. Left from the red vertical line the historical values (1995-2016), right the values as would result from the post-March 2018 policy average production scenario (2017-2025). The black dotted line right of the red line indicates the values under the pre-March 2018 default production scenario.

Geospatially, the zones of high subsidence match those of high compaction shown in previous section and appear mainly around the central area of the reservoir, see Figure 61 for a graphical illustration.

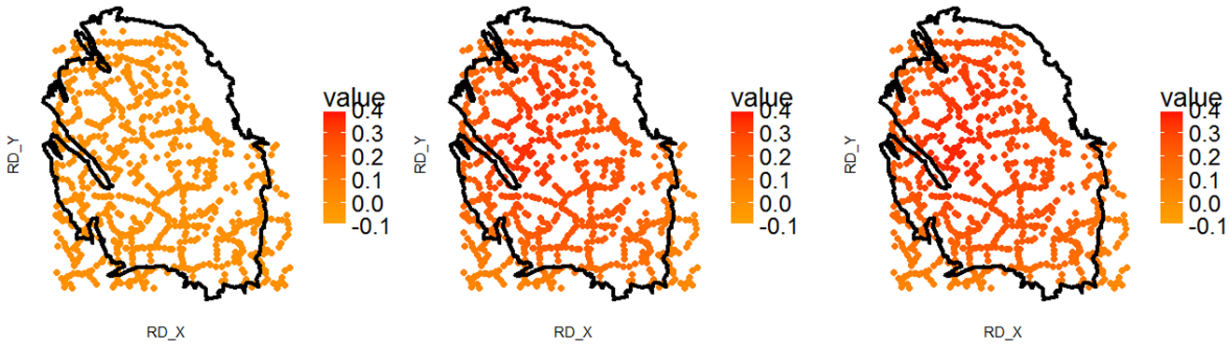


Figure 61: Geospatial subsidence patterns in 1958 (left), 2017 (middle) and 2025 (right, predictions as would result from the post-March 2018 policy average production scenario).

Appendix 2. Feature Correlation Groups

Section 4.4 describes the creation of feature correlation groups, the full set of groups are shown in Table 22 and Table 23 below. For each target for each correlation group, one representative feature is chosen.

Nr.	$M_{min} = 1.5$ $T_{start} = 1995$ $T_{agg} = 3$ months	$M_{min} = 1.2$ $T_{start} = 1995$ $T_{agg} = 3$ months
1	sum.Q.Gas.M3	sum.Q.Gas.M3
2	variance.Q.Gas.M3	variance.Q.Gas.M3
3	sum.dQdT.Gas.M3	sum.dQdT.Gas.M3
4	variance.dQdT.Gas.M3	variance.dQdT.Gas.M3
5	sum.d2QdT2.Gas.M3	sum.d2QdT2.Gas.M3
6	variance.d2QdT2.Gas.M3	variance.d2QdT2.Gas.M3
7	weighted.mean.P weighted.mean..FF.P weighted.mean.PL weighted.mean..FF.PL weighted.mean.HCM weighted.mean..FF.HCM	weighted.mean.P weighted.mean..FF.P weighted.mean.PL weighted.mean..FF.PL weighted.mean.HCM weighted.mean..FF.HCM
8	weighted.mean.dPdT weighted.mean.dPdTrel weighted.mean.d.FF.PdT weighted.mean.d.FF.PdTrel weighted.mean.dPLdT weighted.mean.dPLdTrel weighted.mean.d.FF.PLdT weighted.mean.d.FF.PLdTrel weighted.mean.dHCMdT weighted.mean.d.FF.HCMdT diff.mean.C.S	weighted.mean.dPdT weighted.mean.dPdTrel weighted.mean.d.FF.PdT weighted.mean.d.FF.PdTrel weighted.mean.dPLdT weighted.mean.dPLdTrel weighted.mean.d.FF.PLdT weighted.mean.d.FF.PLdTrel weighted.mean.dHCMdT weighted.mean.d.FF.HCMdT diff.mean.C.S
9	weighted.mean.d2PdT2 weighted.mean.d2PdT2rel weighted.mean.d2.FF.PdT2 weighted.mean.d2.FF.PdT2rel weighted.mean.d2PLdT2 weighted.mean.d2PLdT2rel weighted.mean.d2.FF.PLdT2 weighted.mean.d2.FF.PLdT2rel weighted.mean.d2HCMdT2 weighted.mean.d2HCMdT2rel weighted.mean.d2.FF.HCMdT2 weighted.mean.d2.FF.HCMdT2rel	weighted.mean.d2PdT2 weighted.mean.d2PdT2rel weighted.mean.d2.FF.PdT2 weighted.mean.d2.FF.PdT2rel weighted.mean.d2PLdT2 weighted.mean.d2PLdT2rel weighted.mean.d2.FF.PLdT2 weighted.mean.d2.FF.PLdT2rel weighted.mean.d2HCMdT2 weighted.mean.d2HCMdT2rel weighted.mean.d2.FF.HCMdT2 weighted.mean.d2.FF.HCMdT2rel
10	weighted.mean.HCT weighted.mean..FF.HCT mean.cumS variance.cumS mean.cumC variance.cumC chronOrder	weighted.mean.HCT weighted.mean..FF.HCT mean.cumS variance.cumS mean.cumC variance.cumC chronOrder
11	weighted.mean.dHCTdT	weighted.mean.dHCTdT
12	weighted.mean.dHCTdTrel weighted.mean.d.FF.HCTdTrel	weighted.mean.dHCTdTrel weighted.mean.d.FF.HCTdTrel
13	weighted.mean.d2HCTdT2	weighted.mean.d2HCTdT2
14	weighted.mean.d2HCTdT2rel weighted.mean.d2.FF.HCTdT2rel	weighted.mean.d2HCTdT2rel
15		weighted.mean.d2.FF.HCTdT2rel

16	weighted.mean.d.FF.HCTdT	weighted.mean.d.FF.HCTdT
17	weighted.mean.d2.FF.HCTdT2	weighted.mean.d2.FF.HCTdT2
18	weighted.mean.dHCMdTrel weighted.mean.d.FF.HCMdTrel	weighted.mean.dHCMdTrel weighted.mean.d.FF.HCMdTrel
19	mean.S mean.C	mean.S mean.C
20	variance.S	variance.S
21	mean.dSdT mean.dCdT	mean.dSdT mean.dCdT
22	variance.dSdT	variance.dSdT
23	variance.C	variance.C
24	variance.dCdT	variance.dCdT

Table 22: Feature correlation groups, left for $M_{min} = 1.5$ with $T_{start} = 1995$; right for $M_{min} = 1.2$ with $T_{start} = 1995$.

Nr.	$M_{min} = 1.2$ $T_{start} = 2004$ $T_{agg} = 3$ months	$M_{min} = 1.0$ $T_{start} = 2004$ $T_{agg} = 1$ months
1	sum.Q.Gas.M3	sum.Q.Gas.M3
2	variance.Q.Gas.M3	variance.Q.Gas.M3
3	sum.dQdT.Gas.M3	sum.dQdT.Gas.M3
4	variance.dQdT.Gas.M3	variance.dQdT.Gas.M3
5	sum.d2QdT2.Gas.M3	sum.d2QdT2.Gas.M3
6	variance.d2QdT2.Gas.M3	variance.d2QdT2.Gas.M3
7	weighted.mean.P weighted.mean..FF.P weighted.mean.PL weighted.mean..FF.PL weighted.mean.HCM weighted.mean..FF.HCM	weighted.mean.P weighted.mean..FF.P weighted.mean.PL weighted.mean..FF.PL weighted.mean.HCM weighted.mean..FF.HCM
8	weighted.mean.dPdT weighted.mean.dPdTrel weighted.mean.d.FF.PdT weighted.mean.d.FF.PdTrel weighted.mean.dPLdT weighted.mean.dPLdTrel weighted.mean.d.FF.PLdT weighted.mean.d.FF.PLdTrel weighted.mean.dHCMdT weighted.mean.d.FF.HCMdT diff.mean.C.S	weighted.mean.dPdT weighted.mean.dPdTrel weighted.mean.d.FF.PdT weighted.mean.d.FF.PdTrel weighted.mean.dPLdT weighted.mean.dPLdTrel weighted.mean.d.FF.PLdT weighted.mean.d.FF.PLdTrel weighted.mean.dHCMdT weighted.mean.d.FF.HCMdT diff.mean.C.S
9	weighted.mean.d2PdT2 weighted.mean.d2PdT2rel weighted.mean.d2.FF.PdT2 weighted.mean.d2.FF.PdT2rel weighted.mean.d2PLdT2 weighted.mean.d2PLdT2rel weighted.mean.d2.FF.PLdT2 weighted.mean.d2.FF.PLdT2rel weighted.mean.d2HCMdT2 weighted.mean.d2HCMdT2rel weighted.mean.d2.FF.HCMdT2 weighted.mean.d2.FF.HCMdT2rel	weighted.mean.d2PdT2 weighted.mean.d2PdT2rel weighted.mean.d2.FF.PdT2 weighted.mean.d2.FF.PdT2rel weighted.mean.d2PLdT2 weighted.mean.d2PLdT2rel weighted.mean.d2.FF.PLdT2 weighted.mean.d2.FF.PLdT2rel weighted.mean.d2HCMdT2 weighted.mean.d2HCMdT2rel weighted.mean.d2.FF.HCMdT2 weighted.mean.d2.FF.HCMdT2rel

