Surrogate Regression Modelling for Fast Seismogram Generation and Detection of Microseismic Events in Heterogeneous Velocity Models

Saptarshi Das[1], Xi Chen[1], Michael P. Hobson[1], Suhas Phadke[2], Bertwim van Beest[2], Jeroen Goudswaard[2], and Detlef Hohl[3]

1) *Cavendish Astrophysics Group, Department of Physics, University of Cambridge, Cambridge CB3 0HE, United Kingdom. (e-mail: {sd731, xc253, mph}@mrao.cam.ac.uk)*
2) *Shell India Markets Pvt Ltd, Bangalore 562149, India. (email: {suhas.phadke, bertwim.vanbeest, jeroen.goudswaard}@shell.com).*
3) *Shell Global Solutions International BV, Grasweg 31, 1031 HW Amsterdam, The Netherlands. (email: detlef.hohl@shell.com)*

Summary

Given a 3D heterogeneous velocity model with a few million voxels, fast generation of accurate seismic responses at specified receiver positions from known microseismic event locations is a well-known challenge in geophysics, since it typically involves numerical solution of the computationally expensive elastic wave equation. Thousands of such forward simulations are often a routine requirement for parameter estimation of microseimsic events via a suitable source inversion process. Parameter estimation based on forward modelling is often advantageous over a direct regression-based inversion approach when there are unknown number of parameters to be estimated and the seismic data has complicated noise characteristics which may not always allow a stable and unique solution in a direct inversion process. In this paper, starting from Graphics Processing Unit (GPU) based synthetic simulations of a few thousand forward seismic shots due to microseismic events via pseudo-spectral solution of elastic wave equation, we develop a step-by-step process to generate a surrogate regression modelling framework, using machine learning techniques that can produce accurate seismograms at specified receiver locations. The trained surrogate models can then be used as a high-speed meta-model/emulator or proxy for the original full elastic wave propagator to generate seismic responses for other microseismic event locations also. The accuracies of the surrogate models have been evaluated using two independent sets of training and testing Latin hypercube (LH) quasi-random samples, drawn from a heterogeneous marine velocity model. The predicted seismograms have been used thereafter to calculate batch likelihood functions, with specified noise characteristics. Finally, the trained models on 23 receivers placed at the sea-bed in a marine velocity model are used to determine the maximum likelihood estimate (MLE) of the event locations which can in future be used in a Bayesian analysis for microseismic event detection.

Keywords: Synthetic seismogram generation, time domain compression, surrogate meta-model, microseismic event detection, Gaussian process regression

1. Introduction

Microseismic event detection has emerged as a significant field of research in computational geosciences with an aim of studying the changing geological characteristics of a subsurface reservoir during and after hydrocarbon production. These microseismic events are characterized by low amplitude ground movements and are often indistinguishable from environmental seismic noise (Leet 1949). A frequency band based quantification approach has been adopted in (Groos & Ritter 2009) to grossly classify such events as microtremor (>1 Hz), transitional (0.6-1 Hz) and microseismic (<0.6 Hz), although different sources and background noise in a marine environment (e.g. due to water waves, storms, shipping and anthropogenic activities like drilling) share overlapping frequency bands, making such a detection task quite challenging, using the real field datasets from marine seismic surveys. In order to reliably detect these microseismic events, recent attempts have been made to simulate approximate template seismic waves with known time-frequency domain characteristics using geo-mechanical modelling from first principles. Amongst the available approaches for the forward geophysical modelling given a heterogeneous velocity model, raytracing, acoustic wave and elastic wave propagation modelling are widely used (Chapman 2004). The elastic wave equation-based modelling is the most detailed and accurate geophysical approach for microseismic events in marine velocity models with a rock-water interface, whereas the raytracing method mostly relies on the high frequency wave propagation approximation using separate compressional (P) and shear (S) wave velocity models. In spite of the capabilities of accurate geophysical modeling, with mode conversion between P-wave and S-waves in the boundaries between the rock layers, bulk scale simulation of the 3D elastic wave equation is often not a favorable solution as it suffers from extremely high computational requirements to generate accurate synthetic seismograms from a given velocity model (including density and P/S-wave velocity) with a few million grid points for detailed description of subsurface heterogeneity. However, a GPU based generic elastic wave propagator like the k-Wave solver (Treeby et al. 2014; Treeby & Cox 2010) can reduce the computational burden of bulk simulation significantly and has been used in large scale geophysical wave propagation modelling before e.g. (Guo et al. 2016)(Das et al. 2017). But GPU based forward simulation is still not fast enough to evaluate thousands of batches of single shot elastic wave propagation simulations needed for fast computation of the likelihood values at speculative locations of microseismic events, given recorded noisy seismograms.

The outputs of the governing partial differential equation (PDE) for elastic waves are more numerous (i.e. multi-receiver seismogram time series) than the inputs to the PDE solver (specified microseismic source positions as 3D co-ordinates). Therefore, such a high dimensional mapping from the microseismic event locations to the full set of observables, i.e. the seismic waves recorded on multiple receivers, make such a statistical regression modelling problem quite challenging. This becomes even more challenging since the resulting wave-fields in response to unit strength microseismic events at random locations are sparse in nature, with time localized information embedded in the time-series data as multiple spikes. A direct regression modelling using a few thousands of sparse seismic wave-fields would smear away the predicted seismograms, as the data samples can differ by a few order of magnitude (depending on the distance from the receivers) and most of the information lies in the form of localized spikes in time. A robust compression method is thus needed for predicting the simulated seismic waves, before applying a regression framework as a "*proxy*" for the elastic wave propagator. The compression can be applied in many different ways (time, frequency or time-frequency domains e.g. using Fourier or wavelet bases). The mapping in the compressed domain needs to be smooth to capture the short duration variable amplitude oscillations in the seismic waves. Due to the time-frequency domain duality criteria, small errors incurred in the frequency domain may lead to wider and sustained spurious oscillations in time domain, which suggests the use of time domain robust compression methods over other frequency domain methods (Wood 1974). This approach slightly sacrifices the achievable compression performance and hence the number of observables in the regression model and consequently the size of the trained surrogate model to predict accurate synthetic seismograms.

With the aim of proxy construction, this paper first simulates synthetic elastic wave propagation using GPUs, from a few thousands of unit amplitude explosive microseismic events at random positions in the subsurface and records the resulting seismograms at specified receiver locations at the seabed. These synthetic data are then used to approximate, or '*statistically learn*', the underlying physics of elastic wave propagation, therefore generating a reduced physics model (Wilson & Durlofsky 2013; Wilson et al. 2012), for any random event location within the heterogeneous velocity model. Here we also compare the computational time of the full-scale forward model simulation vs. the trained surrogate meta-models to obtain an estimate of the run-time saving. This can enable an iterative microseismic source inversion process within a realistic time using standard computational resources. Such a *statistical learning* or approximation of physics in the form of PDE solver's outputs has been widely used in various surrogate meta-model assisted optimization methods before e.g. in (Forrester et al. 2008; Forrester & Keane 2009; Forrester et al. 2007).

Surrogate meta-models or proxy methods were traditionally developed for various optimization problems e.g. constrained single or multi-objective optimization problems, missing data problems etc. (Forrester et al. 2008; Forrester & Keane 2009; Forrester et al. 2007). Similar machine learning approaches have been adopted to approximate complicated likelihood functions within a Bayesian analysis framework in the blind accelerated multimodal Bayesian inference (BAMBI) algorithm (Graff et al. 2013; Graff et al. 2012)(Hobson et al. 2014). Surrogate meta-models are also used to learn weighted multiple objective functions within single-objective (Pan & Das 2015), multi-objective (Pan et al. 2014b) and robust optimization frameworks (Babaei, Pan, et al. 2015)(Babaei, Alkhatib, et al. 2015), containing expensive function calls for the forward physics simulation. The trained surrogate meta-models can be viewed as a '*proxy*' for the expensive forward simulations, while it also acts as a *smooth interpolator* in the parameter space of the forward model (i.e. microseismic event locations) which can be verified using an independent testing dataset. Such reduced physics or approximate physics based proxy or surrogate models have been widely used in various other geophysical and geological problems like shale gas production optimization (Wilson & Durlofsky 2013; Kalantari-Dahaghi et al. 2015; Wilson et al. 2012), geological CO_2 storage (Babaei, Pan, et al. 2015)(Pan et al. 2014a; Pan et al. 2014b), water injection in oil reservoirs (Babaei & Pan 2016; Babaei, Alkhatib, et al. 2015), and history matching (Goodwin 2015; Mohaghegh 2006; Rodriguez et al. 2006; Slotte & Smorgrav 2008; Zubarev 2009), in the context of optimization or uncertainty quantification using various Monte Carlo methods. To the best of our knowledge there isn't any study on surrogate meta-model or proxy development for microseismic response modelling via elastic wave propagation, using the sparse spike time series which is difficult to learn unlike in many traditional areas of computational geosciences, except few variable frequency decomposition methods for fixed receiver and source position (Modesto & de la Puente 2016) and some not well-explored concepts of seismic inversion (Weglein et al. 2009).

Seismic data driven geophysical parameter estimation and inverse problems often need a few thousands of such likelihood or objective function calls where the forward geophysics simulation produces a template seismic data to match with the noisy real recordings (Aster et al. 2011)(Mosegaard & Tarantola 2002; Tarantola 2005; Tarantola & Valette 1982; Mosegaard & Tarantola 1995). Previous surrogate-based optimization and Bayesian inference methods trained a proxy for the single valued likelihood functions since the likelihood function is dependent on the data. In applications where the data change frequently, such an approach needs retraining of the surrogate meta-model using the newly recorded data, which may be a computationally wasteful approach. Thus, we take a different approach here of directly learning the raw observables obtained from the geophysical simulation model. This poses mainly two challenges – firstly, the observables (seismograms) recorded on multiple receivers will produce too many parameters for a multivariate regression and secondly, the generated seismic data are sparse which makes it difficult to predict via a standard regression framework. In other words, for elastic wave propagation modelling, the simulated datasets are sparse in nature and also, they are dense multivariate time series, the size of which massively increases with the number of receivers and the sampling frequency. Therefore, the contribution of this paper is to *statistically learn* the sparse physical response of unit size microseismic activity, as a function of input parameter-set in the PDE (i.e. random event locations in this case) which is rather a harder problem than learning scalar valued likelihood functions within an inference problem as shown in (Graff et al. 2013; Graff et al. 2012). Amongst previous approaches to supervised learning of physics based models i.e. fewer model parameter to many observable mapping, the COSMONET algorithm (Auld et al. 2007; Auld et al. 2008) employing multilayer perceptron neural networks is worth mentioning. The present paper extends this idea for predicting sparse data using a robust compression technique. This paper also compares the performance of various smooth interpolation methods available from a pool of supervised learning techniques – starting from robust polynomial regression to kernelized shrinkage regression, support vector machine (SVM), decision tree and ensemble regression, feedforward and cascaded forward neural networks (NN) and Gaussian process (GP) regression with various kernels and basis functions. However the difference between the observable mapping, shown in COSMONET algorithm (Auld et al. 2007; Auld et al. 2008) and our approach is that we here learn each compressed domain prediction separately, rendering multiple partitioned regression models, without leveraging the underlying correlation structure amongst the observations in the compressed domain. Also, in surface seismic data based microseismic activity monitoring, the gross geological characteristics given by the voxelized 3D velocity model are not expected to change within a short span of time and can be considered constants, hence leading to a deterministic mapping of the microsiseismic event parameters on to the observed seismic profiles at various receivers. This motivates us to conceptually follow a similar route proposed in physical measurement domain observable learning as shown in the COSMONET algorithm (Auld et al. 2007; Auld et al. 2008), rather than the specific historical fixed dataset based likelihood learning as reported in the BAMBI algorithm (Graff et al. 2013; Graff et al. 2012).

Therefore, the goal of this paper is to develop a robust method to act as a proxy or surrogate meta-model or fast interpolator for mapping the input parameters in a sufficiently complex PDE model with material heterogeneity onto the sensor or measurement space to be used later in the likelihood calls for fast parameter estimation and probabilistic inference problems. In other words, the broad objective here is to teach the machine learning algorithms to rapidly predict the numerical solution of the elastic wave propagation and then use these predictions to estimate the microseismic event locations in a simple maximum likelihood or even more involved full posterior distribution estimation. Amongst previous efforts on such characterization of microseismic source activity in the subsurface from recorded seismograms using spectral, spectrogram domain methods (Eaton et al. 2014) and phase space domain using polarization diagrams in (Levy et al. 2011) are notable. Also, (Groos & Ritter 2009) proposed a scheme for classifying the sources in microtremor, transitional and microseismic events from the observed seismograms using realistic field data.

The paper is divided in the following objectives to achieve this broader goal and presented in the subsequent sections:

i) Fast GPU based synthetic seismogram simulation for training the supervised learning methods

ii) A robust compression of the sparse seismic signals

iii) Learning a smooth mapping from event location on to the compressed domain seismograms using various machine learning techniques

iv) Comparing accuracy, storage size, training time trade-offs for these supervised learning-based surrogate meta-models and

v) Using the proxy-based fast predictions for calculating maximum likelihood estimates of possible event locations

2. Synthetic Seismic Trace Generation for Training Machine Learning Algorithms

The aim of this work is to train machine learning algorithms to rapidly generate accurate seismograms within each likelihood call. It needs to be trained using some example datasets to help *statistically learn* the elastic wave propagation mechanism without numerically solving the expensive governing PDEs. To generate the synthetic seismograms, we have used the elastic wave equation solver k-Wave, in a specified 3D geometry using the pseudo-spectral method (Treeby et al. 2014). The receivers and microseismic source positions can be modelled using the given 3D voxelised heterogeneous velocity model which can be run using general purpose GPUs with a single precision (32-bit) number representation (Treeby & Cox 2010; Treeby et al. 2012). For synthetic trace generation, the medium can be modelled as simple acoustic (with only P-waves), or elastic (having both P-wave and S-wave sound velocities) or even as viscoelastic with frequency dependent absorption, which is considered as zero in the present geophysical wave propagation modelling. In the simulation process, the stress/strain tensors are iteratively updated using the specified 3D heterogeneous velocity model. During the simulation, the 3-component particle velocity and the acoustic pressure are calculated from the propagating waves at specified receiver locations. In most realistic 3D geophysical simulations, the sound velocity and density model are used for solving the forward wave propagation in acoustic mode (Phadke et al. 2000) or elastic mode (Igel et al. 1995) using a heterogeneous medium where the material properties or the velocity model with $\{c_p, c_s, \rho\}$ being specified as 3D matrices with specified voxel values. Microseismic response simulation on GPUs using the pseudo-spectral method has been explored previously in (Das et al. 2017). However the seismic data generation process using other numerical schemes of PDE discretizing methods are not the main focus here and a similar seismic wave propagation method involving either finite difference, finite element, spectral element or finite volume method can also be employed instead of the pseudo-spectral method (Igel 2016). In our simulations, in order to impose an absorbing boundary condition via the perfectly matched layer (PML), 10 grid points were reserved along each direction before and after the regular grids of the velocity model. The elastic wave propagation due to explosive microseismic sources were run on a 3D domain of 81×81×301 = 1.975×106 grid points where the grid spacing in the three directions are given by $\Delta x = 12.5, \Delta y = 12.5, \Delta z = 10$ m, therefore representing a geological model of dimension 1 km×1 km×3 km along the three directions as shown in Figure 1. The elastic wave equation is solved with a sampling time of 0.8 ms to guarantee numerical stability for this heterogeneous model over a total time interval of 2 sec and then the recorded seismograms are down-sampled to T_s = 4 ms. The strength of the sources are considered as 1 MPa as many recent literature suggest that the typical range for microseismic sources is around 1-10 MPa in sedimentary rocks and >20MPa in crystalline rocks (Rutledge et al. 1998; Collettini & Barchi 2002) whereas for earthquakes it ranges between 5-100 MPa (Dieterich et al. 2015).