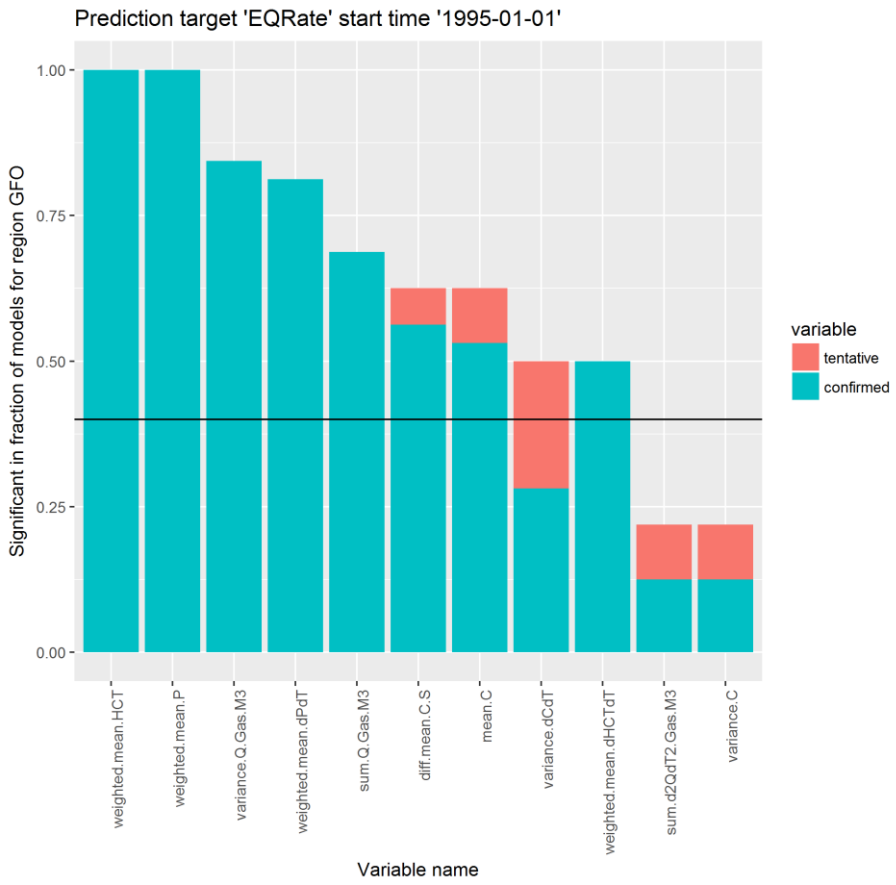
10	weighted.mean.HCT weighted.mean..FF.HCT mean.cumS variance.cumS mean.cumC variance.cumC chronOrder	weighted.mean.HCT weighted.mean..FF.HCT mean.cumS variance.cumS mean.cumC variance.cumC chronOrder
11	weighted.mean.dHCTdT	weighted.mean.dHCTdT
12	weighted.mean.dHCTdTrel weighted.mean.d.FF.HCTdTrel	weighted.mean.dHCTdTrel weighted.mean.d.FF.HCTdTrel
13	weighted.mean.d2HCTdT2	weighted.mean.d2HCTdT2
14	weighted.mean.d2HCTdT2rel	weighted.mean.d2HCTdT2rel
15	weighted.mean.d2.FF.HCTdT2rel	weighted.mean.d.FF.HCTdT
16	weighted.mean.d.FF.HCTdT	weighted.mean.d2.FF.HCTdT2
17	weighted.mean.d2.FF.HCTdT2	weighted.mean.d2.FF.HCTdT2rel
18	weighted.mean.dHCMdTrel weighted.mean.d.FF.HCMdTrel	weighted.mean.dHCMdTrel weighted.mean.d.FF.HCMdTrel
19	mean.S mean.C	mean.S mean.C
20	variance.S	variance.S
21	mean.dSdT mean.dCdT	mean.dSdT mean.dCdT
22	variance.dSdT	variance.dSdT
23	variance.C	variance.C
24	variance.dCdT	variance.dCdT

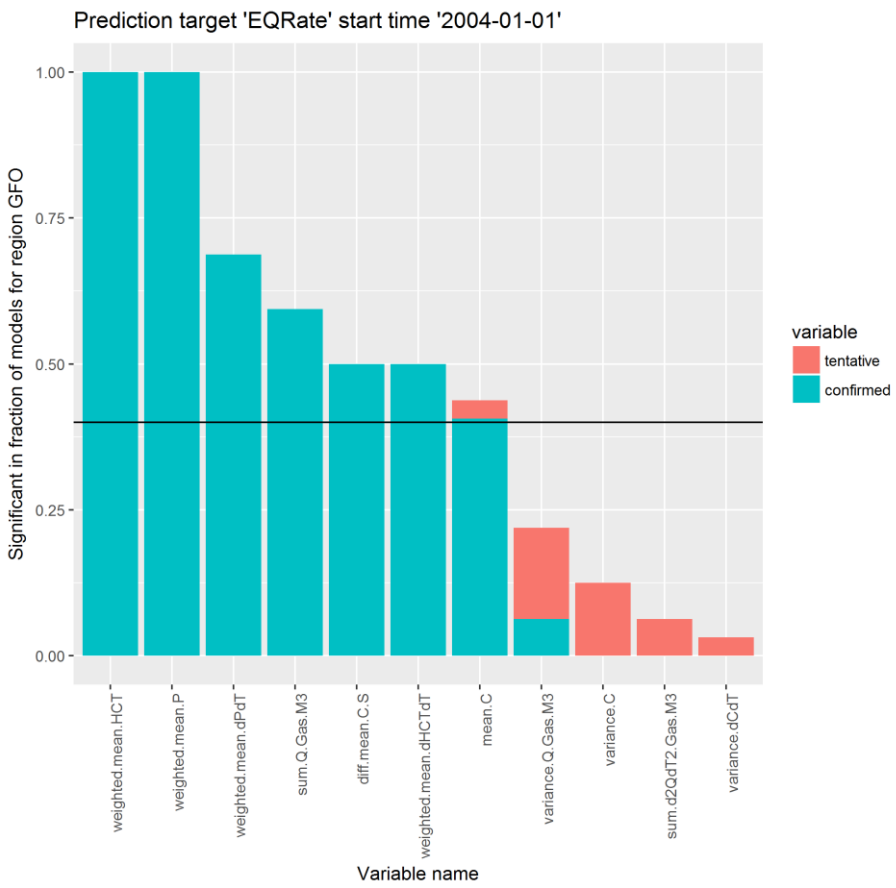
Table 23: Feature correlation groups, left for $M_{min} = 1.2$ with $T_{start} = 2004$; right for $M_{min} = 1.0$ with $T_{start} = 2004$.

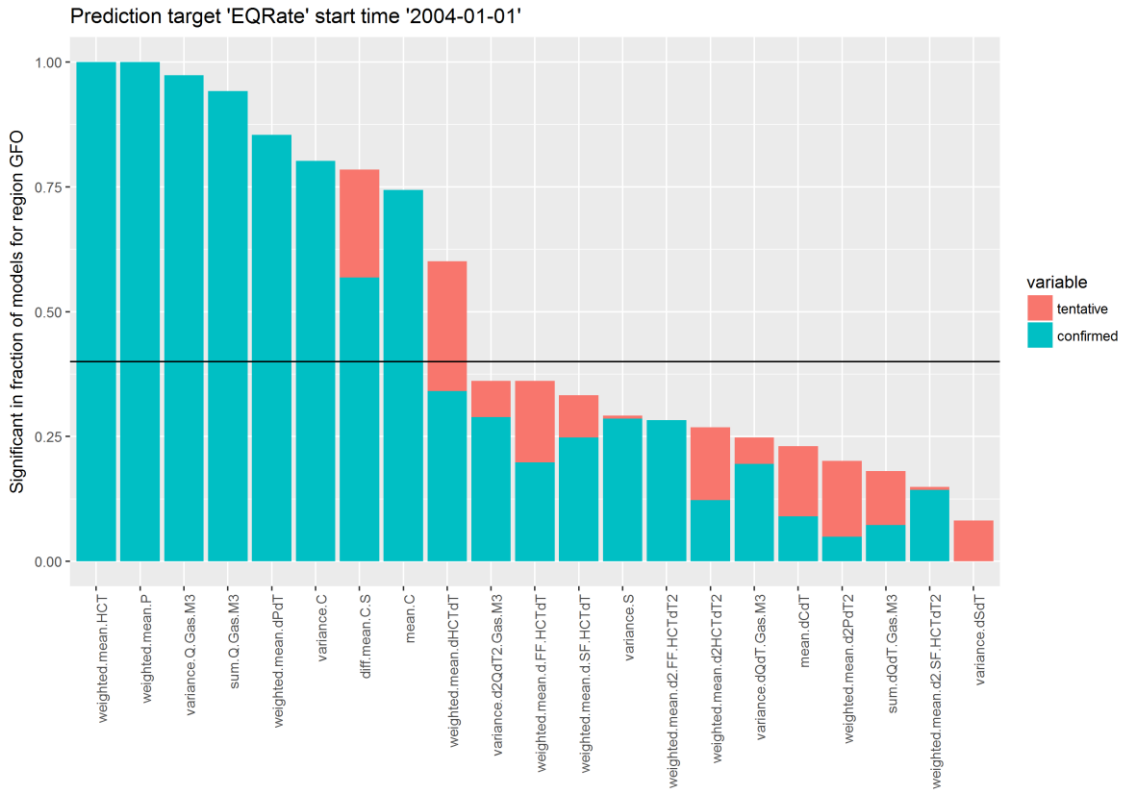
Appendix 3. Feature Significance Plots

Feature significance plots for targets:

- Top: RF-FC35-1.2 for target $M_{min} = 1.2$, $T_{start} = 1995$ and $T_{agg} = 3$ months
- Middle: RF-FC36-1.2 for target $M_{min} = 1.2$, $T_{start} = 2004$ and $T_{agg} = 3$ months
- Bottom: RF-FC107 for target $M_{min} = 1.0$, $T_{start} = 2004$ and $T_{agg} = 1$ months




Prediction target 'EQRate' start time '2004-01-01'
Significant in fraction of models for region GFO
1.00
0.75
0.50
0.25
0.00
weighted.mean.HCT
weighted.mean.P
weighted.mean.dPdT
sum.Q.Gas.M3
diff.mean.C.S
weighted.mean.dHCTdT
mean.C
variance.Q.Gas.M3
variance.C
sum.d2QdT2.Gas.M3
variance.dCdT
Variable name
variable
tentative
confirmed


Prediction target 'EQRate' start time '2004-01-01'
Significant in fraction of models for region GFO
1.00
0.75
0.50
0.25
0.00
weighted.mean.HCT
weighted.mean.P
variance.Q.Gas.M3
sum.Q.Gas.M3
weighted.mean.dPdT
variance.C
diff.mean.C.S
mean.C
weighted.mean.dHCTdT
variance.d2QdT2.Gas.M3
weighted.mean.d.FF.HCTdT
weighted.mean.d.SF.HCTdT
variance.S
weighted.mean.d2.FF.HCTdT2
weighted.mean.d2HCTdT2
variance.dQdT.Gas.M3
mean.dCdT
weighted.mean.d2PdT2
sum.dQdT.Gas.M3
weighted.mean.d2.SF.HCTdT2
variance.dSdT
variable
tentative
confirmed

Appendix 4. Machine Learning Model Details

Chapter 6 provides a high-level overview of the machine learning models used in this study and section 9.4 illustrates how their hyperparameters are tuned. For the models used in this study, this appendix provides an illustrative overview of the key hyperparameters, as well as some advantages and limitations. Please note that this overview does not intend to be exhaustive but merely illustrative. We refer the interested reader to the references mentioned in the introduction of chapter 6.

A4.1 Generalized Linear Models (Elastic net)

Hyperparameters

Here we present the 2 most important for a complete list refer to the algorithm section in Elements of Statistical Learning (Friedman, Tibshirani, & Hastie, 2009):

- $\boldsymbol{\alpha}$: also referred as the elastic net mixing parameter and it defines the weight given to the L1 and L2 regularizations respectively. A value of $\alpha = 0$ means that the solution will use only Ridge regression while a value of $\alpha = 1$ means it will use LASSO. Any other value $0 < \alpha < 1$ hence assigns a relative weight to each regularization strategy. One use of α is for numerical stability; for example, the elastic net with $\alpha = 1 - \epsilon$ for some small $\epsilon > 0$ performs much like the LASSO, but removes any degeneracies and wild behaviour caused by extreme correlations.
- $\boldsymbol{\lambda}$: this is the penalty parameter and specifies the strength of the penalty to be applied. From a Bayesian perspective, λ can be interpreted as the prior-uncertainty of the model parameters.

Advantages

- Easy to understand and interpret.
- Extrapolation straightforward.
- GLM Nets can deal with situations when the number of features is greater than the number of samples, and with correlated features.

Disadvantages

- As the name states, it is a parametric model that assumes linear relation between features and target.

A4.2 K-Nearest Neighbours

Hyperparameters

Belonging to the family of non-parametric algorithms, KNN only has a very limited set of levers for us to use when influencing the algorithm's outcome, here we present the 2 most important for a complete list refer to the algorithm section in Elements of Statistical Learning (Friedman, Tibshirani, & Hastie, 2009):

- $\boldsymbol{K}$: the number of examples, i.e., "nearest neighbours", which will be taken into account when producing a prediction for a new instance. Setting K too low can lead to overfitting, as a prediction can be based on as little as one training example, leading to an overly flexible decision boundary. Conversely, setting K too high will lead to a poor (under)fit. As an illustration, consider setting K equal to n, the number of observations in our training set.

The unweighted prediction for a new observation would then be based on all the training samples, equivalent to predicting a naïve group mean.

- **Distance metric**: to measure the distance between two observations x and y in a multidimensional feature space, a variety of distance metrics can be used, both for continuous features and categorical features. Some distance metrics are the Euclidean distance, the Minkowsky distance, the Manhattan distance and the Chi-Square distance. See (Hu, Huang, Ke, & Tsai, 2016) for a comparative review.

Advantages

- The algorithm is intuitive to understand.
- Non-parametric.
- KNN can use weighting to influence the impact of certain features that are deemed more important; this also allows for the integration of expert knowledge into the model building process.

Disadvantages

- As the dimensionality of the problem increases, KNN faces a very large search problem and might start to deteriorate in performance.
- The range of predicted outcomes coming from a KNN will be constrained by the range of the target value of the training set. No native method for extrapolation.
- KNN can have poor performance if the number of training samples is small.

A4.3 Random Forests

Hyperparameters

The following is a list of the key hyperparameters. More hyperparameters exist and the interested reader is encouraged to check the available documentation (Bischl, et al., 2016) .

- **Number of trees**: increasing the number of trees can make for a more stable bootstrap aggregate, but this comes at the expense of computation time.
- **Percentage of features randomly selected**: at each split in the decision trees, not all features in our training set are available to split on. Setting a sufficiently low value for this hyperparameter is key in decorrelating the individual trees and growing an effective Random Forest, though setting the value too low might prevent the algorithm from capturing interaction effects.
- **Minimum leaf size**: the minimum number of observations that must be present in the child nodes for an individual tree to make that split. This generally reduces overfitting when applied to a single decision tree. However, in the Random Forest paradigm, it can be beneficial to grow very deep individual trees, since the overfitting effects of single trees can get smoothed out in the ensembling process.

Advantages

- Non- Parametric.
- Robust vs. overfitting: the bootstrap aggregation paradigm makes Random Forests robust to overfitting, as overfitting on single trees gets "smoothed out" when considering the entire ensemble.
- Implicit feature selection: because decision trees must select one feature that returns the best split at each step of the tree growing process, they perform implicit feature selection.

Disadvantages

- Unable to output predictions that exceed the range of the dependent variable in the training set or equivalently, not able to extrapolate.
- Because a prediction generated for a new observation is the amalgamation of the predictions of hundreds or even thousands of trees, it can be very difficult to figure out which decision rules applied to a particular instance.
- When using Random Forests on time series data, the out-of-bag error rates should not be used, as these are based on traditional cross-validation, rather than walk-forward testing. In this research, we therefore always focus on walk-forward error estimates.

A4.4 SVR

Hyperparameters

The following is a list of the key hyperparameters. More hyperparameters exist and the interested reader is encouraged to check the available documentation (Bischl, et al., 2016)

- **Kernel**: Several kernels are available that allow the SVM to have nonlinear decision boundaries in the original space by performing linear separation in kernel space.
- **Gamma**: the size of the sphere of influence of individual points when training the model.
- C: The cost parameter of misclassifying an instance. A high C therefore implies that the model is incentivized to try to fit all the observations in the training set more closely, which can result in a more jagged decision function. A lower C-value results in a smoother decision function, potentially preventing overfitting.

Advantages

- Non- parametric.
- Able to make extrapolations outside the range of the dependent variable in the training set.
- SVR's are still effective when the number of features k is larger than the number of training samples n.

Disadvantages

- When the number of features gets much greater than n, SVM's face overfitting challenges.
- SVM's provide no probabilistic estimates, be it of class membership or point estimates.
- The support vector set is not mathematically guaranteed to be sparse, so in practice it is possible to have a large part of the dataset designated as support vectors, negating some of the efficiency benefits mentioned earlier.

A4.5 ARIMA

Hyperparameters

ARIMA models are defined in terms of the three components we have described in the previous sections. These are referred to as the P, D, and Q parameters:

- P: order of the autoregressive term in the model.
- D: order of differencing (/integration) applied to the target series.
- Q: order of the moving average term in the model.

It is common to see these parameters stated in a fixed order as ARIMA(p, d, q). For example, an ARIMA$(2, 1, 0)$ model refers to a model with an autoregressive component of order 2, and to which differencing has been applied once.

Advantages

- Native ability to deal with time-dependent structure in the data, and use this information in the forecast.
- Can extrapolate outside of the range of the dependent variable in the training set.
- Extensible for (among others) seasonal effects and external regressors.

Disadvantages

- Parametric
- ARIMA models are essentially a linear combination of their AR and MA parameters, which can be considered a restriction when trying to model multiple non-linear dependencies.
- A major disadvantage is that they only use the information in the prediction target. They don't make use of other features which is a major drawback since not all information is in the time series itself.

A4.6 Neural Networks

Hyperparameters

Beyond "traditional" hyperparameter tuning (listed below), trying to fit an appropriate neural net also means determining the right architecture: how many hidden layers will the neural net contain, what is the size of each of these layers, and which activation function will the nodes use? For smaller problems, this can be tuned using, e.g., grid search, like we do with traditional hyperparameters. For larger problems where training a neural network can take hours, it is wise to base the model on established industry architectures.

Beyond the size and shape of the neural net, some of the most important hyperparameters are:

- **Learning rate**: The learning rates determines the size of the weight updates during backpropagation. As with other algorithms we discussed, a model with a low learning rate will take a long time to converge. On the other hand, a model with a high learning rate can cause divergence. Generally, this parameter is adaptive and decreased over time.
- **Activation functions**: the output of a node within a neural network is not simply the sum of its connected input weights. If this were the case, the entire neural net could be collapsed to one single matrix operation, which would be unable to approximate non-linear functions.
- **Number of training iterations**: it is needed to tune the number of training iterations to prevent overfitting. Most commonly, this is done via early stopping, a technique that adaptively stops training once performance on a out of sample set of data stops increasing.

Advantages

- Non-parametric.
- Able to extrapolate to values outside the range of the dependent variable in the training set.
- NN's are generally good at dealing with data with high heteroskedasticity i.e. data with high volatility and non-constant variance, thanks to the fact that it can learn hidden relationships without imposing fixed relationships in the data.

Disadvantages

- Often can be a hard to interpret "black-box" since it may be very difficult to elucidate the relation between inputs and outputs.
- Require mindful and methodical tuning of hyper parameters to improve performance, which is why often "out-of-the-box" NN's do not perform as expected due to random initialization of parameters.
- Generally, require many observations and training examples. NN's are a good example of when more data does often equate to better results.

A4.7 Gradient Boosting Machines (GBM)

Hyperparameters

GBMs have a rather large number of tuneable parameters, resulting in increased tuning difficulty over, e.g., Random Forests. A subset of these parameters that we consider particularly important:

- **Learning rate**: the learning rate determines the impact of the individual learners on the overall outcome. Smaller learning rates are generally preferred, as they have a higher generalization capacity. This does come at a computational expense, as smaller learning rates also requires more weak learners (e.g. trees) to be fitted in the ensemble.
- **Number of estimators**: number of weak learners in the ensemble.
- **Subsample**: the fraction of observations selected for each tree. As with Random Forests, setting this parameter < 1 can help to reduce variance of the overall model.

In addition to these meta-parameters, it is of course also possible to change parameter values that pertain to the individual weak learners.

Advantages

- Non- Parametric.
- In terms of base predictive performance and width of applicability, GBMs are among the strongest models around.
- Since in general, GBMs can achieve their performance using fewer base trees than Random Forests, the resulting models are lightweight in comparison.

Disadvantages

- Whether extrapolation is possible depends on the weak learners chosen.
- GBMs are more prone to overfitting than a lot of other methods.
- GBMs face a longer training time because of the sequential nature of the boosting paradigm (training on residuals of prior weak learners).

Appendix 5. Guards against spurious false positives

Given the large amount of experiments and the complex workflow underlying it, we carried out a series of tests with randomized data to guard against spurious detections and provide evidence for the correctness of the implementation. Tests with random data examining complex workflows containing statistical procedures have been popularized by (Bennett, Baird, Miller, & Wolford, 2011). The first section describes tests with random permutation of input data and the second section tests with random permutation of the target. The tests in these sections show very limited performance for the randomly permuted data, as expected and in stark contrast with original non-permuted data. While these, of course, cannot be taken as definite proof, they do show that there are no obvious issues in the aspects that are covered by the tests.

A5.1 Random Permutation of Input Data

Since we derive many features from the raw input data (see section 3) some of them may, by chance, be related to the prediction target. Especially if also shifts are applied to the data. For this reason, we want to estimate the false positive rate of our workflow to detect features as statistically significant in the context of the (other) given features with respect to predicting the chosen prediction target. Hence, we have randomly and individually permuted a selected subset of the raw input features and tested via our workflow how often some feature derived from the randomly permuted data would be tested as predictive. Since these tests incur a significant amount of computational cost we limit ourselves to compaction, subsidence and production data.

The test procedure works as follows:

1. Randomly shuffle the input data under investigation: destroying all temporal structure in the data but preserving the range and distribution of points.
2. Run a random subset of the factorial setup as described in Subsection 8.1.
3. Test in which fraction of the models any of the features derived from the randomly shuffled data is deemed significant by the Boruta test with a significance threshold of 0.05.
4. Repeat steps 1-3 in total 5 times, always with the same subset of models used in step 2.
5. Calculate the standard deviation for the fraction of features tested as significant in each of the 5 iterations.

Note that since we are deriving several features from each individual raw data set, it is expected that the false discovery is higher than 0.05 and essentially proportionally related to the number of derived features (around 5). The results for the features that we have randomly permuted is shown in Table 24. We observe that there is a clear separation between original and randomly permuted data. That illustrates that our workflow is robust to data which is just by chance correlated with our prediction target. Additionally, this also provides evidence that the original data in form of production, subsidence and compaction is linked to earthquake rate and can thus be used to create a predictive model.

Feature name	Fraction of runs in which a feature derived from the original data was tested as significant	Fraction of runs in which a feature derived from the permuted data was tested as significant
Production	86%	$23 \pm 16\%$
Subsidence	82%	$27 \pm 15\%$
Compaction	87%	$23 \pm 16\%$

Table 24: Overview of feature significance test results with randomly permuted production, subsidence and compaction data when predicting earthquake rate

A5.2 Random Permutation of Prediction Target

In addition to the test outlined in Table 24, we have also performed a series of tests to guard against workflow bugs related to data integration and pre-processing that could lead to information leakage. To spot those, we randomly permute our prediction target several times, to then run a (random) subset of the factorial setup and test in how many cases we beat the best naïve baseline. Note that the best baseline is previously determined to be the moving average with automatically tuned step length. In case of a significant amount of the tests indicate that we beat the baseline on a randomly permuted prediction target, information leakage is most likely happening. The test procedure works as follows:

1. Randomly permute the prediction target, the earthquake rate.
2. Run a random subset of the factorial setup as described in Subsection 8.1.
3. Test if any of the models in the factorial setup managed to beat the baseline in the RMSLE and/or MAE metric in any of the regions.
4. Repeat steps 1-3 in total 5 times with always the same subset of models in step 2.

The outcome of the experiment is captured in Table 25, showing that in none of the factorial setups we beat the baseline if randomly permuted earthquake rates were used as the prediction target.

	Original data	**1**	**2**	**3**	**4**	**5**
Best model(s) beat baseline in either MAE or RMSLE metric in any of the regions.	Yes	No	No	No	No	No

Table 25: Our workflow only manages to beat the simple baseline when the original earthquake rate is used as prediction target.

Appendix 6. Overview of Model Performance for all Error Metrics

As pointed out in Subsection 5.2., we have focused our analysis on the MAE and RMSLE error metrics. While the results in terms of relative ranking across different metrics are mostly consistent a more detailed analysis should follow at a later stage. For the sake of completeness we present the complete table of error metrics for one prediction with experiment $M_{min} = 1.5$, $T_{start} = 1995$ and T_{agg} = 3 months below in Table 27.

$M_{min} = 1.5$ $T_{start} = '95$ $T_{agg} = 3$ m	Model: Random Forest	Model: SVM	Model: KNN	Model: GLM Top	Baseline: Moving Average	Baseline: Depletion Moving Average
MAE	0.018±0.002	0.019±0.002	0.019±0.002	0.020±0.002	0.019±0.002	0.020±0.002
RMSLE	0.025±0.003	0.025±0.003	0.026±0.003	0.026±0.003	0.025±0.002	0.026±0.002
R^2	0.209±0.183	0.180±0.187	0.158±0.192	0.117±0.212	0.214±0.158	0.159±0.153
RMSE	0.026±0.003	0.027±0.003	0.027±0.003	0.027±0.003	0.026±0.003	0.029±0.003
MPL	0.131±0.009	0.132±0.009	0.131±0.009	0.132±0.009	0.131±0.009	0.131±0.009

Table 26: Error metrics of three models for meta parameter setting FC01-1.5 for the target $M_{min} = 1.5$, $T_{start} = '95$ and $T_{agg} = 3$ months. Error metrics for the best statistical and best simple physical baseline in the MAE metric are also shown.

Appendix 7. Quantitative Evaluation of $T_{start} = 2004$ targets

The error measures for the targets with $M_{min} = 1.2$ (Table 27) and $M_{min} = 1.0$ (Table 28) with $T_{start} = 2004$ are shown below. R^2 is the only error metric that can be compared cross targets – we observe that R^2 for $M_{min} = 1.0$ is larger than for $M_{min} = 1.2$, which is expected in light of the increase in number of events between both targets. Compared with the targets starting at 1995 R^2 is quite a bit lower, hence possibly earlier times might be easier to forecast or help to train the models but further investigation is needed to reach more definite conclusions on this point.

For $M_{min} = 1.2$ the Random Forest, KNN and SVM outperform the best baseline in all metrics, but neither paired nor unpaired these differences are statistically significant in the current setup: for the Random Forest we get $p = 0.264$ with test statistic $V = 440$ for the paired Wilcoxon test and $p = 0.515$ and test statistic $W = 972$ for the unpaired Wilcoxon test. Results for the KNN and SVM are similar. For $M_{min} = 1.0$ the MAE for the Random Forest and the best baseline are similar and all other machine learning models perform worse – hence without testing we conclude none is better.