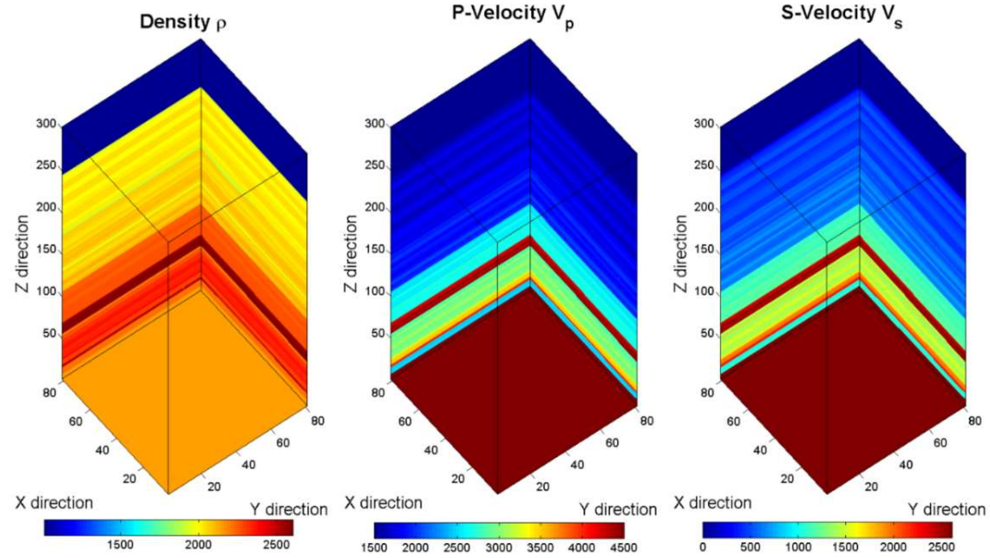


Figure 1: Heterogeneous velocity model of 1 km×1 km×3 km, comprising of the density (kg/m³), compressional and shear velocity (m/sec) at each grid-point. Heterogeneity is higher in depth compared to the lateral directions.

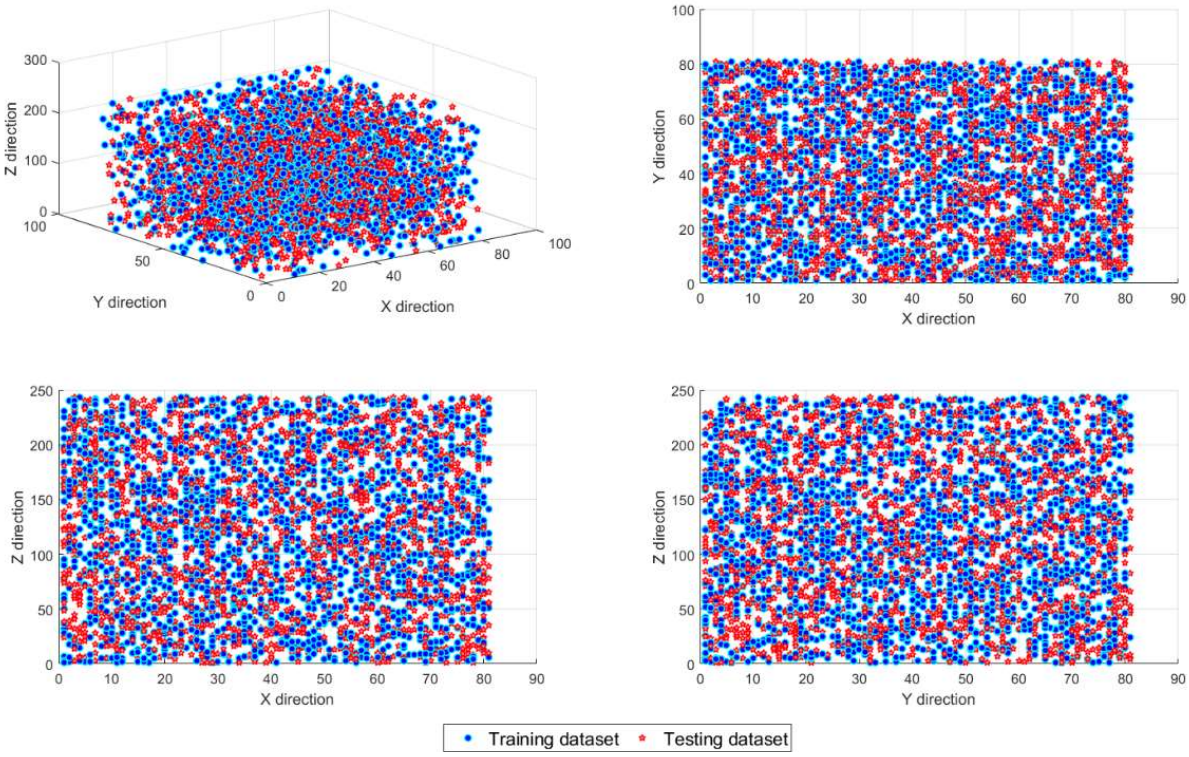


Figure 2: Latin hypercube samples for event locations for training and testing dataset in the supervised learning. Both training and testing samples are selected almost uniformly from the whole volume.

In total 4000 random Latin Hypercube (LH) samples for speculative source positions were used here for the forward simulations to generate the training and testing seismic data for different machine learning algorithms. We have randomly split 2000 source positions for training the surrogate meta-models and then use the remaining 2000 sources for testing the performance of the trained meta-model. Both these data-sets are uniformly distributed throughout the volume of the velocity model as shown in Figure 2. The forward simulations were run on the Wilkes GPU cluster at the University of Cambridge, UK with non-interactive batch mode as separate Matlab scripts. Each batch contained 40 random event locations and 10 such batches (400 sources) were run simultaneously. The generated seismic waves of 2 sec length were recorded at the surface receivers placed at the interface between the rock layers and the water column in the velocity model in Figure 1. The synthetic data after down-sampling to $T_s = 4$ ms, becomes 182 GB for 4000 LH source locations. Previous 3D elastic wave modelling attempts on GPUs e.g. (Mu et al. 2013a; Mu et al. 2013b) used 0.03×10^6 and 0.3×10^6 voxels respectively, whereas our present model is significantly larger (65.8 times and 6.5 times respectively) than the results reported there. On the 1 square kilometer surface at the sea-bed the 23 receivers are placed with the arrangement shown in Figure 3. This paper initially develops the proxy meta-model for the central receiver (R-12) and then extends it to all the 23 receivers. We also show the effect of choosing different sub-sets of these receivers on the final maximum likelihood detection performance of the events. Amongst the 4000 forward simulations, 10 representative cases are shown in the supplementary material in map view of the propagating acoustic pressure wave-field at a fixed time instant $T = 1.4$ s, where the respective microseismic source positions in the volume are mentioned in the title of the subplots. The seismic traces recorded at the 23 receiver locations can be seen in Figure 4 where the corresponding map views of the acoustic pressure wave-fields are shown in the supplementary material. On the seismogram wiggle plots in Figure 4, the appearance of multiple arrivals are actually an effect of strong P-wave, followed by weak delayed S-waves and also the receiver arrangement where source to receiver distance does not uniformly vary in different trace numbers, since the receivers are not placed along a line but distributed all over the surface.

In the next section, we aim to learn a statistical mapping between the event locations and the resulting seismograms at these 23 receivers without running the expensive forward simulations, for trained or new test event positions. In order to achieve this goal, the recorded seismic data needs to be compressed first in order to reduce the number of outputs of the surrogate regression meta-model i.e. 23 receiver × 501 time samples = 11,523 data points per microseismic event location. This 3 to 11,523 dimensional mapping is inherently a difficult learning problem because the output has complex correlation structures and moreover are sparse in nature with time localized spikes. As discussed earlier, fewer observables in the non-sparse and smooth cases can be statistically learned using multiple-input multiple-output (MIMO) regression frameworks e.g. using various neural network architectures (Auld et al. 2007; Auld et al. 2008)(Pandey et al. 2016). However most generic regression model involving nonlinear kernels like SVMs, decision tree, polynomials and Gaussian Processes can mostly accommodate a many-to-one mapping thus leading to a multiple input single output (MISO) regression problem. In general, neural networks, as universal function approximators, can accommodate both MIMO and MISO regression framework, e.g. a comparison has been reported in (Pandey et al. 2016), but in general NNs are sensitive to outliers, noise, and may not adequately learn sparse datasets, as it requires several heuristics for choosing the right combination of hidden nodes, number of layers, activation functions and optimizers. On the other hand the kernelized Gaussian process models have been widely used in geostatistical modelling and kriging that can naturally accommodate noisy data for regression and outperformed many other family of algorithms especially on regression problems as shown in (MacKay 1997)(Sitharam et al. 2008; Samui & Sitharam 2010). In order to provide a fair comparison here we have tested 9 different classes of regression models which can learn several many-to-one (MISO) mapping under the same framework i.e. given 3 event location parameters (x, y, z) the prediction of 100 compressed domain seismograms on the 23 receivers, instead of learning a many-to-many (MIMO) mapping that may capture the correlations between the data in the compressed domain components and also between different receivers. Learning this collection of many-to-one statistical mapping for the compressed seismograms gives a smooth and robust method for predicting the seismic waves due to microseismic sources, as explored in the next sections.

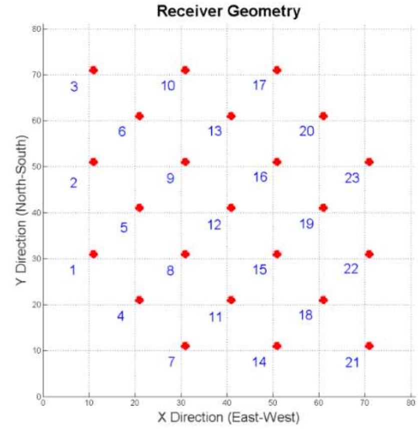


Figure 3: Receiver placement geometry in the sea-bed. Receivers are placed at fixed depth of z = 244 in a 2.44 km deep velocity model of rock layers.

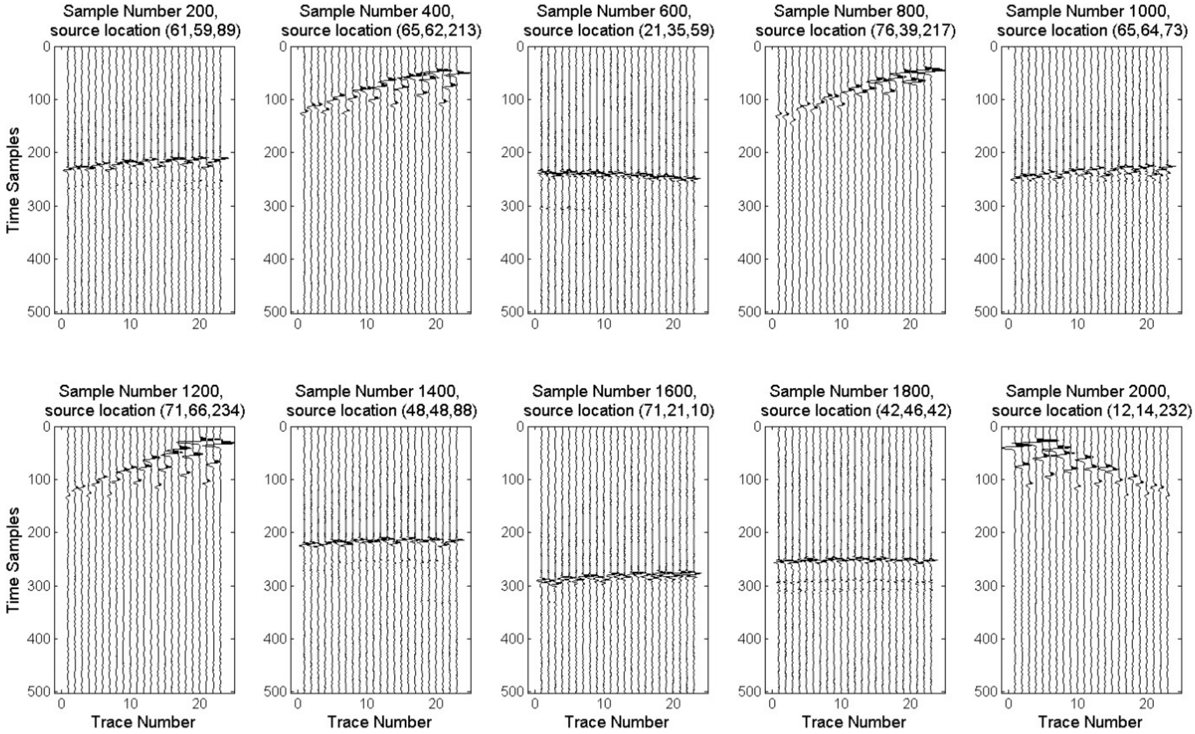


Figure 4: Seismograms for the 23 receivers recording of the acoustic pressure in the forward simulation with fixed velocity model. Sample numbers and source locations are mentioned in the subplot titles. 500 samples represent 2 sec of seismic data with Δt = 0.004.

3. Time Domain Compression of Seismic Traces and Surrogate Regression Meta-Modelling

3.1. Compressed Representation of Seismograms for Regression Meta-Modelling

This section first describes the robust time domain compression method for the time localised seismic datasets shown in Figure 4. Each seismic trace at a specified receiver location (in the horizontal x, y

plane) has been first sorted in decreasing order of absolute amplitude and only the strongest 100 samples (either capturing positive or negative pressure) are retained while the rest of the entries in the 501 sample long time-series (equivalent to 2 sec of data) are set to zero. This helps in identifying the dominant time instants within the sparse array of seismic traces, due to a smoothed delta-function like microseismic event $\delta(t)$ at different locations (x, y, z) in the heterogeneous volume. Smoothing of the source level spatial delta function is required and can be implemented using Blackman window which may otherwise create temporal oscillations, where more details on this can be found in (Treeby et al. 2012)(Das et al. 2017). This transforms the original long but sparse time series in two different components – dominant amplitude (*Si*) and the corresponding index terms (*Idx*) for these non-zero temporal instants, which are extracted for all the event locations. Such a simple time-domain compression technique is thus able to reduce the number of observables to be predicted, to a lower value (from 501 to 200 i.e. for both S_i and *Idx*, only 100 values). Upon reconstruction using the signal amplitude *Si* and the index terms *Idx*, the 2D correlation coefficient between the original and compressed images is R_{2D}>0.99, indicating almost lossless compression while also maintaining a smooth mapping of the observables in the event parameter space. Using frequency or time-frequency domain compression techniques involving Fourier or Wavelet transforms can achieve a better compression ratio but often learning the *Idx* term for such representation need to be very accurate, otherwise the compressed signals upon reconstruction may get shifted to different locations which needs further investigation. Each time domain compressed seismic dataset has been sorted in ascending order of *Idx*, thus producing a smooth pattern in the location of the dominant parts of the seismic traces. The first 500 realizations of the sorted data are shown in Figure 5.