$M_{min} = 1.2$ $T_{start} = '04$ $T_{agg} = 3\,\mathrm{m}$	Model: Random Forest	Model: KNN	Model: GLM Net	Model: SVM	Baseline: Auto Moving Average	Baseline: Depletion Moving Average
MAE	0.034±0.004	0.035±0.004	0.036±0.004	0.035±0.004	0.036±0.005	0.036±0.005
RMSLE	0.038±0.004	0.039±0.004	0.041±0.005	0.041±0.005	0.043±0.006	0.044±0.005
R^2	0.131±0.187	0.063±0.197	-0.025±0.238	0.050±0.229	-0.129±0.281	-0.173±0.246

Table 27: Error metrics of the best three models for meta parameter setting FC36-1.2 for the target $M_{min} = 1.2$, $T_{start} = '04$ and $T_{agg} = 3$ months. Error metrics for the best statistical and best physical baseline are also shown. In case rankings differed for various metrics the MAE has been used as guiding metric.

$M_{min} = 1.0$ $T_{start} = '04$ $T_{agg} = 1\,\mathrm{m}$	Model: Random Forest	Model: SVM	Model: GLM Top	Model: GLM Net	Baseline: Moving Average	Baseline: Depletion Start
MAE	0.056±0.004	0.059±0.004	0.059±0.004	0.057±0.004	0.058±0.004	0.056±0.004
RMSLE	0.064±0.005	0.067±0.005	0.067±0.005	0.067±0.005	0.067±0.006	0.066±0.005
R^2	0.214±0.138	0.142±0.138	0.152±0.144	0.138±0.160	0.147±0.170	0.161±0.154

Table 28: Error metrics of the best three models for meta parameter setting FC107 for the target $M_{min} = 1.0$, $T_{start} = '04$ and $T_{agg} = 1$ month. Error metrics for the best statistical and best physical baseline are also shown. In case rankings differed for various metrics the MAE has been used as guiding metric.

Appendix 8. Random Forest Seismicity Drivers

This appendix shows which features drive Random Forest event rate forecasts using Variable Importance plots as explained in section 7.1. How these driving features on average influence seismicity can be explored with Individual Conditional Expectation (ICE) plots, discussed in section 7.3. The variable importance plots of the selected MMPs for $M_{min} = 1.5$ and $M_{min} = 1.2$ with $T_{start} = 1995$ are shown in Figure 62 and Figure 64 respectively. For both MMPs the weighted mean HCT and weighted mean P are the most important drivers, followed at a more modest place by the first temporal difference $\Delta HCT/\Delta t$. For the MMP of $M_{min} = 1.2$ the geospatial variance of gas production $var(Q)$ and the first temporal difference $\Delta P/\Delta t$ also have a modestly driving effect.

The ICE plots for the top-three drivers HTC, P and $\Delta HCT/\Delta t$ are shown for both MMPs in Figure 63 and Figure 65. We see that in general a decreasing pressure P or an increasing Hydrocarbon Column Thickness HCT increases seismicity. The effect of $\Delta HCT/\Delta t$ is less pronounced: from around $\frac{\Delta HCT}{\Delta t} \sim 4 \cdot 10^{-5}\ \frac{\text{m}}{\text{month}}$ both an increase and decrease seems to result in a small seismicity increase, but uncertainties are large compared to the size of the effect.

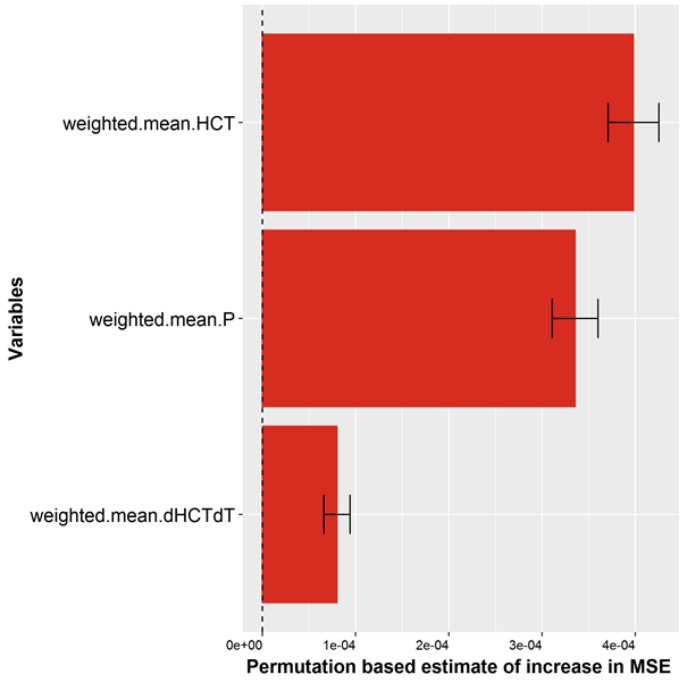


Figure 62: variable importance plot of FC-01-1.5 for GFO.

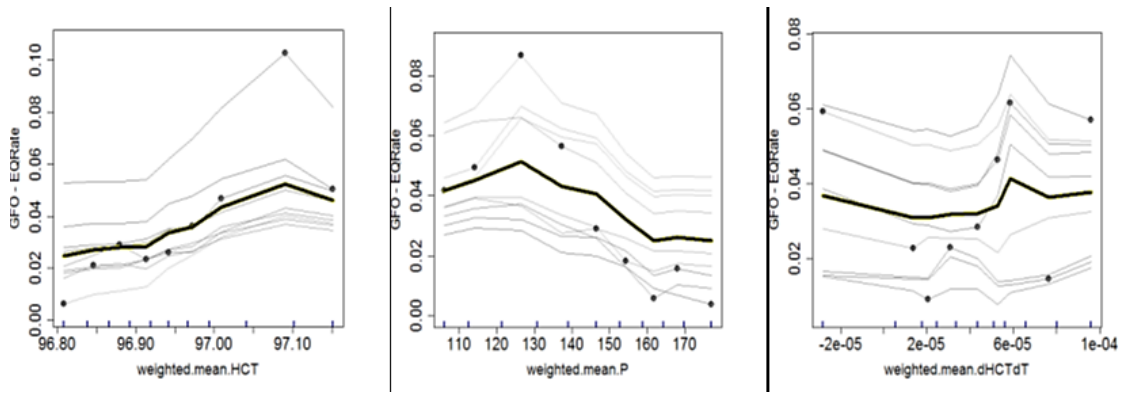


Figure 63: ICE plots for the seismicity drivers of FC-01-1.5. From left to right in decreasing order of importance: weighted mean HCT, weighted mean P, weighted mean $\Delta HCT/\Delta t$.

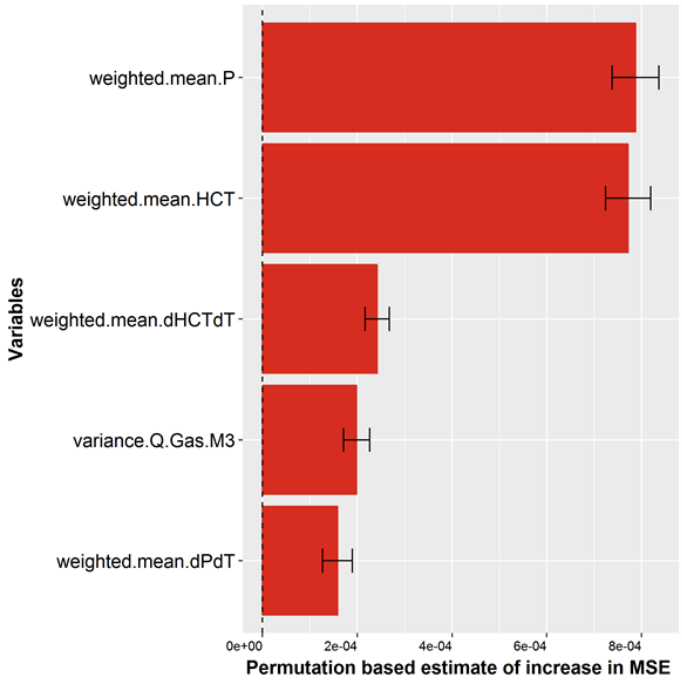


Figure 64: variable importance plot of FC-35-1.2 for GFO.

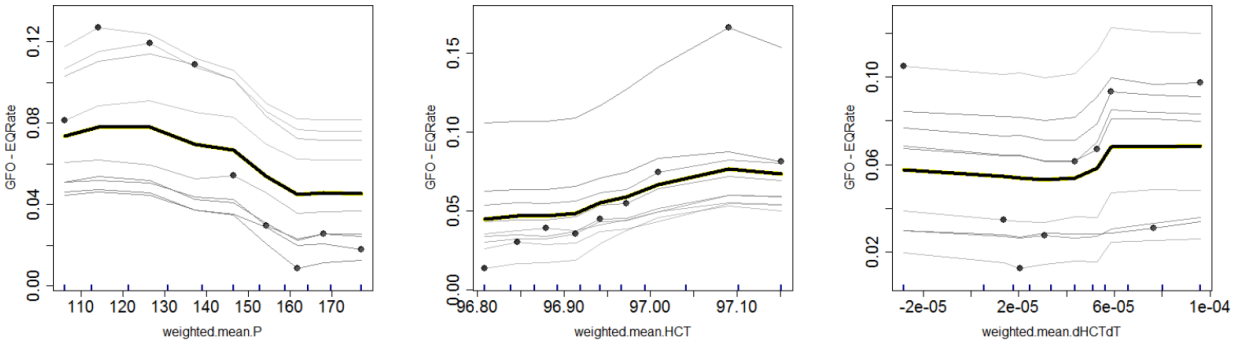


Figure 65: ICE plots for the top three seismicity drivers of FC-35-1.2 for GFO. From left to right in decreasing order of importance: weighted mean $\boldsymbol{P}$, weighted mean $\boldsymbol{HCT}$, weighted mean $\boldsymbol{\Delta HCT/\Delta t}$.