Here the compression is done on the seismic response for a single unit amplitude microseismic event. For 2 sec of data with 501 time samples, retaining only the strongest 100 samples gives us 99% reconstructed accuracy for a single microseismic response with strong P-wave and then trailing S-waves. It is worth noting here that the purpose of the compression here is to reduce the number of regression outputs for noiseless template seismic responses for unit events and not noisy seismic traces with multiple events. For other types of datasets like different source mechanisms or different size of the velocity model, the length of the template noise free seismic trace may vary and under such a scenario, the compression ratio might need to be retuned, but a similar method needs to be adopted to reconstruct seismograms with an accuracy of R>0.99.

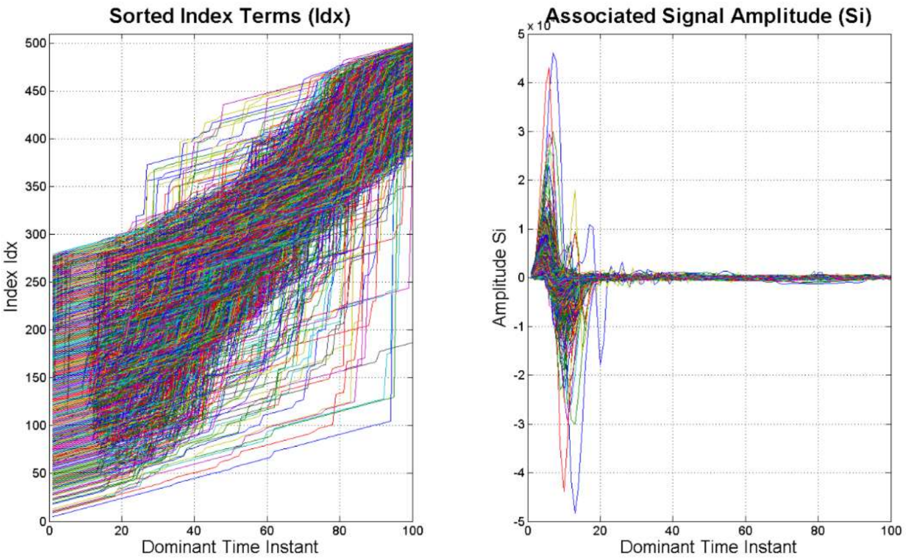


Figure 5: Sorted Index terms (Idx) and corresponding signal amplitudes (Si) on the central receiver's seismic data for the first 100 dominant time instants of 500 random microseismic events.

The time-domain robust compression method can be described using the following three steps:

Step 1: Out of the 501 samples in each 2 sec of seismogram, sort and isolate first 100 strongest positive or negative (absolute) amplitudes (*Si*)

Step 2: Sort the strongest signal values with increasing *Idx* (to get a monotonically increasing occurrence of these dominant time instants)

Step 3: Sort the strongest signal values *Si*, according to the respective time instants of *Idx*

The aim is now to map each sample of these compressed domain sorted seismic data (both *Si* and *Idx*) as a function of the event locations (x, y, z). Through such a regression modelling, the seismic traces can be accurately and smoothly interpolated within the heterogeneous medium without running the computationally expensive PDE solver for other event locations which have not been used while training the surrogate meta-model. We found that apart from the three co-ordinates of the event locations (x, y, z), in addition its distance from a fixed receiver location can also be an useful predictor for modelling seismic data recorded on that particular receiver. Here the distance (d) refers to the

Euclidean norm of the differential co-ordinates between the source and the receiver. In the next subsection, we explain with representative examples why compressing the seismic data in each receiver, prior to statistical learning or the regression modelling is a necessity.

3.2. Need for Compressed Domain Representation of the Seismic Data within Regression

The aim to predict each value in a 2D seismic snapshot at a fixed time slice is a regression problem on a sparse image with important information embedded as time localised spikes as shown in Figure 4. Therefore, a direct pixel by pixel regression approach fails to provide sufficient accuracy due to the presence of too many zeros in the training dataset, because the informative spiky signals get smeared away, under such direct regression framework. Apparently it might seem that there is a clear linear horizontal pattern for the seismic data amplitudes in Figure 6 as a function of $\bar{\mathbf{x}} = \{x, y, z, d\}$ at the four fixed time slices (t = 0.25, 0.5, 0.75, 1 sec), but actually the useful information lies only in the few outliers above and below the baseline, containing mostly low amplitude fluctuations close to zero. Therefore, in order to predict the amplitude and temporal location of such spiky seismic signals, a straight forward regression analysis cannot be applied, as any flexible machine learning algorithm will pick up most of the frequently occurring zeros and not the few time-localised spikes or outliers in a sparse seismic data. Although the maximum amplitude shows a smoother variation with respect to z and d (in the bottom row, last two entries of Figure 6 as function of z and d), it can occur anywhere in the long time trace of the seismic trace and cannot be used to recreate the full seismic wave. There is another disadvantage of such direct prediction of the sparse seismograms as a function of event parameters, apart from the computational burden of having more regression models i.e. 501 samples in the case of a single seismic trace at a single receiver for 2 sec of data. Our sparse prediction approach essentially identifies the informative region in the seismic trace and predicts only the dominant values at the respective temporal points (i.e. the two components – *Si* and *Idx*), while considering the rest of signal as sparse with zero values, whereas the voxel by voxel prediction generates small noise-like fluctuations even at locations where there is not actually any significant information.

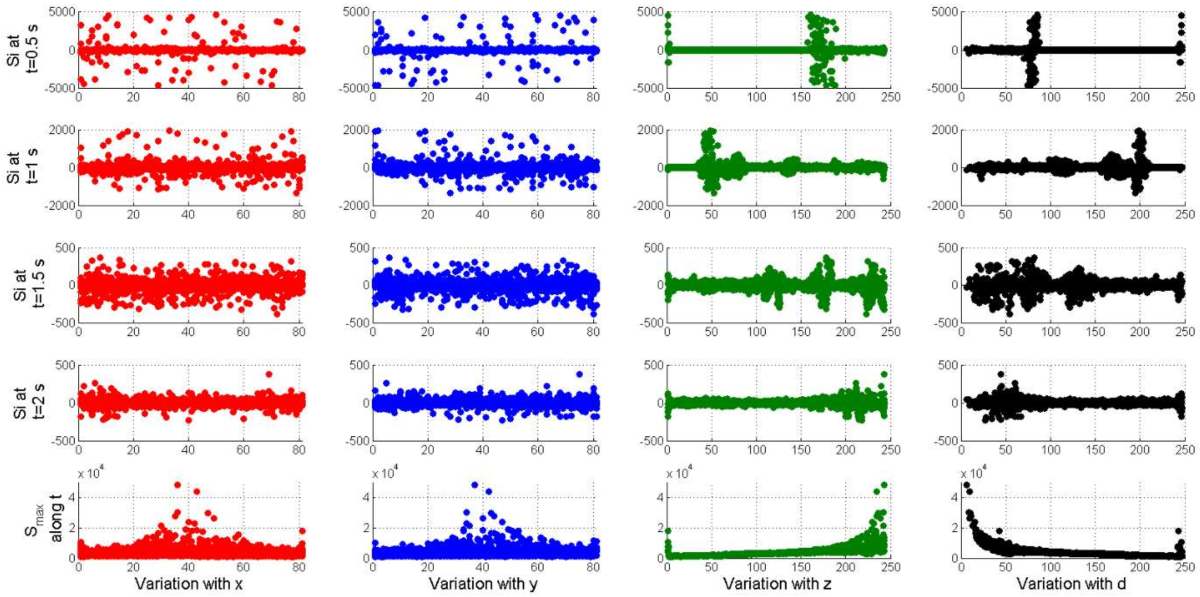


Figure 6: Variation in raw seismic signal amplitude as a function of microseismic source locations and its distance from the central receiver {x, y, z, d}. The few outliers contain the most useful information of the seismograms as high/low amplitudes.

In order to remove any bias in the regression process, which may be introduced due to the different ranges of input predictors or output observables ($\bar{\mathbf{x}}$) in the compressed domains, all the predictor and compressed observables are standardized to zero mean and unit variance using (1) and the respective standardization constants ($\mu_{\bar{\mathbf{x}}}, \sigma_{\bar{\mathbf{x}}}$) are also stored apart from the trained proxy meta-models for rescaling the new predictions to the actual physical scale:

$$\bar{\mathbf{x}}_{\mathbf{standard}} = \left(\bar{\mathbf{x}} - \mu_{\bar{\mathbf{x}}}\right) \big/ \sigma_{\bar{\mathbf{x}}} . \tag{1}$$

Compared to the uncompressed signal representation shown in Figure 6, a smoother variation is observed in the compressed domain, and can be seen from the 50th dominant time instant of the sorted seismic data in the form of its two compressed components {*Si*, *Idx*} as a function of {*x*, *y*, *z*, *d*} in Figure 7. The patterns in the compressed domain are prominent and not sparse and hidden in the form of outliers as in the previous case, as a function of these four covariates. In addition, the difference in the signal amplitudes depending on the depth of the source are also an important factor as shown in Figure 5, since the response of the deep source may get smeared away as numerical noise under a standard regression without any compression and normalization at each dominant time instant. From Figure 7, it is apparent that the two covariates {*z*, *d*} give rise to more correlated but complicated patterns with few islanded regions which might be an effect of heterogeneous nature of the velocity model and complex structure of the elastic waves with both P and S-waves in the two parts of compressed domain.

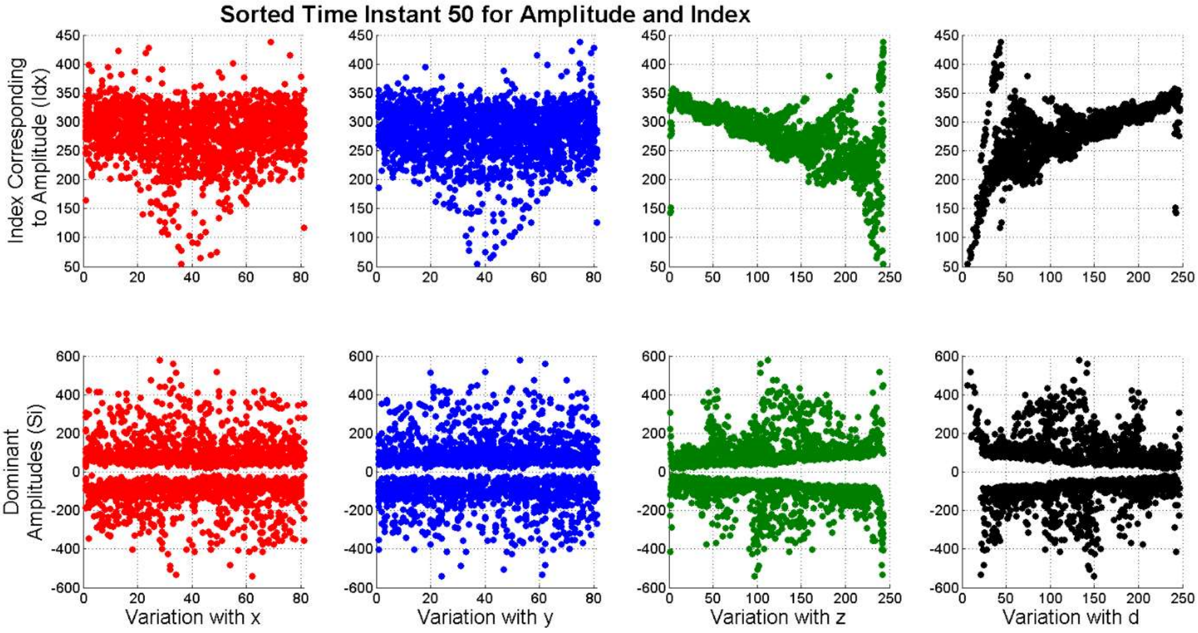


Figure 7: Variation in compressed dominant amplitudes and their index terms as a function of source location and distance from receiver {x, y, z, d}. Depth and distance show prominent structure for predicting the two compressed components.

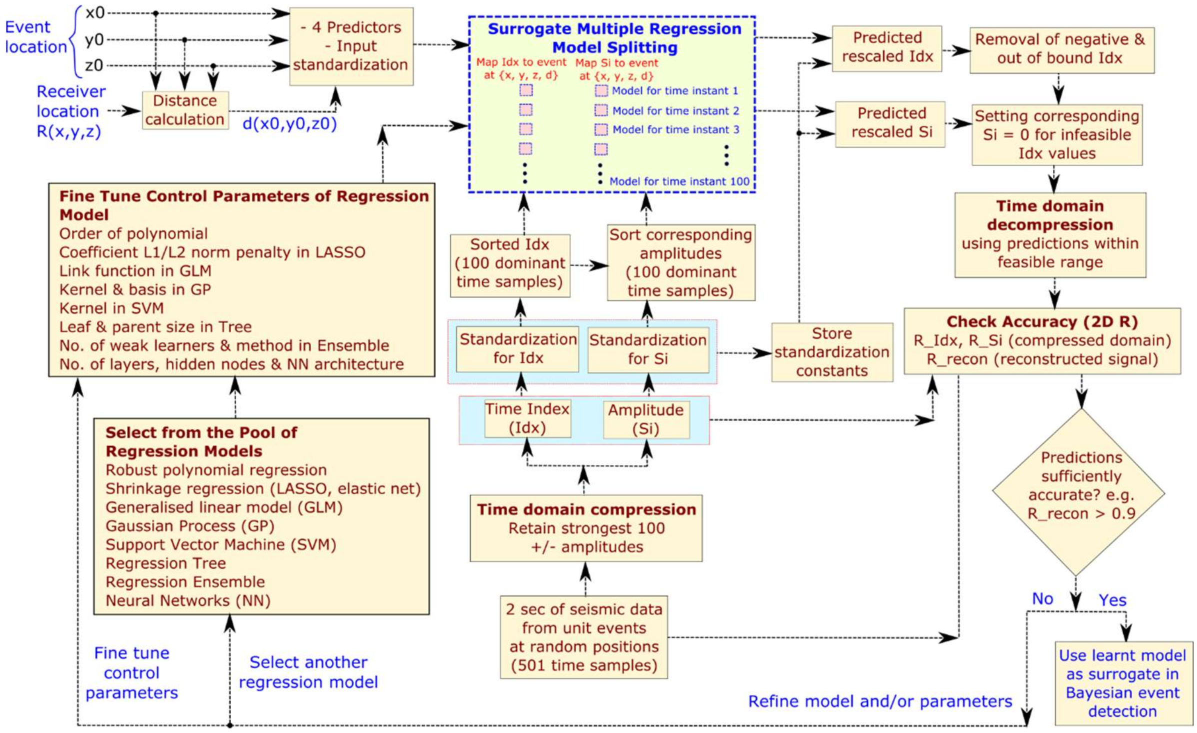

Event location
x0
y0
z0
Receiver location R(x,y,z)
Distance calculation
d(x0,y0,z0)
- 4 Predictors
- Input standardization
Surrogate Multiple Regression Model Splitting
Map Idx to event at {x, y, z, d}
Map Si to event at {x, y, z, d}
Model for time instant 1
Model for time instant 2
Model for time instant 3
Model for time instant 100
Predicted rescaled Idx
Removal of negative & out of bound Idx
Predicted rescaled Si
Setting corresponding Si = 0 for infeasible Idx values
Time domain decompression
using predictions within feasible range
Check Accuracy (2D R)
R_Idx, R_Si (compressed domain)
R_recon (reconstructed signal)
Predictions sufficiently accurate? e.g. R_recon > 0.9
No
Yes
Use learnt model as surrogate in Bayesian event detection
Refine model and/or parameters
Fine Tune Control Parameters of Regression Model
Order of polynomial
Coefficient L1/L2 norm penalty in LASSO
Link function in GLM
Kernel & basis in GP
Kernel in SVM
Leaf & parent size in Tree
No. of weak learners & method in Ensemble
No. of layers, hidden nodes & NN architecture
Select from the Pool of Regression Models
Robust polynomial regression
Shrinkage regression (LASSO, elastic net)
Generalised linear model (GLM)
Gaussian Process (GP)
Support Vector Machine (SVM)
Regression Tree
Regression Ensemble
Neural Networks (NN)
Fine tune control parameters
Select another regression model
Sorted Idx (100 dominant time samples)
Sort corresponding amplitudes (100 dominant time samples)
Standardization for Idx
Standardization for Si
Store standardization constants
Time Index (Idx)
Amplitude (Si)
Time domain compression
Retain strongest 100 +/- amplitudes
2 sec of seismic data from unit events at random positions (501 time samples)

Figure 8: Schematic diagram of time domain compression, multiple regression learning in compressed domain and decompression for predicted seismic trace generation. The unit event seismograms are compressed and learnt with parameter selection of different regression models to achieve the best predictive accuracy.

Next, we apply the same compression method on all the microseismic source locations as shown in the LH samples in Figure 2. Therefore, using a few thousand microseismic event locations, each containing 501 time-samples for a single receiver position, the time domain compression yields 100 dominant sorted time index (*Idx*) and the corresponding signal amplitudes (*Si*), i.e. 200 data points per compressed seismogram. Now, we aim to learn several regression meta-models between $\{x, y, z, d\}$ as predictors and the amplitudes of 100 *Si* and the corresponding sorted time-index *Idx* values as the observables, in order to statistically learn the patterns represented in Figure 7. After the compressed representation of the seismograms, we choose a structure from a pool of regression models and independently learn the compressed data at 100 dominant time-instants. The standardizing constants computed before the regression are stored and then used to rescale the predictions to physical units. The predicted *Si* and *Idx* values can be easily combined in the decompression step to generate the predicted seismogram, as soon as a new input location for an event comes in. Each regression model adjusts its parameters by minimizing a mean squared error criterion between the ground truth vs. the predictions in the two compressed domains in the normalized scale. Upon reconstruction and rescaling the predictions, the predictive accuracy of the two components along with the reconstructed signals are calculated between the ground truth (G_{ij}) seismograms and the corresponding predicted (P_{ij}) versions by different machine learning algorithms using the 2D Pearson correlation coefficient in (2), for choosing the best model parameters or suggesting a new model structure:

$$
\begin{gathered}
R_{2D} = \frac{\sum_i \sum_j \left(G_{ij} - \overline{G}\right)\left(P_{ij} - \overline{P}\right)}{\sqrt{\left(\sum_i \sum_j \left(G_{ij} - \overline{G}\right)^2\right)\left(\sum_i \sum_j \left(P_{ij} - \overline{P}\right)^2\right)}}, \\
\overline{G} = \frac{1}{N_i N_j} \sum_i \sum_j G_{ij}, \quad \overline{P} = \frac{1}{N_i N_j} \sum_i \sum_j P_{ij},
\end{gathered}
\tag{2}
$$

where, $\left\{\overline{G}, \overline{P}\right\}$ denote the 2D mean of the ground truth and predicted signals in either compressed/reconstructed domains.

Although the split regression models only see a smaller sub-problem with a goal of minimizing the mean squared error (MSE) between the grounds truth vs. the predictions, the combination of 200 such predictions generate the full seismic traces for all the event locations in the training dataset. In order to judge whether a structure is good enough from the pool of regression models or which parameters of the models should be fine-tuned, a fixed threshold on 2D correlation coefficient (2) as R_{2D}>0.9 has been used for the initial screening. If the model structure is found to be less flexible to accommodate the patterns in the compressed domain seismic data or a wrong control parameter is selected, a different model and/or control parameter(s) are suggested. The proxy or surrogate meta-model training workflow is schematically shown in Figure 8, starting from the event locations, then finding the compressed domain representation of seismic signals and then learning 100 split models for the dominant time indices *Idx* and the corresponding signal *Si*, from the pool of regression models, along with fine tuning of the associated control parameters. The next section briefly introduces the model structures in the pool of regression models and discusses the control parameters to learn the patterns in the seismic waves in the two-component compressed form as shown in Figure 7.

4. Machine Learning Techniques for Multivariate Compressed Domain Regression Meta-Modelling

4.1. Splitting High Dimensional MIMO Regression as Multiple MISO Regression Problems

We here explore the predictive performance, computing requirements for training and also the required storage for the trained surrogate models, using the following classes of regression techniques:

i) Robust polynomial regression

ii) Gaussian process (GP) regression

iii) Support vector machine (SVM) regression

iv) Decision tree regression

v) Ensemble regression using tree method

vi) Kernelized shrinkage regression using $\mathcal{L}_1/\mathcal{L}_2$ norm

vii) Generalized linear model (GLM) regression

viii) Kernelized shrinkage based GLM regression

ix) Multilayer Perceptron (MLP) Neural Network regression

There are also few hyper-parameters in each of the regression models that make a particular technique more flexible for learning complex patterns in the data over other classes of models. A sufficiently accurate surrogate regression meta-model can then be used in the inference or for optimization purposes, hence a comparison of storage requirements for such models and the training time are also important factors in such proxy design.

The regression models were trained in the Matlab programming platform on a 64 bit 12-core Linux CPU with 32 GB memory and Intel Xeon E5, 2.5 GHz processor, while each of the 100 regression problems for *Idx* and *Si* in Figure 8 were parallelized independently over 12 cores using the parallel for (*parfor*) loops in Matlab. In the simplest implementation, the 100 dominant compressed time instants are learned independently using a separate model without considering a correlation structure between them. This could have been otherwise learned as a 3 to 200 parameter regression problem but only MLP neural networks would be usable, with the possibility of accommodating a full MIMO regression instead of a combination of multiple MISO regression problems (Pandey et al. 2016). However, such an approach here has not yielded a good prediction accuracy due to the large number of predictors (100 or 200) compared to the covariates (only 4) using moderate size NNs and hence we here focus only on splitting the MIMO regression as a collection of multiple MISO regression problems, since here the main purpose is to get a good predictive accuracy without solving the full computationally expensive elastic PDE solver when called for fast likelihood calculation.

4.2. Predicting Compressed Domain Seismic Data at a Fixed Receiver Location as a Function of Event Location Parameters

Initially we explore the performance of various machine learning (regression) algorithms for predicting the seismic traces, recorded at the central receiver (at $N_x/2$, $N_y/2$) for the sake of simplicity. Here we use all the four predictors i.e. position and distance $\{x, y, z, d\}$, as this has been found to yield a better fitting performance over other subsets of covariates. The regression models have been assumed to be different on the S_i and *Idx*, since their patterns are found to be quite different in Figure 7. In each of the predictors we fit e.g. a higher order kernel with flexibility to adjust the order of the polynomial using an exhaustive search that maximizes the 2D correlation coefficient between the compressed vs. the regression model predicted image for both *Si* and *Idx*, hereafter called as R_{Si} and R_{Idx} respectively. The regression uses the robust option to reject outliers and fit a smoother response in its predictions. Also, the maximum order of polynomial kernel has been kept up to 4 in each predictor $\{x, y, z, d\}$, as more complex models are prone to overfit inconsistent patterns, and higher order models with more degrees of freedom have a larger number of parameters to estimate, hence producing more uncertainty in the estimates and increased variance on the predictions. The highest accuracy achieved through simple polynomial regression was insufficient because of the complexity of the data as shown in Figure 7, a representative case for sorted 50th time instant, which may not be fitted well with simple quadratic, cubic or quartic polynomial functions of the predictors. The polynomial case is used here as an example from the pool of 9 classes of regression models in Figure 8 and the different cases the polynomial order replaces the respective free hyper-parameters of the algorithm to fine tune. Also, some of the predictions for *Idx* may be negative or exceed the maximum time instant, yielding an unfeasible region (as the signals cannot lie in negative time) or increased time series upon reconstruction. Therefore, after the predictions by the regression models, the $Idx<0$ and $Idx>501$ are thresholded at the lower and upper bounds respectively and the corresponding signal amplitudes are set to zero. In both the predictions of *Si*, *Idx* and the reconstructed seismic traces, the 2D correlation coefficient is used as reported in the tables as $\{R_{Si}, R_{Idx}, R_{Recon}\}$ using (2). It is to be noted that in all the cases, the machine learning algorithms have been trained on the two compressed components of the signal to select the model with maximum $\{R_{Si}, R_{Idx},\}$. The reconstructed accuracies on the seismograms are calculated outside the training process to calculate R_{Recon}.

4.3. Robust Polynomial Regression

The robust regression method gives better estimates in the presence of outliers and noise, compared to the commonly used ordinary least square (OLS) method incorporating the Moore-Penrose pseudo-inverse. Let us consider the regression problem in (3), $X_i = \begin{bmatrix} x_i & y_i & z_i & d_i \end{bmatrix}^T$ is the predictors and

$Y_i = \left[Idx_i \quad Si_i \right]^T$ is the observables with model weight β and the prediction error (ε_i) being independent and identically distributed (iid) with a scale factor σ for the modelling error:

$$Y_i = X_i^T \beta + \sigma \varepsilon_i. \tag{3}$$

The estimate of the weight $\widehat{\beta}$ can be calculated from a given estimate of scaling factor $\widehat{\sigma}$, considering the data and the weighted function (ψ) of error to be uncorrelated i.e.

$$(1/N)\sum_{i=1}^{N} X_i \psi\left(\left(Y_i - X_i^T \widehat{\beta}\right)/\widehat{\sigma}\right) = 0. \tag{4}$$

In the robust regression, starting from an initial estimate $\left\{\widehat{\beta}, \widehat{\sigma}\right\}$, residuals of the estimates are calculated as $r_i = \left(Y_i - X_i^T \widehat{\beta}\right)/\widehat{\sigma}$. The weights are defined as $w_i = \psi(r_i)/r_i$ and the estimates are updated with a least square estimate with weight w_i. The iterative update continues unless the algorithm converges (Street et al. 1988). There can be different choice of weight functions (Holland & Welsch 1977) for robust regression as in (5):

$$\begin{aligned}
&\text{Andrews: } w = (|r| < \pi)\sin(r)/r, \text{ Bisquare: } w = (|r| < 1)\left(1 - r^2\right)^2, \text{ Cauchy: } w = 1/\left(1 + r^2\right), \\
&\text{Fair: } w = 1/\left(1 + |r|\right), \text{ Huber: } w = 1/\left(1 + \max\left(1, |r|\right)\right), \text{ Logistic: } w = \tanh(r)/r, \\
&\text{Welsch: } w = \exp\left(-r^2\right), \text{ Talwar: } w = |r| < 1.
\end{aligned} \tag{5}$$

Here the value of r is calculated as $r = res/\left(const_{tune}\widehat{\sigma}\sqrt{1-h}\right)$, with $\{res, h\}$ being the residual from the previous iteration and leverage value from OLS fit respectively. The standard deviation of error is calculated as $\widehat{\sigma} = MAD/0.6745$, using the median absolute deviation (MAD) of the residuals, considering it to be normally distributed.

In the present problem with robust polynomial regression, we first transform the input parameter space i.e. microseismic event locations and distance from the central receiver $\{x, y, z, d\}$ using a polynomial kernel function *xnfx* (Ieong 2012) with a chosen order of 2 to 4 in order to form a design matrix e.g. $\{1, x, y, z, xy, xz, yz, x^2, y^2, z^2, \ldots\}$ in the case of quadratic kernel, as an example. This high dimensional transformed feature matrix is then used in the robust linear regression framework through the weight functions in (5). The maximum order of the kernel has been chosen as 4 keeping in mind the lower number of predictors (also 4) and to lower the possibility of overfitting. The results of robust polynomial regression are reported in Table 1.

4.4. Gaussian Process (GP) Regression

Starting from a linear model ($Y = X^T \beta + \varepsilon, \varepsilon \sim \mathcal{N}\left(0,\sigma^2\right)$) the GP explains the prediction using the latent variables $F\left(X_i\right), i = 1,2,\cdots,n$ (for modelling the smoothness of the output) and the explicit basis *H* (for projecting predictors in high dimensional space). If $F\left(X\right), X \in \mathbb{R}^d$ be a GP having mean $m\left(X\right)$ and covariance $k\left(X_i, X_j\right)$, then given *n* observations $\{X_1, X_2, \cdots, X_n\}$ the joint distributions of the latent variables $\left\{F\left(X_1\right), F\left(X_2\right), \cdots, F\left(X_n\right)\right\}$ are also Gaussian. Now let us consider the model as (6), with $H\left(X\right), H:\mathbb{R}^d \rightarrow \mathbb{R}^p$ being the basis and coefficients of the basis are $\beta \in \mathbb{R}^{p\times 1}$:

$$Y = H\left(X\right)^T \beta + F\left(X\right), F\left(X\right) \sim \mathcal{GP}\left(0, k(X_i, X_j)\right). \tag{6}$$

The probabilistic predictions of GP regression is given by (7):

$$P\left(Y_i \middle| f\left(X_i\right), X_i\right) \sim \mathcal{N}\left(Y_i \middle| H\left(X_i\right)^T \beta + F\left(X_i\right), \sigma^2\right). \tag{7}$$

The GP regression utilizes the fact that two closely lying predictor values $\left\{X_i, X_j\right\}$ will have similar response $\left\{f\left(X_i\right), f\left(X_j\right)\right\}$ and the similarity is represented by the kernel or covariance function $k\left(X_i, X_j \middle| \theta\right)$ with the hyper-parameter vector θ . The kernels vary mainly due to two parameters i.e. the signal standard deviation (σ_f) and characteristic length scale (σ_l) which control how fast the correlation between two points change. Given a set of input-output data the GP algorithm estimates the basis coefficients β , noise variance and the kernel hyper-parameters θ . We used three different basis functions where the model is extended by different basis matrix (*H*) by multiplying with the vector of basis coefficients (*β*) i.e. the extended model becomes $H\times\beta$. For the constant, linear and quadratic cases, the basis matrix can be represented as (8)

$$\text{constant} \Rightarrow H = 1; \quad \text{linear} \Rightarrow H = [1, X];$$
$$\text{quadratic} \Rightarrow H = [1, X, X''],\ X'' = \begin{bmatrix} X_{11}^2 & X_{12}^2 & \cdots & X_{1d}^2 \\ X_{21}^2 & X_{22}^2 & \cdots & X_{2d}^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1}^2 & X_{n2}^2 & \cdots & X_{nd}^2 \end{bmatrix}. \tag{8}$$

Along with variation in the basis function we also explored six different kernel functions – squared exponential, Matern 3/2, Matern 5/2 and also their automatic relevance discovery (ARD) versions (Neal 1996; Rasmussen & Williams 2006):

$$\begin{aligned} k_{squared-\exp}\left(X_i, X_j \middle| \theta\right) &= \sigma_f^2 \exp\left[-\left(X_i - X_j\right)^T \left(X_i - X_j\right) / 2\sigma_l^2\right] \\ k_{Matern-3/2}\left(X_i, X_j \middle| \theta\right) &= \sigma_f^2 \left(1 + \frac{\sqrt{3}r}{\sigma_l}\right) \exp\left(-\frac{\sqrt{3}r}{\sigma_l}\right) \\ k_{Matern-5/2}\left(X_i, X_j \middle| \theta\right) &= \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) \exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right), \end{aligned} \tag{9}$$

where, $r = \sqrt{\left(X_i - X_j\right)^T \left(X_i - X_j\right)}$ is the Euclidean distance between the points $\{X_i, X_j\}$.