Appendix 9. Definitions, Mathematical Concepts, Abbreviations

Physical quantities are denoted by the following variables:

- Q Gas produced (m^3);
- P Pressure in reservoir (bar);
- S Subsidence (m);
- C Compaction (m);
- HCT Hydrocarbon Column Thickness (m);
- HCM Hydrocarbon Column Mass (kg)

Machine learning concepts used:

- **Target**: the (preprocessed) variable we like to predict. For example, the target used throughout this study is the number of earthquakes of $M \geq 1.0$ per region per month. Alternative targets we could have used are e.g. the number of earthquakes of $M \geq 1.5$ per 10 km ^{2} per year or the number of earthquakes of $M \geq 0.5$ per fault property per mm subsidence.
- **Feature**: a (preprocessed) source data variable which might be a predictor for the target. Note that one source data variable might give rise to multiple features or the other way around. For example, in this study features are: the mean pressure in a region per month, the mean pressure decrease in a region per month, the total production in a region per month, the geospatial variance of the production in a region per month, etc. Confusingly, sometimes features are also called covariates or plainly variables.
- **Covariates**: see features.
- **Machine learning model**: a formula or association rule associating feature values to target values. Usually, machine learning models are complex and do not provide intuitive insights. Various model types are discussed in chapter 6.

Mathematical notations and definitions used, unless otherwise specified:

- $\mathbb{N}$ denotes the set of natural numbers;
- $\mathbb{R}$ denotes the set of real numbers;
- $[a, b]$ is the interval between a and b, including boundary values;
- (a, b) is the interval between a and b, excluding boundary values;
- m is the number of features/variables/covariates in a model;
- n is the number of data points that are available for each features/variables/covariate
- $x_i \in \mathbb{R}^m$ is the vector of features/variables/covariates of a model at time interval i;
- t_i is the target value at time interval i;
- $d_i = (x_i, t_i)$ denotes a data point d at time interval i, consisting of the m features/variables/covariates and the target;
- f denotes a model or association rule
- $p_i = f(x_i)$ is the prediction of model f based on features $x_i \in \mathbb{R}^m$ on time i

Geospatial coordinate systems mentioned:

- **RD**: the Netherlands triangulation system [Rijksdriehoekstelsel] is a coordinate system used at the national level in the European part of the Netherlands. It has two perpendicular coordinates x and y. For details and transformations to other coordinate systems see (Kadaster, 2018). The transformations to and from the Latitude/Longitude and the RD coordinate systems have been done using the "proj4" and "sp" R packages which have special functionalities for the transformation and application of different cartographic projections. For more details see the package descriptions (Urbanek, 2015).

Key abbreviations used throughout the text:

- GBM Gradient Boosting Machine, a machine learning model (see section 6.8);
- GFO Groningen Field Outline (see section 3.2);
- GLM Generalized Linear Model, a machine learning model (see section 6.2);
- ICE Individual Conditional Expectations (see section 7.3);
- I.i.d. Independently and Identically Distributed (see section 5.4);
- KNN K-Nearest Neighbours, a machine learning model (see section 6.3);
- MAE Mean Absolute Error, an error metric (see section 5.2);
- M_c Magnitude of Completeness (see section 3.2);
- M_{min} Minimum Magnitude (see section 3.2);
- MMP Model and Meta Parameter combination (see chapter 9);
- MoReS Dynamic Reservoir Model (see section 3.4);
- PSHRA Probabilistic Seismic Hazard and Risk Assessment (see chapter 2);
- R Statistical Computing Environment (see Appendix 10);
- RF Random Forest, a machine learning model (see section 6.4);
- RMSLE Root Mean Square Absolute Error (see section 5.2);
- SE Standard Error (see section 5.3);
- SVM Support Vector Model, a machine learning model (see section 6.5);
- SVR Support Vector Regression, an SVM used for regression.

Appendix 10. Tools

This study was executed in R (R Core Team, 2017). Three key R packages were used: (i) MLR for the general forecasting setup; (ii) Boruta to identify significant features and (iii) the I-Race package for tuning. These three packages are discussed in some more detail below.

A10.1. The MLR Package

In order to facilitate the setup of the forecasting experiments, we make use of the R package MLR which is described in more detail in (Bischl, et al., 2016). By using the MLR package as our general framework for the benchmarking experiments we avoid duplication of code and potential bugs in critical parts of the code related to estimating model performance. The MLR package provides a modular interface to the following common tasks in the context of machine learning workflows and benchmarking experiments:

- Common pre-processing routines as removal of constant or duplicated columns, and normalization of the covariate columns. MLR implements common scaling techniques like standardization but also provides access to more advanced pre-processing routines like those exposed through a wrapper for the pre-processing routines offered by the Caret package (Kuhn M. , 2017). Those include Box-Cox transformations, Yeo-Johnson transformations and also a set of different imputation techniques.
- Feature selection, based on different criteria (e.g. correlation, RF variable importance, …).
- Definition of a sub-sampling strategy for the out-of-sample prediction experiments. For instance cross-validation and different versions of the bootstrap. Note that the walk forward resampling strategy is not implemented in MLR ≤ 2.11, which is why we had to implement it manually.
- Definition of several common error measures like MAE, RMSE, Kendall-Tau and R^2.
- A convenient interface to around 80 machine learning algorithms for both regression and classification. Those include (regularized) linear models, Support Vector Regression, tree based methods and different flavours of neural networks.
- Support for automated parameter tuning of those algorithms using different tuning strategies like basic grid searches bot also more advanced gradient based techniques.
- Setup and results reporting of benchmarking experiments for several techniques.

A10.2. The Boruta package

For feature selection and for the purpose of detecting "significant" features with respect to the selected prediction target, we make use of the Boruta package (Kursa, 2010). This record of significant and potentially significant variables is stored for every prediction experiment. It is a heuristic procedure that uses the variable importance measure calculated by implementations of the random forest algorithm. As such, non-linear effects and interactions between parameters are taken into account. In order to counter effects related to the multiplicity of noise variables the algorithm is iterative in nature. A sketch of how the algorithm works and how we use it, is contained in chapter 7.

A10.3. The I-Race package

The i-race package implements the i-race algorithm which was introduced in (M. López-Ibáñez, 2016). It was conceived to automatically tune the parameters of any algorithm to optimize a given objective function that is related to the parameters of the algorithm in an unknown way. The algorithm supports the tuning of both categorical and continuous parameters.

13 Acknowledgements

The authors are indebted to Taco den Bezemer (NAM) and Jan van Elk (NAM) for their strong and continued support for this study – without their enthusiasm and trust this study would not be here.

We are indebted to the lead reviewers for insightful discussions throughout this study, much of which has been incorporated in this study. In alphabetic order:

- Dr Stijn Bierman (Shell P&T);
- Dr Stephen Bourne (Shell P&T);
- Dr Franz Király (University College London);

This study has been reviewed within a larger audience at successive stages of its maturity. We owe a big "Thank You" to the extended review team for their constructive feedback. In alphabetic order:

- Dr Peter van den Bogert (Shell P&T);
- Dr Xander Campman (Shell P&T);
- Dr Pandu Devarakota (Shell P&T);
- Dr Munish Goyal (IBM Services)
- Dr Kees Hindriks (Shell P&T)
- MSc Stephen Lord (IBM Services);
- Dr Roger Yuan (Shell P&T);
- Dr Rick Wentinck (Shell P&T);
- Dr Mo Zhang (IBM Services).

As can be read in chapter 3, this study builds on the data provided by specialists. We would like to thank (alphabetically):

- Hermann Baehr (NAM), for providing subsidence data;
- Stijn Bierman (Shell P&T), for sharing his subsidence and compaction interpolations;
- Leendert Geurtsen (NAM), Per Valvatne (NAM) and Assaf Mar-Or (NAM) for providing both the dynamic reservoir data from MoReS and the production forecasts;
- Gerard Joosten (Shell P&T) for his support in exporting the HCT, HCM and related properties from MoReS;
- Richard Vietje (NAM), for providing historical production data;
- Clements Visser (NAM), for providing the fault data and estimates for the Groningen regions;
- Onno van der Wal (NAM), for providing compaction and subsidence data;
- Alan Wood (Shell P&T), for sharing a recent version of the Petrel model.

Finally, we would like to thank Robin Bakker (Shell SIEP), Harry van der Burg (Shell SITI), Maarten Veldhuizen (NAM), Mando Rotman (IBM Services), Jonito Douwes Dekker (IBM Services) and Phaedra Kortekaas (IBM Services) for their organizational support in realizing this study.

14 Bibliography

Anderson, D. W., Kish, L., & Cornell, R. G. (1980). On Stratification, Grouping and Matching. *Scandinavian Journal of Statistics*, 61–66.

Asencio Cortez, G., Martinez-Alvarez, F., Morales-Esteban, A., & Reyes, J. (2016). A sensitivity study of seismicity indicators in supervised learning to improve earthquake prediction. *Knowledge-based systems*, 15-30.