Considering separate length scale (σ_m) for each of the predictors ($m = 1, 2, \cdots, d$), the covariance (kernel) function implementing the ARD (Neal 1996) takes the form in (10):

$$\begin{aligned} k_{ARD-squared-\exp}\left(X_i, X_j \middle| \theta\right) &= \sigma_f^2 \exp\left[-\frac{1}{2}\sum_{m=1}^{d} \frac{\left(X_{im} - X_{jm}\right)}{\sigma_m^2}\right] \\ k_{ARD-Matern-3/2}\left(X_i, X_j \middle| \theta\right) &= \sigma_f^2 \left(1 + \sqrt{3}\tilde{r}\right) \exp\left(-\sqrt{3}\tilde{r}\right) \\ k_{ARD-Matern-5/2}\left(X_i, X_j \middle| \theta\right) &= \sigma_f^2 \left(1 + \sqrt{5}\tilde{r} + \frac{5}{3}\tilde{r}^2\right) \exp\left(-\sqrt{5}\tilde{r}\right) \end{aligned} \tag{10}$$

where, $\tilde{r} = \sqrt{\sum_{m=1}^{d} \left(\left(X_{im} - X_{jm}\right)^2 / \sigma_m^2\right)}$.

The training of ARD kernels in (10) e.g. ARD Materns are computationally more expensive than their basic versions in (9) e.g. the simple Matern kernels. The ARD function automatically finds out moving how far along a particular predictor will make the predictions uncorrelated. The inverse of the length scale determines how relevant a predictor is, as also discussed in (Rasmussen & Williams 2006) in a detailed manner. In all the regression models, the compact representation was adopted to store the models as *compact Matlab objects* that stores only the necessary information, instead of the full model with information about training data etc. for a reduced storage requirement. Also, while training the GP models with constant basis, sometimes the Cholesky decomposition of the covariance can be ill-conditioned, resulting in convergence failure of the GP algorithms. In such a case, a different initial value of the kernel parameters, initial value and increased lower bound of the GP noise standard deviation may improve the solution. For all the GP kernels, the computation for the log-likelihood and gradient, the standard QR factorization and Quasi-Newton optimizer have been used for parameter estimation. Gaussian processes with certain kernels are known to have equivalent representation of neural networks with infinite hidden nodes and are found to outperform many benchmark supervised learning methods, especially complex regression problems due to its non-parametric Bayesian nature (MacKay 1997)(Sitharam et al. 2008; Samui & Sitharam 2010), albeit being computationally expensive during the training process, compared to the NN learning. The GP based predictions in the compressed as well as reconstructed domain signals along with the training time and storage requirements are shown in Table 2. Here, the accuracies of the GP regression models are worth noticing in comparison with the other classes of regression structures in Figure 8.

4.5. Support Vector Machine (SVM) Regression

In SVM regression similar to the simple linear regression problem $Y = \beta^T X + b$, the inputs ($X_i, i = 1, \cdots, n$) can be mapped to a high dimensional space using a kernel $\Phi\left(X_i, X_j\right)$. We used three popular kernels viz. linear, polynomial and radial basis function (RBF) kernel in (11), with p being the tuning parameters for the polynomial kernel (Friedman et al. 2001)(Rogers & Girolami 2015):

$$\begin{aligned} \Phi\left(X_i, X_j\right) &= X_i^T X_j && :\text{Linear} \\ &= \left(1 + X_i^T X_j\right)^p, p = \{2,3,4\} && :\text{Polynomial}. \\ &= \exp\left(-\left\|X_i - X_j\right\|^2\right) && :\text{RBF} \end{aligned} \tag{11}$$

The SVM regression algorithm terminates using either of the three convergence criteria – feasibility gap (Δ), gradient difference (∇L), or largest Karush-Kuhn-Tucker (KKT) violation. The KKT criteria act as constraints to the optimization problem which can be solved using the sequential minimal optimization (SMO) algorithm which is faster than the traditional quadratic programming approach (Huang et al. 2006). The comparative results of the SVM regression models using the above three kernels *viz.* linear, polynomial kernel of order 2-4 and the RBF have been reported in the supplementary material as they do not yield high enough accuracy.

4.6. Decision Tree Regression

In the decision tree regression, deep trees are grown first and then the optimal sequence of subtrees are determined by pruning. Firstly, the predictor space is divided into non-overlapping regions $R_j, j=1,2,\cdots,J$ and then for every observation falling in a particular region, the prediction becomes mean of the response values in R_j. The regions are found out by minimizing the sum of squared error (SSE) in (12) between the real (Y_i) and the mean response of training samples within a particular box ($\widehat{Y}_{R_j}$) (James et al. 2013):

$$SSE = \sum_{j=1}^{J} \sum_{i \in R_j} \left(Y_i - \widehat{Y}_{R_j} \right)^2 . \tag{12}$$

As the controlling parameters, the effects of varying the minimum number of leaf nodes (N_{leaf}) and minimum number of parent/branch nodes (N_{parent}) are explored here and the accuracy vs. data storage size trade-offs are also shown in the supplementary material. Tree methods allow complicated nonlinear and partitioned boundaries as non-overlapping regions, especially naturally modelling corners in the input parameter space which are difficult to model with other regression models, thus often providing good predictive accuracy, although generalization and storage size for large trees are inherent challenges (James et al. 2013). Also, the surrogate split option is used which is known to improve predictive accuracy by randomly splitting the data at most 10 times in each leaf node. Pruning of decision trees is another option that has been used to produce a smaller tree with fewer splits. The results of the tree methods with different leaf and parent size (3 to 20) have been reported in the supplementary material.

4.7. Ensemble Regression with Tree Method

In ensemble regression, normally two algorithms are commonly used viz. least square boosting (LSBoost) and bootstrap aggregation (Bag) (Barutçuouglu & Alpaydin 2003). Bagging grows multiple weak learner trees on many resampled (bootstrap) replicas of the dataset and the predicted response is the average prediction from all these trees. Minimal leaf size of bagged regression tree is kept fixed at 5 and as the controlling parameter the number of learners (N_{learn}) are varied from 100 to 1000. In LSBoost every step fits a new learner using the difference between the observed and the aggregated prediction of all the learners trained so far while minimizing the MSE. While using the bagging method, the size of the surrogate meta-models become huge (>1 GB) with just 100 learners and for just one receiver location. Therefore, bulk scale simulation using this approach is not recommended due to unmanageable storage size of the trained proxy meta-models. The performance results of the ensemble regression with several independently grown tree learners (100 to 1000) with bagging and boosting method have been compared in the supplementary material.

4.8. Shrinkage Regression with Polynomial Kernel

When dealing with redundant or few less important predictors, the Shrinkage methods give improved performance over traditional regression methods that gives more priority on significant predictors over the insignificant ones. In our four predictor $\{x, y, z, d\}$ based regression problem, the covariates are first projected on to a higher dimensional space using a polynomial kernel of order 2-4 via the kernel function *xnfx* (Ieong 2012), since in many cases the kernel order for the design matrix exceeding the dimension of the original inputs yield spurious results. As an example, a 3D event location $\{x, y, z\}$ under such a 3rd order polynomial kernel mapping would yield series of predictors like $\{1, x, y, z, xy, xz, yz, x^2, y^2, z^2, xyz, x^3, y^3, z^3\}$ etc. Amongst these combinations, the shrinkage methods are expected to pick up the most useful predictors from these new kernelized predictors, while pushing rest of the insignificant ones to zero. The three variants of shrinkage regression i.e. Least absolute shrinkage and selection operator (Lasso), elastic net and Ridge regression solve the following minimization problem in (13) as weighted sum of prediction error and penalty term on the coefficients (Zou & Hastie 2005):

$$\min_{\beta_0,\beta}\left[\frac{1}{2N}\sum_{i=1}^{N}\left(Y_i-\beta_0-X_i^T\beta\right)^2+\lambda P_\alpha\left(\beta\right)\right] \tag{13}$$

where, $P_\alpha\left(\beta\right)=\alpha\left\|\beta\right\|_1+\left(\left(1-\alpha\right)/2\right)\left\|\beta\right\|_2$ is the penalty term of elastic net interpolating between the mixture of $\mathcal{L}_1/\mathcal{L}_2$ norm of the model coefficients and N is the number of samples for training. The above elastic net problem approaches the Lasso at $\alpha=1$, thus making Lasso penalize the $\mathcal{L}_1$ norm only, whereas the problem approaches Ridge regression when $\alpha\to0$ thus giving full penalty on the $\mathcal{L}_2$ norm only (Zou & Hastie 2005; Friedman et al. 2001). Therefore, to implement the three Shrinkage regression methods $\alpha=\left\{1,0.5,10^{-6}\right\}$ have been considered using the four kernelized predictors $\{x,y,z,d\}$ with an increasing polynomial order of 2-4. The regularization parameter (λ) in (13) controls the penalty between the prediction error and a chosen norm ($\mathcal{L}_1/\mathcal{L}_2$) of the model coefficients. A 10-fold cross validation has been adopted to automatically choose the optimum λ with minimum average error across the folds of the training data and hence the best model is automatically chosen with optimum λ for each of the 100 multiple-regression problems. Apart from the 10-fold cross-validation based optimum model selection or kernel hyper-parameter tuning during the training phase, the best models on the training set are also tested with a separate hold out dataset which is explored in the subsequent sections.

4.9. Generalised Linear Model (GLM) Regression

GLM is a special class of nonlinear models that still use linear methods for prediction. A linear model ($\mu = X\beta$) based predictions can be interpreted as a normal distribution with mean μ where coefficients β map each input on to the predictions linearly. In GLM (14) the response can have a wide variety of distributions $f(\cdot)$, known as the link function with mean μ :

$$f(\mu)=X\beta\ . \tag{14}$$

For normal distribution the link function becomes the mean i.e. $f(\mu)=\mu$, but for other complex distributions, the canonical link functions and the mean inverse functions can be chosen in different ways. For real valued outputs, choosing a normal distribution in GLM is recommended which suits our standardized outputs, whereas for positive/strictly integer values, other distributions like gamma, inverse gamma, Poisson or binomial can also be used.

4.10. Shrinkage Based GLM Regression with Polynomial Kernel

These regression techniques have the advantages of both the Shrinkage and GLM methods, as described in the earlier subsections. Similar to the standard elastic net, norm based penalties are chosen as $\alpha = \left\{1, 0.5, 10^{-6}\right\}$ to implement Lasso, elastic net and Ridge regression respectively. In addition, a 10-fold cross validation on the training data is also implemented to automatically select the regularization parameter λ in each of the multiple regression sub-problems. A normal distribution on the outputs and an identity link function is considered in Lasso-GLM and other variants (Friedman et al. 2010). For the case of normal distribution as the link function, the predictions closely approach the base versions without the GLM, whereas GLM enhancements are more popular in classification problems over the regression problems. The kernelized shrinkage, GLM and kernelized shrinkage GLM based prediction results have been compared in the supplementary material, where none of them attain a good predictive accuracy.

4.11. Multi-Layer Perceptron (MLP) Neural Network (NN) Regression

Neural networks are widely used as universal function approximators and thus a popular choice in many regression problems using a multiple inputs and multiple outputs (MIMO) architecture. With an aim of a fair comparison with other regression methods, we here employ a collection of multiple input single output (MISO) implementation of MLP neural networks with moderate size hidden nodes. Often neural networks are prone to pick up inconsistent patterns or outliers in the data, thus we used a regularization constant of $\gamma = 0.5$ in the cost function (J_{reg}), to keep an equal balance on both the penalties due to the MSE and the mean squared weight (MSW) during the training process:

$$\begin{aligned} J_{reg} &= \gamma \times MSW + \left(1-\gamma\right) \times MSE \\ &= \gamma \times \left(1/M\right) \sum_{j=1}^{M} w_j^2 + \left(1-\gamma\right) \times \left(1/N\right) \sum_{i=1}^{N} \left(Y_i - \widehat{Y}_i\right) \end{aligned} . \tag{15}$$

Although there have been recent developments on optimizers for fast training of large and deep networks in classification problems, the traditional Levenberg-Marquardt (LM) backpropagation algorithm has been shown to outperform on a wide variety of regression problems as it produces low MSE and high speed for training small to medium size networks with <1000 weights and bias terms (Plumb et al. 2005). As the size of the network grows, there are even efficient optimizers like scaled conjugate gradient (SCG) compared to traditional training algorithms like Levenberg-Marquardt. In the present scenario, the whole dataset here during the NN training has been randomly divided in training (70%), testing (15%) and validation (15%) set for each of the MISO regression problems. The hyperbolic tangent sigmoid (*tansig*) activation function in (16) is employed in the hidden layers and a pure linear (*purelin*) activation function in the output layer which is commonly used for regression modelling:

$$a = \text{tansig}\left(n\right) = \frac{2}{1+e^{-2n}} - 1 = \frac{1-e^{-2n}}{1+e^{-2n}}, \; a = purelin\left(n\right) = n \; . \tag{16}$$

We also explored two different NN architectures – feedforward and cascaded-forward networks with single and double layer while the number of nodes is varied from 10 to 100 in each hidden layer to keep the storage and training time comparable with other methods. The cascaded forward network has similar architecture like feedforward networks except that it has an extra connection to the input directly in each hidden layer, apart from the inputs from previous layer. The comparative performance of these two NN architectures are shown in the supplementary material.

5. Results and Discussion

From the velocity model in Figure 1, it is evident that along the *y*-direction, there is relatively small variation in the rock properties, compared to the variation along the *x*-direction. Also, the density and P/S-wave velocities have rapid variation along the *z*-direction. Therefore, the effect of the heterogeneity will be different on different receivers placed at the sea-bed and thus finally affecting the likelihood calculation in different ways. We here explore 6 different receiver arrangements for calculating the likelihood by taking a subset of the 23 receivers, as shown in Figure 3 (all 23, along principle diagonal, anti-diagonal, central one, lower and upper triangular parts). In this section, the best regression model from the previous section has been selected and the parameters are fine-tuned based on the seismic responses recorded at the central receiver from the microseismic sources anywhere in the subsurface as shown in Figure 2.