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. *URL http://archive. ics. uci. edu/ml*.

Bence, J. R. (1995). Analysis of Short Time Series: Correcting for Autocorrelation. *Ecology*, 628-639.

Bengio, C. N. (2003). Inference for the Generalization Error. *Machine Learning*, 239-281.

Bennett, C. M., Baird, A. A., Miller, M. B., & Wolford, G. L. (2011). Neural Correlates of Interspecies Perspective Taking in the Post-Mortem Atlantic Salmon: An Argument For Proper Multiple Comparisons Correction. *Journal of Serendipitous and Unexpected Results, Vol. 1 (2011), pp. 1-5 Key: citeulike:9170380, 1*, 1-5.

Bierman, S. (2017). *Seasonal variation in rates of earthquake occurrences in the Groningen field.* Shell Global Solutions International.

Bierman, S. M., Kraaijeveld, F., & Bourne, S. J. (2015). *Regularised direct inversion to compaction in the Groningen reservoir using measurements from optical leveling campaigns.* Shell Global Solutions International.

Bierman, S., Paleja, R., & Jones, M. (2015). *Statistical methodology for investigating seasonal variation in rates of earthquake occurrence in the Groningen field.* Shell Global Solutions International.

Bierman, S., Paleja, R., & Jones, M. (2016). *Measuring seasonal variation in rates of earthquake occurrence in the Groningen field - Improved methodology following independent external review.* Shell Global Solutions International.

Bierman, S., Randell, D., & Jones, M. (2017). *Bayesian methods for reservoir compaction estimation, applied to the Groningen gas field.* Shell Global Solutions International.

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., . . . Jones, Z. (2016). mlr: Machine Learning in R. *Journal of Machine Learning Research*, 1-5.

Bishop, C. (2007). *Pattern Recognition and Machine Learning* (Vol. 1st ed. 2006. Corr. 2nd printing 2011). Springer.

Bourne, S. J., & Oates, S. J. (2015). *An activity rate model of induced seismicity within the Groningen Field (part 1).* NAM.

Bourne, S. J., & Oates, S. J. (2015). *An activity rate model of induced seismicity within the Groningen Field (Part 2).* NAM.

Bourne, S. J., & Oates, S. J. (2017). Development of statistical geomechanical models for forecasting seismicity induced by gas production from the Groningen field. *Netherlands Journal of Geosciences*, s175–s182. doi:10.1017/njg.2017.35

Bourne, S. J., & Oates, S. J. (2017). Extreme Threshold Failures Within a Heterogeneous Elastic Thin Sheet and the Spatial-Temporal Development of Induced Seismicity Within the Groningen Gas Field. *Journal of Geophysical Research: Solid Earth, 122*, 10299-10320.

Bourne, S. J., Oates, S. J., Van Elk, J., & Doornhof, D. (2014). A seismological model for earthquakes induced by fluid extraction from a subsurface reservoir. *Journal of Geophysical Research: Solid Earth*, 8991-9015. doi:10.1002/2014JB011663

Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32.

Breiman, L. (2001). Statistical Modeling: The Two Cultures. *Statistical Science, 16*(3), 199-231.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Buijze, L., Van den Bogert, P. A., Wassing, B. B., Orlic, B., & Ten Veen, J. (2017). Fault reactivation mechanisms and dynamic rupture modelling of depletion-induced seismic events in a Rotliegend gas reservoir. *Netherlands Journal of Geosciences*, 131-148.

Cao, A., & Gao, S. S. (2002). Temporal variation of seismic b-values beneath northeastern Japan island arc. *Geophysical Research Letters*, 1334. doi:10.1029/2001GL013775

Carrasquilla, J., & Melko, R. G. (2017). Machine learning phases of matter. *Nature Physics*, 431-434. doi:10.1038/NPHYS4035

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 273-297.

Czando, C., Gneiting, T., & Held, L. (2009). Predictive Model Assesment for Count Data. *Biometrics*, 1254-1261.

Dalgaard, P. (2008). *Introductory Statistics with R.* New York: Springer Science & Business Media.

Delgado M.F, C. E. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research* , 3133-3181.

Dempsey, D., & Suckale, J. (2017). Physics-based forecasting of induced seismicity at Groningen gas field, the Netherlands. *Geophysical Research Letters*, 1-10. doi:10.1002/2017GL073878

Dost, B., & Haak, H. (2002). *A comprehensive description of the KNMI seismological instrumentation.* De Bilt: KNMI.

Dost, B., Goutbeek, F., Van Eck, T., & Kraaijpoel, D. (2012). *Monitoring induced seismicity in the North of the Netherlands: status report 2010.* De Bilt: KNMI.

Dost, B., Ruigrok, E., & Spetzler, J. (2017). Development of seismicity and probabilistic hazard assessment for the Groningen gas field. *Netherlands Journal of Geosciences*, s235-s245. doi:10.1017/njg.2017.20

Efron, B., & Stein, C. (May 1981). The Jackknife Estimate of Variance. *The Annals of Statistics*, 586–596.

Fagerland, M., & Sandvik, L. (2009). Performance of five two-sample location tests for skewed distributions with unequal variances. *Contemporary Clinical Trials*, 490–496.

Friedman, J., Hastie, T., & Tibshiranie, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 1-22.

Friedman, J., Tibshirani, R., & Hastie, T. (2009). *The Elements of Statistical Learning - Second Edition.* New York: Springer.

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2014). Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *ArxiV*.

Harris, C., & Bourne, S. (2015, Nov). *Maximum Likelihood Estimates of b-Value for Induced Seismicity in the Groningen Gas Field.* Shell Global Solutions International.

Hastie, T. (2007). Comment: boosting algorithms: regularization, prediction and model fitting. *Statistical Science*, 513-515.

Hettema, M. H., Jaarsma, B., Schroot, B. M., & Van Yperen, G. C. (2017). An empirical relationship for the seismic activity rate of the Groningen gas field. *Netherlands Journal of Geosciences*, s149-s161.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation 9*, 1735–1780.

Hogg, R. V., & Tanis, E. A. (2005). *Probability and Statistical Inference 9th Edition.* Pearson.

Hopfield, J. (1988). Artificial neural networks. *IEEE Circuits and Devices Magazine*, 3-10.

Hu, L.-Y., Huang, M.-W., Ke, S.-W., & Tsai, C.-F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. *Springerplus*, 1304.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning with Applications in R* (corr. 7th ed.). Springer.

Jordan, M., & Mitchell, T. (2015). Machine learning: trends, perspectives, and prospects. *Science*, 255-260.

Kadaster. (2018, Jan 20). *Rijksdriehoekstelsel.* Retrieved from https://www.kadaster.nl/rijksdriehoeksstelsel

Kim, S., Kim, H., & Namkoong, Y. (2016). Ordinal Classification of Imbalanced Data with Application in Emergency and Disaster Information Services. *IEEE Intelligent Systems*, 50-56.

Kirkpatrick, C., & Dahlquist, J. (2010). *Technical Analysis: The Complete Resource for Financial Market Technicians.* FT Press.

KNMI. (n.d.). *Live Seismogrammen*. Retrieved jan 20, 2018, from https://www.knmi.nl/nederland-nu/seismologie/stations/live-seismogrammen

Kortekaas, M., & Jaarsma, B. (2017). Improved definition of faults in the Groningen field using seismic attributes. *Netherlands Journal of Geosciences*, 71-85.

Kraaijpoel, D., Caccavale, M., Van Eck, T., & Dost, B. (2015, Mar). PSHA for seismicity induced by gas extraction in the Groningen Field. *Presentation given at the 2015 Schatzalp Workshop on Induced Seismicity*. http://www.seismo.ethz.ch/en/static/schatzalp/2015/Kraaijpoel.pdf.

Kuhn, M. (2017). *caret: Classification and Regression Training.* CRAN.

Kuhn, M., & Johnson, K. (2018). *Applied Predictive Modelling* (corr 2nd ed.). Springer-Verlag.

Kursa, M. (2010). Feature Selection with Boruta Package. *Journal of Statisical Software*, Vol. 36 Issue 11.

Langley, P. (1988). Editorial: Machine learning as an experimental science. *Machine learning*, 5-8.

Last, M., Rabinowitz, N., & Leonard, G. (2016, Jan). Predicting the Maximum Earthquake Magnitude from Seismic Data in Israel and Its Neighboring Countries. *PLOS One.*

Lele, S. P., Garzon, J. L., Hsu, S.-Y., DeDontney, N. L., Searles, K. H., & Sanz, P. F. (2015). *Groningen 2015 Geomechanical Analysis.* NAM.

Liu, W. a. (2018). Integrating machine learning to achieve an automatic parameter prediction for practical continuous-variable quantum key distribution. *Phys. Rev. A*, 022316.

M. López-Ibáñez, J. D.-L. (2016). The irace package: Iterated Racing for Automatic Algorithm. *Operations Research Perspectives - Vol. 3*, 43-58.

Magdalena Graczyk, T. L. (2010). Nonparametric Statistical Analysis of Machine Learning Algorithms for Regression Problems. *Knowledge-Based and Intelligent Information and Engineering Systems.*

Magel, R., & Wibowo, S. (1997). Comparing the Powers of the Wald-Wolfowitz and Kolmogorov-Smirnov Tests. *Biometrical Journal*, 665-675.

Makridakis, S. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE.*

Mariano, F. X. (1995). Comparing Predictive Accuracy. *Journal of Business and Economic Statistics*, 253-265.