5.1. Prediction of a Single Seismogram at the Central Receiver

Table 1: Accuracy, computation time and size of the robust polynomial regression learning surrogates

Polynomial Order	**Robustness Criteria**	$\boldsymbol{R_{Si}}$	$\boldsymbol{R_{Idx}}$	$\boldsymbol{R_{Recon}}$	**Training time (s)**	**Model Size for single Receiver (KB)**
2	Andrews	0.7720	0.9626	0.3406	3.70	16
	Bisquare	0.7723	0.9626	0.3402	3.52	16
	Cauchy	0.7864	0.9661	0.3388	3.45	16
	Fair	0.7986	0.9679	0.3291	3.44	16
	Huber	0.7932	0.9672	0.3283	3.43	16
	Logistic	0.7946	0.9674	0.3322	3.85	16
	OLS	0.7913	0.9695	0.0951	3.13	15
	Talwar	0.7720	0.9629	0.3324	3.30	16
	Welsch	0.7756	0.9635	0.3410	3.54	16
3	Andrews	0.8451	0.9665	***0.7822***	4.32	21
	Bisquare	0.8453	0.9665	0.7798	4.62	21
	Cauchy	0.8531	0.9700	0.5995	4.00	21
	Fair	0.8616	0.9720	0.6258	3.83	21
	Huber	0.8587	0.9713	0.6312	3.88	21
	Logistic	0.8593	0.9715	0.6254	4.08	21

	OLS	0.8706	0.9736	0.3289	3.31	21
	Talwar	0.8454	0.9667	0.7814	3.58	21
	Welsch	0.8465	0.9671	0.6816	4.22	21
4	Andrews	0.8648	0.9725	0.7119	4.68	25
	Bisquare	0.8642	0.9725	0.7081	4.65	25
	Cauchy	0.8743	0.9759	0.7144	4.44	25
	Fair	0.8825	0.9776	0.7611	4.14	25
	Huber	0.8799	0.9770	0.7374	4.26	25
	Logistic	0.8803	0.9771	0.7414	4.25	25
	OLS	0.8827	0.9791	0.1963	3.54	25
	Talwar	0.8670	0.9729	0.6999	4.06	25
	Welsch	0.8660	0.9732	0.7091	4.54	25

Table 2: Accuracy, computation time and size of the Gaussian process learning surrogates

Basis	**Kernel**	R_{Si}	R_{Idx}	R_{Recon}	**Time in hour**	**Compact Model Size (in MB)**
Quadratic	Squared Exponential	0.9575	0.9911	0.7851	1.96	5.017
	Matern 3/2	0.9728	0.9939	0.8729	2.53	5.024
	Matern 5/2	0.9647	0.9925	0.7136	2.42	5.023
	ARD Squared Exponential	0.9631	0.9945	0.9377	29.68	5.032
	ARD Matern 3/2	0.9707	0.9963	0.9377	9.29	5.031
	ARD Matern 5/2	0.9668	0.9956	0.9427	9.79	5.03
Linear	Squared Exponential	0.9522	0.9908	0.8125	2.04	5.022
	Matern 3/2	0.9535	0.9938	0.8434	3.25	5.02
	Matern 5/2	0.9624	0.9922	0.6873	3.16	5.02

	ARD Squared Exponential	0.9625	0.9943	0.9186	4.65	5.032
	ARD Matern 3/2	0.9696	0.9962	***0.9467***	8.12	5.027
	ARD Matern 5/2	0.9664	0.9954	0.9431	9.64	5.027
Constant	Squared Exponential	0.9453	0.9902	0.8617	1.78	5.036
	Matern 3/2	0.9692	0.9916	0.8312	4.60	5.036
	Matern 5/2	0.9587	0.9919	0.6559	3.29	5.051
	ARD Squared Exponential	0.9621	0.9937	0.8812	36.19	4.995
	ARD Matern 3/2	0.9703	0.9957	0.9428	7.08	4.941
	ARD Matern 5/2	0.9409	0.9857	0.2389	1.53	4.991

Here the central receiver (R-12 in Figure 3) is considered to be seated at a fixed location ($Nx/2$, $Ny/2$) at z = 244th grid point, whereas the sources can roam around anywhere in the rock volume underneath. We first aim to predict the seismic traces at the central receiver using a regression meta-model, fitted using the 2000-unit amplitude microseismic events at different LH sample locations as shown in Figure 2. Under an exhaustive search for the best polynomial kernel combining the right polynomial order in the 4 different predictors, it is revealed from Table 1 that using the random source positions, a 3rd order polynomial with Andrews robustness criterion yields the best prediction accuracy upon reconstruction, with a 2D correlation coefficient of $R_{Si} = 0.8451$, $R_{Idx} = 0.9665$ and $R_{Recon} = 0.7822$ with respect to the original $N_{source} \times N_t = 2000 \times 501$ samples of seismic dataset. The other combinations like 3rd and 4th order polynomials and different robustness criteria work fairly similarly, except the ordinary least square as this is prone to outliers and non-normal datasets. The training time and the storage of the robust regression coefficients are minimal amongst all the proxy meta-models, explored in this section. Other complex models can push the predictive accuracy to a higher value which are explored next although they need more computational time for training. In this section, we show comparison of different classes of regression models in terms of training accuracy (R_{2D}) for the two compressed parts having 100 dominant time instants, accuracy of the reconstructed seismograms, training time and proxy storage size, utilizing the 2000 training samples, with 10-fold cross validation to select hyper-parameters of different family of regression models. The best classes of models found with this exploration as an initial screening has been further tested with 2000 independently held out testing samples and reported in the following sections.

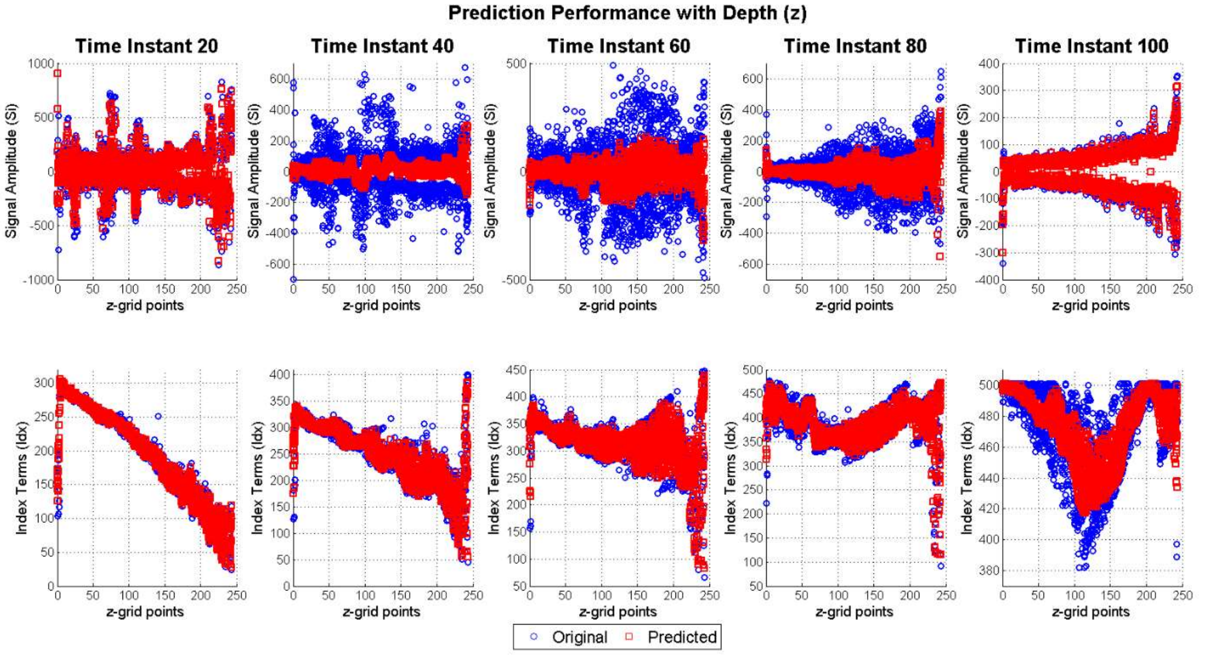


Figure 9: Ground truth vs. predicted variation in Si and Idx along depth (z-direction) using Gaussian process surrogates.

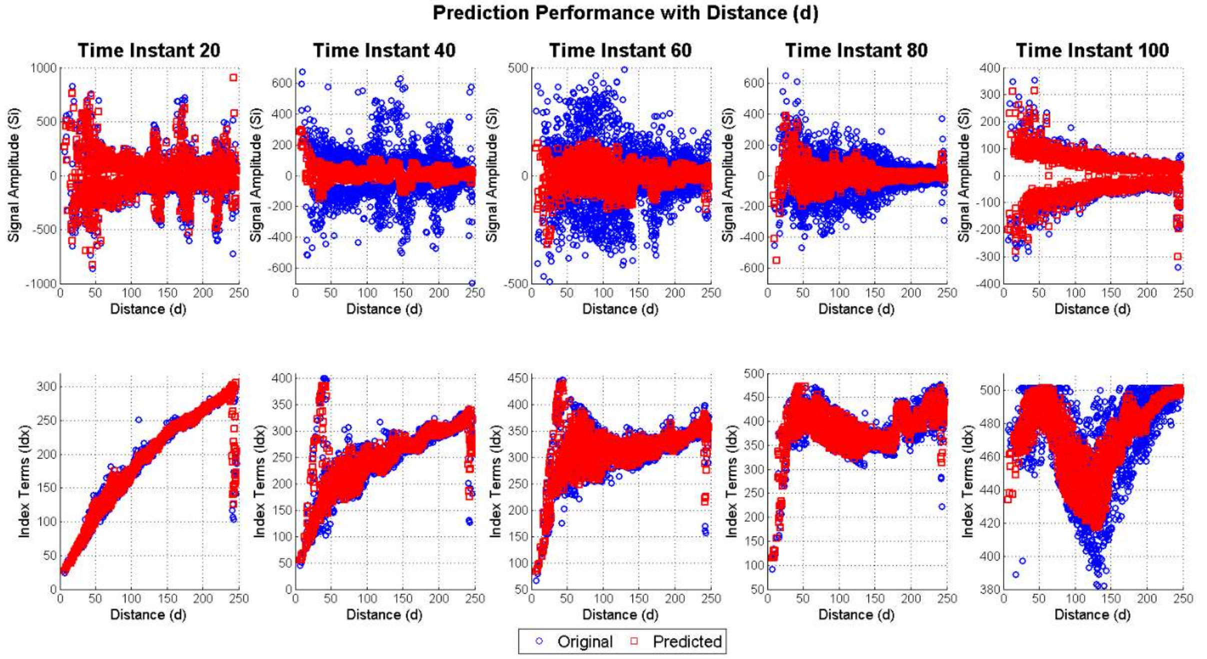


Figure 10: Ground truth vs. predicted variation in Si and Idx as a function of distance d using Gaussian process surrogates.

Most variants of Gaussian Process surrogates with quadratic and linear basis are found to have a high predictive accuracy particularly with ARD covariance structures, although it is more computationally expensive as compared in Table 2. The squared exponential kernel produces inferior results compared to the Matern 3/2 and Matern 5/2 versions with both ARD and the basic kernels. The best accuracy has been obtained using the ARD Matern 3/2 kernel using linear basis on the training dataset in the initial screening. Validation of the prediction performance on the 2000 testing datasets and moreover on a sample by sample seismogram prediction using 1D Pearson correlation coefficient, instead of average 2D correlation coefficient has been shown in the subsequent sections, for the final choice of best GP model structure.

The SVM with polynomial and RBF kernel, GLM and/or kernelized shrinkage regressions like Lasso, Ridge, elastic nets and moderate size neural networks have produced a poor predictive performance, particularly most of them fail to partition between the positive and negative pressures in the scatter diagrams as a function of the predictors. The increased computation time in Lasso and elastic net is due to the cross-validation based automatic selection of hyper-parameter λ by finding a balance between penalizing prediction error and the model coefficients.

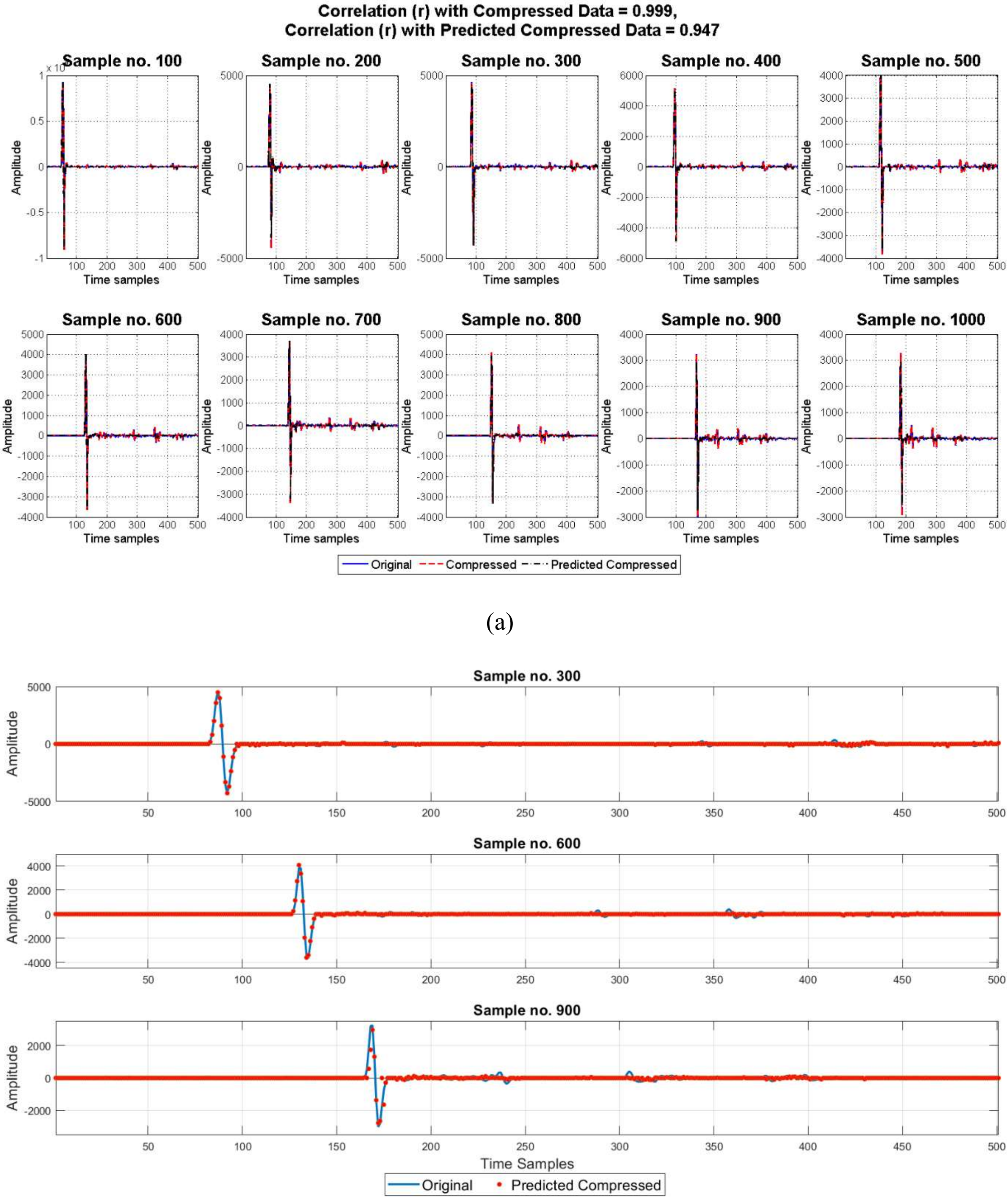
As described before, the decision tree method can produce high accuracy particularly with lesser value of N_{parent} thus producing large size of the tree and hence larger model size. In general, within the ensemble methods, boosting trees produced better results than the bagging tree methods with the same number of weak learners. With 1000 weak tree learners, the prediction accuracy reaches around $R \approx 0.6$ with the bagging method, while the sizes of the learned models become greater than few GBs and hence not investigated further with higher number of ensemble learners. In each case of the tabulated results using various family of regression models which can be found the supplementary material, the best predictive accuracy and the associated tuned parameters has been highlighted as bold italics entries for reconstructed accuracy R_{Recon}.

From these exhaustive comparisons, it is found that the GP with linear basis function and ARD Matern 3/2 kernel outperform all the rest of the model families to predict an accurate seismogram on the central receiver and hence has been chosen for further analysis. However, although the accuracy of ARD family of kernels produce best results, they can be computationally expensive during the training process and also depending on the number of data-points in the training set for the covariance estimation. Apart from the GP models, the decision tree with lower number of parents and in the family of ensemble learning methods, bagging with higher number of weak learners also perform well but these models have a larger storage requirement. Therefore, as a compromise between the achievable accuracy, training time, and storage requirement, the GP model with ARD Matern kernel is found to be the best choice for this regression problem in seismology. However, it is important to note that the regression models map each location in two compressed domains which are indeed smooth, as can be seen from Figure 9-Figure 10. However, using the predicted samples in the compressed domain each seismogram is reconstructed using the decompression steps that may not finally make the whole event co-ordinate to seismic trace mapping to be smooth enough. A slight decrease in the index term may not also yield smooth reconstructed seismograms which makes the final reconstructed correlation coefficient (R_{Recon}) to have a lower value for most of the learning algorithms compared to the Gaussian process regression models.

Also, from the results with increasing number of layers and number of hidden nodes in both the feedforward and cascaded neural networks, the compressed domain accuracies are found to be fair (provided in the supplementary material). However, they take more time during the training process. Training of very large networks (>100 nodes in 3 layers) has not been attempted since they need more time during the training process and since the alternative models are already giving better accuracy within similar training time-frame. Also, according to the "*no free lunch*" theorem, for different statistical learning scenario, there is no consensus that one single class of models whether neural network or Gaussian process would consistently outperform other family of learners and the best recommendations are to try a pool of models amongst which a class of models wins for a specific application. This fact is even more prominent in the context of difficult regression problems, as discussed in (Lattimore & Hutter 2013; Wolpert 2002; Goutte 1997; Domingos 2012), to achieve high enough accuracy compared to the well-researched classification problems where NNs are shown to outperform in contemporary research.

Previously in Figure 7, the variation of the two compressed components of the original seismic signal – *Si* and *Idx* have been shown. Here, the predictions of the machine learning algorithms have also been shown on these two components as a function of 4 covariates $\{x, y, z, d\}$ in Figure 9 and Figure

10 respectively showing variation with depth (z) and distance (d) and the two lateral directions in the supplementary material. The ground truth of compressed domain data is presented as the circles and the corresponding predictions are shown as square boxes. It is evident that the Gaussian process meta-model is capable of learning the split predictions for the positive and negative pressure values and the corresponding complicated shape for *Idx* as found with respect to {z, d} in Figure 9 and Figure 10 respectively.



(a)

(b)

Figure 11: (a) Original, compressed and predicted reconstructed seismograms using Gaussian process surrogates. (b) Zoomed seismic traces with predicted samples. The amplitude is in Pascal and 500 samples represent 2 sec of seismic data.

It is understandable from the schematic diagram in Figure 8 that the regression models are trained to predict only the two components in the compressed domain and the respective accuracies have been reported as R_{Si} and R_{Idx} in the tables. The predictions in compressed domain are then used to decompress and obtain the sparse predicted seismograms as shown in Figure 11(b), corresponding to the reconstructed accuracies R_{Recon} in the tables. Figure 11(a) compares the original (elastic PDE simulations), compressed and predicted compressed seismograms which show minimal loss of information with the compressed representation ($R = 0.999$) and predicted compressed ($R = 0.947$) data. In particular, the arrival times of the seismograms are accurately predicted which carry most of the useful information in a source location inversion process (Tarantola 2005), as also evident from the 10 representative samples from the 2000 data-points along with a zoomed version of three seismograms in Figure 11(b). The predictions of the surrogate regression models are most commonly visualized in the form of cross plots as the deviation around the optimal least square line which are shown in the two compressed as well as the reconstructed domains in Figure 12, along with the achieved predictive accuracies mentioned in the titles of the subplots.

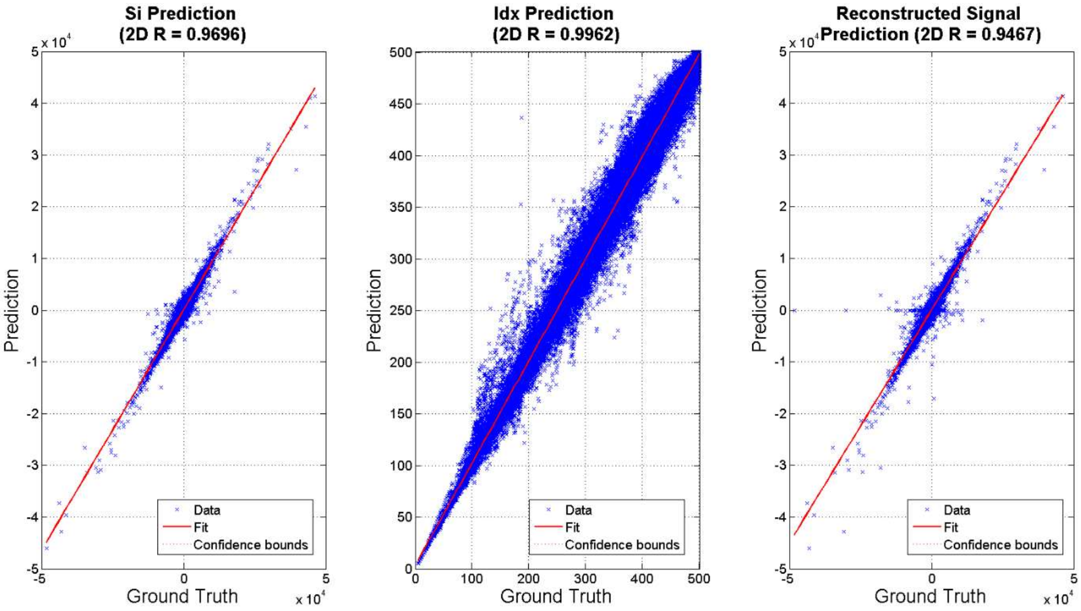


Figure 12: Cross-plots between the ground truth and the best GP predictions in compressed and reconstructed domains.

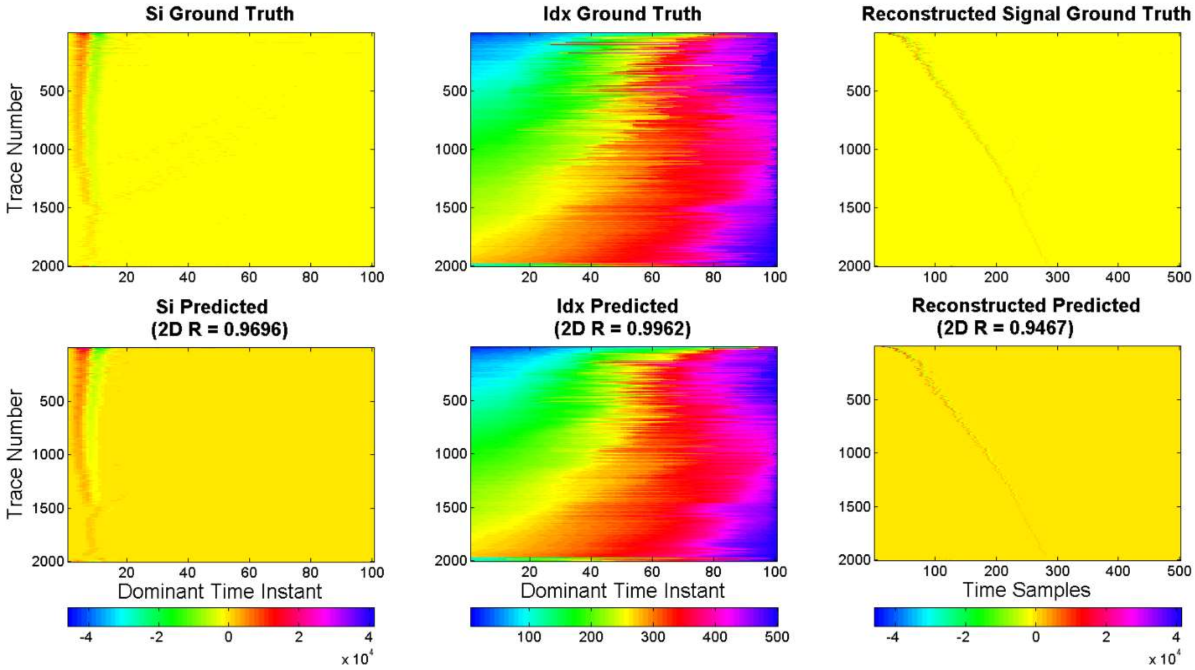


Figure 13: 2-D visualization of ground truth and the GP predictions in compressed and reconstructed domains.

The predictions can also be visualized in the form of a 2D image where the rows indicate different trace numbers, sorted against the distance from the central receiver and the columns denote the dominant time instant in the compressed domain and time samples in the reconstructed seismograms as shown in Figure 13, using the best found regression model i.e. Gaussian process with linear basis and ARD Matern 3/2 kernel. On the reconstructed image of the seismograms in the last column of Figure 13, the power law behaviour in the dominant amplitude as a function of increasing distance from the central receiver is evident. The first column of Figure 13 clearly shows an oscillatory i.e. first increase in pressure, followed by a pressure drop. The corresponding dominant time instants in Figure 13 (also incorporating the first arrival time) has more complexity due the heterogeneity, complex ray paths and P/S-wave mode conversion of the seismic waves.

5.2. Learning Curve Analysis and Computation Speed Up Using the GP Regression Models

The best found trained model structures reported in the previous subsection are now evaluated for their predictive accuracy on both the 2000 training and 2000 testing LH samples shown in Figure 2 with a random choice of the subset of samples and multiple shuffles, commonly known as the learning curve analysis. The learning curve analysis shows the accuracy vs. robustness trade-off for a trained model and helps in selecting the minimum number of training samples required to get a fair predictive accuracy. Here the number of samples are gradually increased as shown in Figure 14 while the average prediction accuracy of 100 Monte Carlo shuffles are carried out to select a subset of samples from each of the 2000 training and 2000 testing LH samples. Both the training and testing datasets seem to converge after 1000 samples. A smaller gap between training and testing dataset is reflected in the 2D correlation coefficient of the data and indicates an improved performance over the other methods. Figure 14 also shows that the ARD Matern 3/2 kernel produces slightly better accuracy with the linear basis over the quadratic basis on both training and testing dataset. However, the results seem to converge closely using the ARD Matern 5/2 kernel. Depending on the heterogeneity of the velocity model, the learning curves on the training and testing datasets may vary, in other studies.

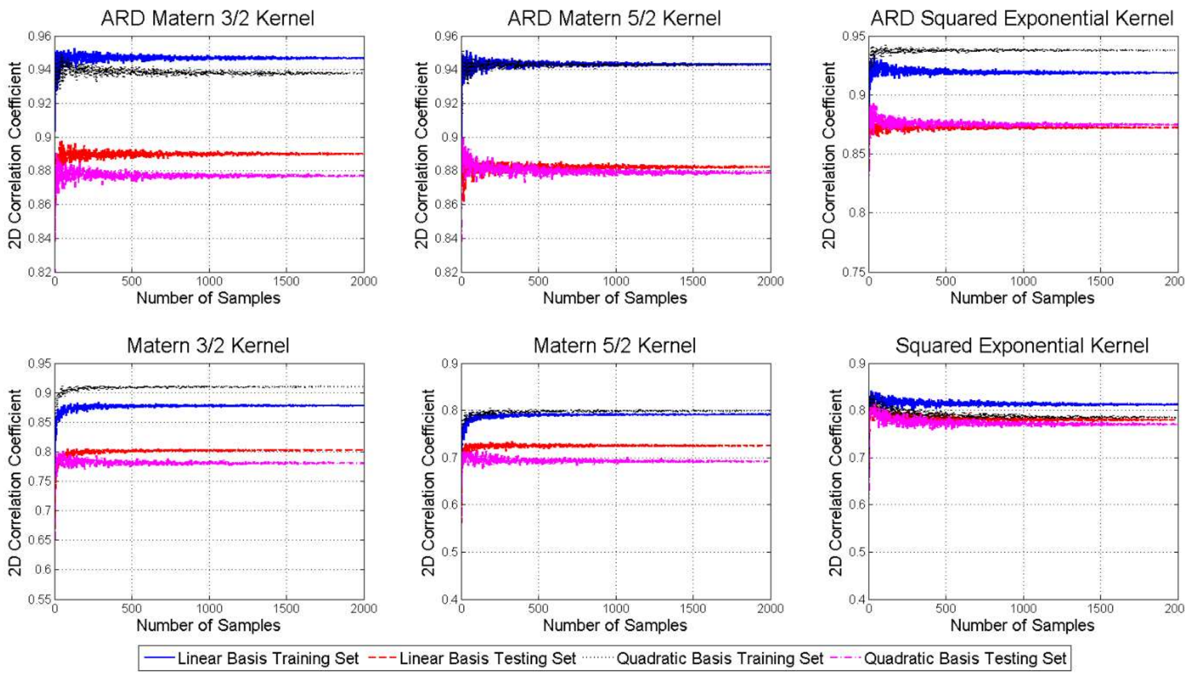


Figure 14: Learning curves of the GP models using the training and testing data using 100 Monte Carlo shuffles of the datasets.

In Figure 15, we show the run time distributions of the 2000 synthetic seismogram simulations using various trained GP proxy meta-models. It is evident from Figure 15 that the time required to generate a single seismogram is less than a sec using the surrogate model as compared to the GPU based full elastic wave equation solving, as shown in earlier sections. However, there is an intermediate computationally expensive step to train the surrogate meta-models as shown in Table 2, which gradually increases with the number of samples for Gaussian process regression and particularly with ARD family of kernels, although they provide more accurate results than other methods. This has been investigated in the next subsection.