Marzocchi, W., & Zechar, J. D. (2011). Earthquake Forecasting and Earthquake Prediction: Different Approaches for Obtaining the Best Model. *Electronic Seismologist*, 442-448. doi:10.1785/gssrl.82.3.442

Melnikov, A. A. (2018). Active learning machine learns to create new quantum experiments. *Proceedings of the National Academy of Sciences*.

Mignan, A., & Woessner, J. (2012). *Estimating the magnitude of completeness for earthquake catalogs*. Community Online Resource for Statistical Seismicity Analysis. doi:10.5078/corssa-00180805

Ministry of Economic Affairs and Climate. (2018, Mar 29). *Kabinet: einde aan gaswinning in Groningen [Cabinet: end of gas production in Groningen]*. Retrieved from Rijksoverheid Nieuws [State News]: https://www.rijksoverheid.nl/actueel/nieuws/2018/03/29/kabinet-einde-aan-gaswinning-in-groningen

Ministry of Economic Affairs and Climate. (2018, Mar 29). Kamerbrief over gaswinning Groningen [Letter to Parliament about gas production Groningen]. Netherlands. Retrieved from https://www.rijksoverheid.nl/documenten/kamerstukken/2018/03/29/kamerbrief-over-gaswinning-groningen

NAM. (2015). *Hazard and Risk Assessment for Induced Seismicity Groningen - Interim Update November 2015*. Technical Report.

NAM. (2016). *Study and Data Acquisition Plan Induced Seismicity in Groningen, Update Post-Winningsplan 2016*. NAM.

NAM. (2016). *Winningsplan Groningen Gasveld 2016*. NAM.

NAM. (2017). *Groningen Measurement and Control Protocol (Translated into English from the Original Dutch document)*. Technical Report.

NAM. (2017). *Optimisation of the Production Distribution over the Groningen field to reduce Seismicity*. Technical Report.

Nelder, J., & Wedderburn, R. (1972). Generalized Linear Models. *Journal of the Royal Statistical Society*, 370-384.

Nepveu, M., Van Thienen-Visser, K., & Sijacic, D. (2016). Statistics of seismic events at the Groningen field. *Bull Earthquake Eng, 14*, 3343-3362. doi:10.1007/s10518-016-0007-4

Paleja, R., & Bierman, S. (2016). *Measuring changes in earthquake occurrence rates in Groningen - update October 2016*. Shell Global Solutions International.

Panakkat, A., & Adeli, H. (2009). Recurrent Neural Network for Approximate Earthquake Time and Location Prediction Using Multiple Seismicity Indicators. *Computer-Aided Civil and Infrastructure Engineering*, 280-292.

Pathak, J., Hunt, B., Girvan, M., Zhixin, L., & Ott, E. (2018). Model-free prediction of large spatiotemporally choatic systems from data: a reservoir computing approach. *Physical Review Letters*, 1-5. doi:10.1103/PhysRevLett.120.024102

Perol, T., Gharbi, M., & Denolle, M. A. (2017, Feb 8). Convolutional Neural Network for Earthquake Detection and Location. *arXiv*.

Pijpers, F. P. (2016). *A phenomenological relationship between reservoir pressure and tremor rates in Groningen*. CBS [Statistics Netherlands].

Pijpers, F. P. (2016). *Trend changes in tremor rates Groningen - update Nov. 2016*. CBS [Statistics Netherlands].

Pijpers, F. P. (2017). *Interim report: correlations between reservoir pressure and earthquake rate.* CBS [Statistics Netherlands].

Pijpers, F. P., & Van der Laan, D. J. (2016). *Trend changes in ground subsidence in Groningen - update November 2015.* CBS [Statistics Netherlands].

Post, R. A. (2017). *Statistical inference for induced seismicity in the Groningen gas field.* Eindhoven University of Technology.

Postma, T., & Jansen, J.-D. (2017). The Small Effect of Poroelastic Pressure Transients on Triggering of Production-Induced Earthquakes in the Groningen Natural Gas Field. *Journal of Geophysical Research: Solid Earth*, 401-417. doi:10.1002/2017JB014809

R Core Team. (2017). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing.

Ramirez Jr., J. a. (2011). Machine Learning for Seismic Signal Processing: Seismic Phase Classification on a Manifold. *10th International Conference on Machine Learning and Applications* (pp. 382-388). IEEE.

Razali, N., & Wah, Y. B. (2011). Power comparisons of Shapiro–Wilk, Kolmogorov–Smirnov, Lilliefors and Anderson–Darling tests. *Journal of Statistical Modeling and Analytics*, 21-33.

Robert Fildes, N. K. (2011). Validation and forecasting accuracy in models of climate change. *International Journal of Forecasting*, 968-995.

Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A. (2017). Machine Learning Predicts Laboratory Earthquakes. *Geophysical Research Letters*, 9276-9282.

Ruxton, G. D. (2016). The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test. *Behavioral Ecology*. doi:10.1093/beheco/ark016

Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine Learning*, 197-227.

Shao, J. a. (1995). *The Jackknife and Bootstrap.* New York: Springer Verlag.

Spetzler, J., & Dost, B. (2017). Hypocentre estimation of induced earthquakes in Groningen. *Geophysical Journal International*, 453-465. doi:10.1093/gji/ggx020

Thalia Anagno, A. S. (1988). A review of earthquake occurrence models for. *Computational Mechanics Publications Vol 3. No. 1.*

Timothy D., M. K. (2014). Comparing Forecast Skill. *Monthly Weather Review*, 4658-4678.

TNO, Geology Service Netherlands. (n.d.). *Groningen Gasfield.* Retrieved 12 19, 2017, from NLOG: http://www.nlog.nl/en/groningen-gasfield

Udovičić, M. (2007). What we need to know when calculating the coefficient of correlation? *Biochemia Medica* .

Urbanek, S. (2015, February 20). *A simple interface to the PROJ.4 cartographic projections library.* Retrieved from rforge: http://www.rforge.net/proj4/

Van den Bogert, P. A. (2015). *Impact of various modelling options on the onset of fault slip and the fault slip response using 2-dimensional Finite-Element modelling.* Technical Report, Shell Global Solutions International.

Van den Bogert, P., & Yuan, R. (2017, Sep). Understanding Earthquakes in the Groningen Field. *Personal Communication*.

Van Oeveren, H., Valvatne, P., Geurtsen, L., & Van Elk, J. (2017). History match of the Groningen field dynamic reservoir model to subsidence data and conventional subsurface data. *Netherlands Journal of Geosciences*, s47-s54.

Van Thienen-Visser, K., Fokker, P., Nepveu, M., Sijacic, D., Hettelaar, J., & Van Kempen, B. (2015). *Recent developments on the seismicity of the Groningen field in 2015*. TNO.

Van Thienen-Visser, K., Sijacic, D., Van Wees, J.-D., Kraaijpoel, D., & Roholl, J. (2016). *Groningen field 2013 to present Gas production and induced seismicity*. TNO.

Van Wees, J.-D., Fokker, P. A., Van Thienen-Visser, K., Wassing, B. B., Osinga, S., Orlic, B., . . . Pluymaekers, M. (2017). Geomechanical models for induced seismicity in the Netherlands: inferences from simplified analytical, finite element and rupture model approaches. *Netherlands Journal of Geosciences*, s183-s202. doi:10.1017/njg.2017.38

Van Wees, J.-D., Osinga, S., Van Thienen-Visser, K., & Fokker, P. A. (2018). Reservoir creep and induced seismicity: inferences from geomechanical modeling of gas depletion in the Groningen field. *Geophysical Journal International*, 1487-1497. doi:10.1093/gji/ggx452

Visser, C. (2012). *Groningen Field Review: static modeling and hydrocarbon volume determination*. NAM.

Welch, B. (1947). The Generalization of 'Student's' Problem when several different population variances are involved. *Biometrika*, 28-35. doi:10.1093/biomet/34.1-2.28

Wiemer, S., & Wyss, M. (2000). Minimum Magnitude of Completeness in Earthquake Catalogs: Examples from Alaska, the Western United States, and Japan. *Bulletin of the Seismological Society of America*, 859-869.

Zhang, Y., Burton, H. V., Sun, H., & Shokrabadi, M. (2018). A machine learning framework for assessing post-earthquake structural safety. *Structural Safety, 72*, 1-16.

Zimmerman, D. W. (2012). Correcting Two-Sample z and t Tests for Correlation: An Alternative to One-Sample Tests on Difference Scores. *Psicológica 33*, 391-418.

15 Bibliographic information

Title	Evaluation of a Machine Learning methodology to forecast induced seismicity event rates within the Groningen Field
Author(s)	J. Limbeck (GSNL-PTX/D/S) F. Lanz (IBM Services) E. Barbaro (IBM Services) C. Harris (SUKEP-UPO/W/T) K. Bisdom (GSNL-PTX/S/RM) T. Park (GSNL-PTX/D/S) W. Oosterbosch (IBM Services) H. Jamali-Rad (GSNL-PTX/S/IA) K. Nevenzeel (IBM Services)
Keywords	Seismicity, earthquakes, event rate, activity rate, analytics, data science, machine learning, NAM, Groningen, model meta learning
Date of Issue	August 2018
Period of Work	May 2017 to July 2018
US Export Control	Non US - Non Controlled
Issuing Company	Nederlandse Aardolie Maatschappij B.V. Upstream International Schepersmaat 2 9405 TA Assen The Netherlands