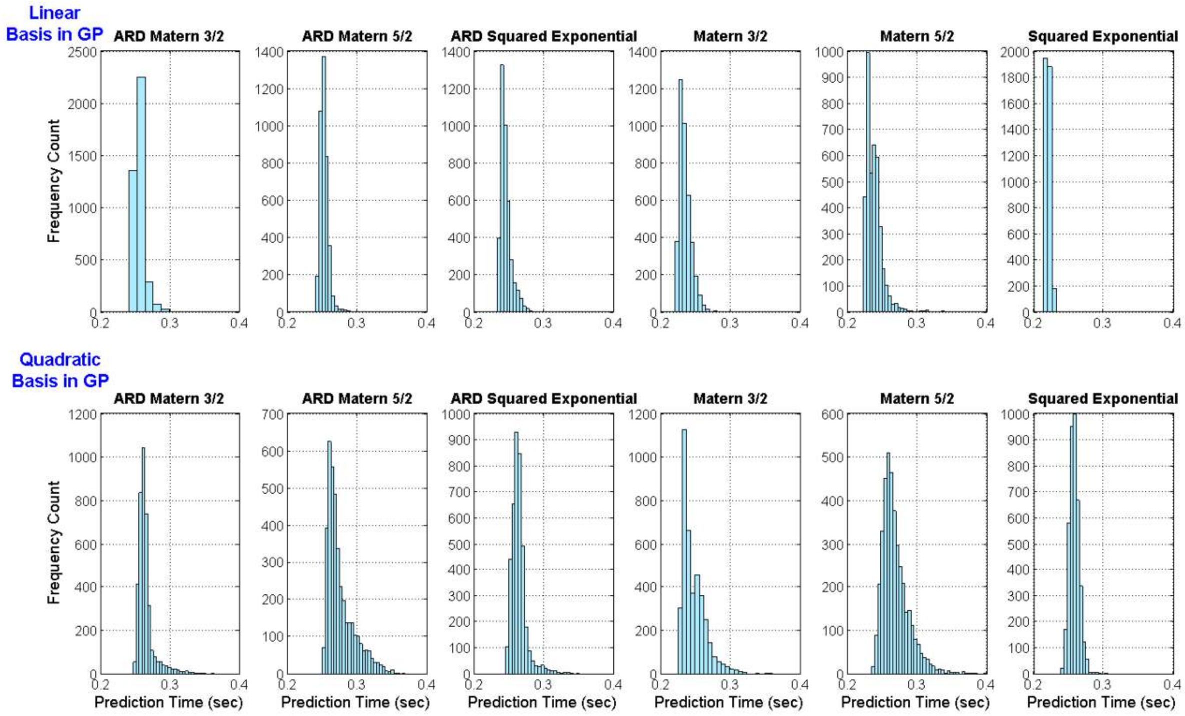


Figure 15: Histogram of a single seismogram generation time at the central receiver using the trained surrogate model and 4000 data-points.

5.3. Effect of Training Sample Size on the Regression Modelling

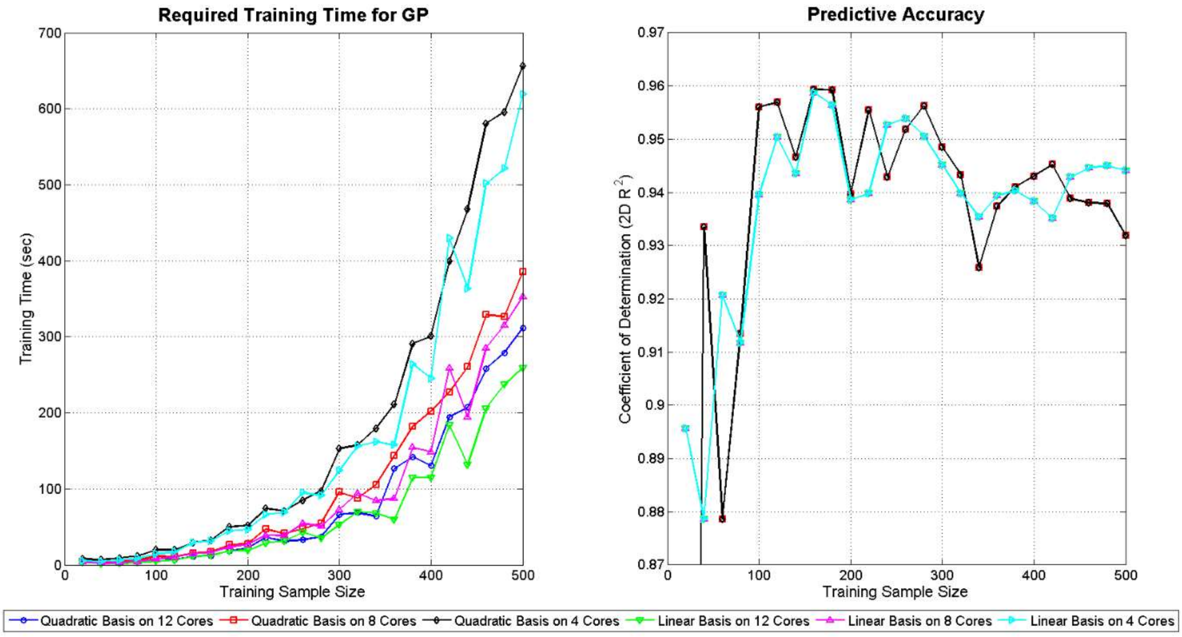


Figure 16: Training time requirements and predictive accuracy with increased training samples size using ARD Matern 3/2 kernel with quadratic and linear basis function. Saving in training is clear using 12 parallel cores (over 4 and 8 parallel cores) and with linear basis function (over the quadratic one). Predictive accuracies are comparable between linear and quadratic basis with increased sample size.

Here we explore the training time and predictive performances with increasing training sample size for the best GP models which are found to outperform the other family of regression models. The GP ARD Matern 3/2 model has been trained in parallel using 4, 8 and 12 core CPU via the parallel for (*parfor*) loops in Matlab on the 100 dominant time instants to learn the compressed seismograms. In order to show the scalability of the training process, the number of samples have been increased from 20 to 500 in steps of 20 samples and the required training times along with the corresponding predictive accuracies have also been shown in Figure 16, as a function of the sample number (N_{sample}). It is evident that there is a steep increase in training time for over N_{sample}>300 and even more with the quadratic basis in the GP ARD Matern 3/2 kernel while the predictive accuracy fluctuates around 2D R^2 = 0.93-0.95. Also the small fluctuations on the training accuracy can be observed in Figure 16, as the random samples come from different positions of the heterogeneous velocity model and thus introducing certain set of samples may slightly reduce the overall accuracy but varies within a small range and finally settles down. This is more evident in a finer resolution in the learning curve analysis in Figure 14 on the trained model using 2000 training/testing data with 100 Monte Carlo shuffles of increasing subset of samples, as presented in the previous subsection.

5.4. Prediction Enhancement by Using Smoothing Filter

Since the GP models predict the dominant 100 time-instants and the corresponding signal values independently, as a function of event spatial locations, without explicitly considering the temporal correlation of the seismogram time series, sometimes the predicted signals may not be smooth in time. Especially in some cases, rapid positive and negative pressure fluctuation may be encountered i.e. with reverse polarity with a small movement of the event locations as shown in Figure 9 and Figure 10. Therefore, a moving average (MA) smoothing filter is applied on the GP predicted seismic data while varying its span size from 1-10 in order to select the best filter settings for ensuring the smoothness of the seismogram time series. Since the dominant signal values are predicted separately in the learning process, without considering the temporal information between two consecutive time samples, the predicted seismic signals may not vary smoothly in few cases. Here, the purpose of the smoothing filter is thus to introduce some amount of inertia against rapid fluctuation of the signals against changing polarity within a short span of time. It is apparent that a larger span of the smoothing filter introduces a delay in the seismograms and hence the performance degrades gradually, as evident from Figure 17 showing a sweep over MA smoothing filter window size from 1-10 consecutive time samples. In the smoothed versions of the seismograms, both the ARD Matern 3/2 and ARD Matern 5/2 kernels with either linear or quadratic basis win over the other combinations, particularly on the test-set. Figure 17 also suggests that these GP settings with a MA smoothing filter of span size of 3 samples are capable of producing accurate predictions both in the training and testing set with a 2D *R*>0.91. Representative examples of the predicted reconstructed and smoothed seismograms are shown in Figure 18 and Figure 19 from the training and testing set respectively.

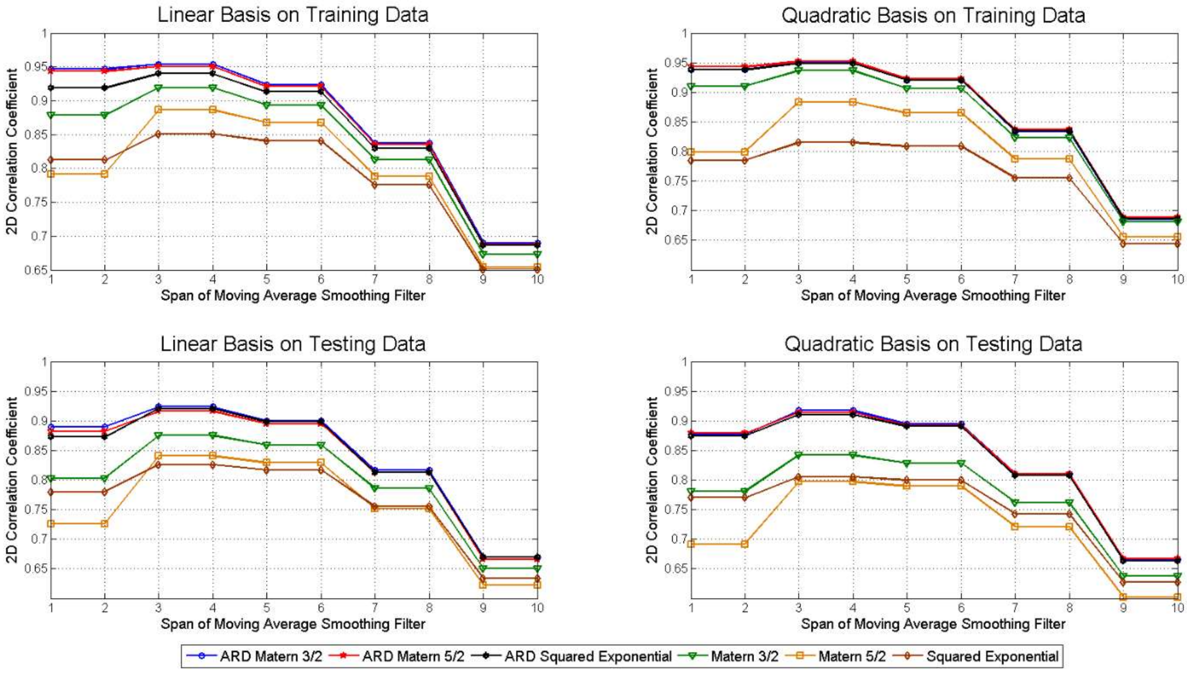


Figure 17: Performance of smoothed seismogram predictions using GP quadratic basis and six kernels. Both ARD Matern 3/2 and 5/2 kernels for both linear and quadratic basis give good prediction performance on both the training and testing set.

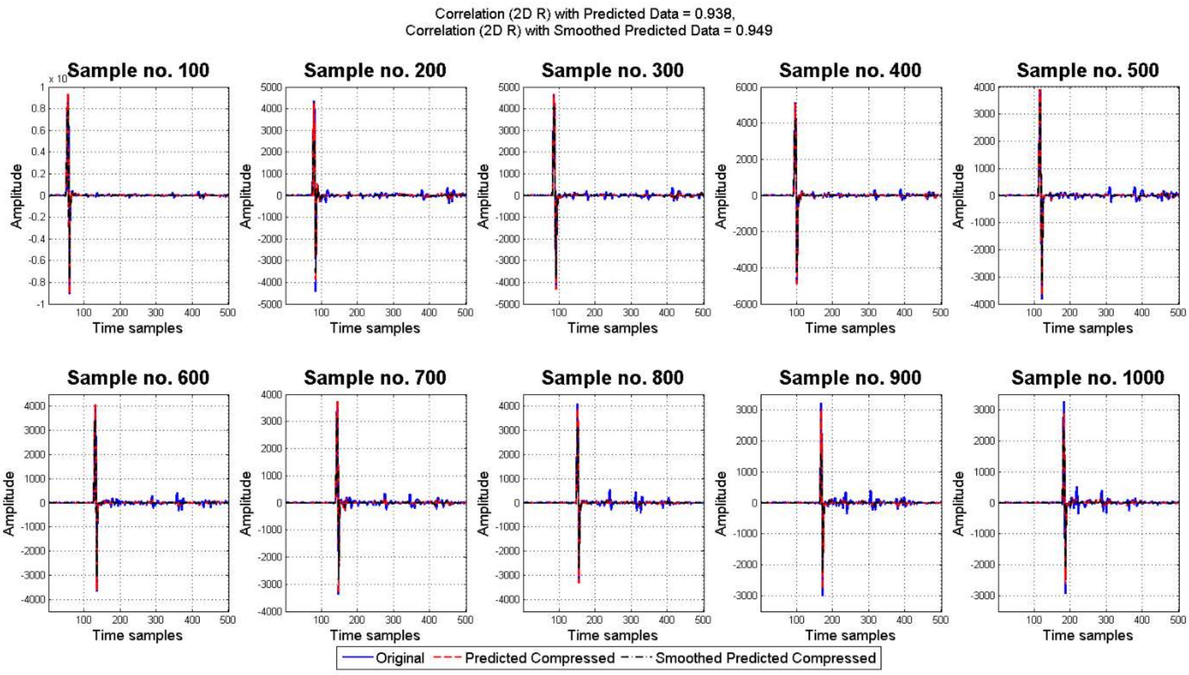


Figure 18: Original simulated, GP predicted and smoothed reconstructed seismograms in the training dataset.

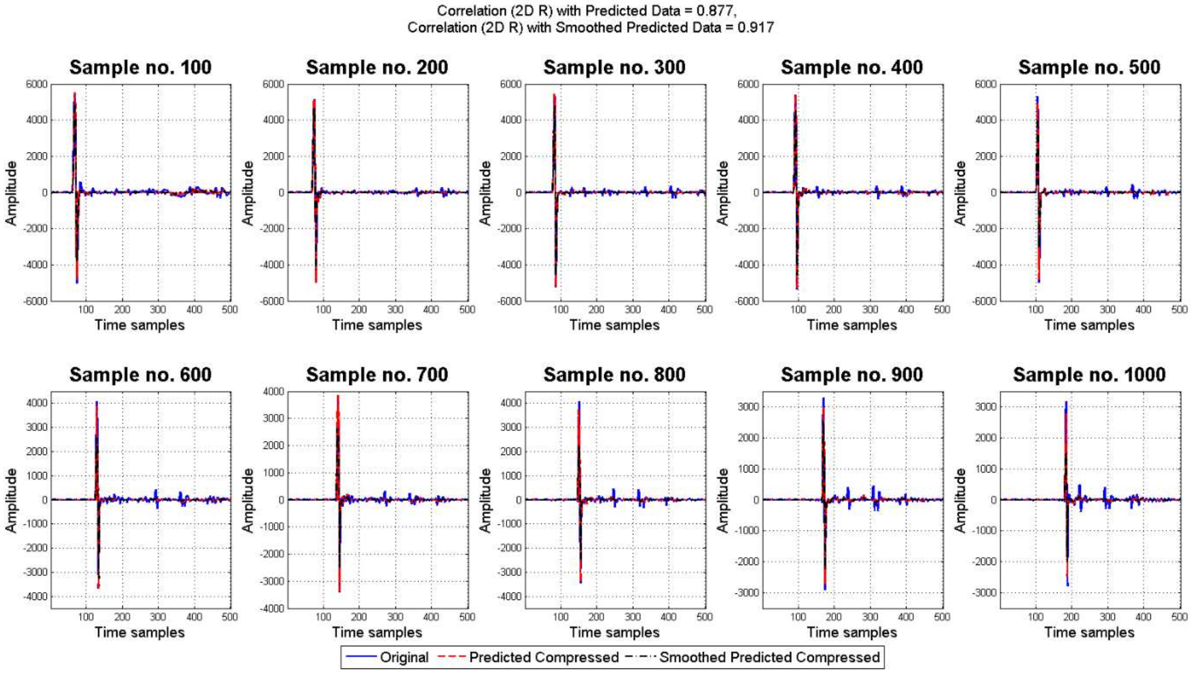


Figure 19: Original simulated, GP predicted and smoothed reconstructed seismograms in the testing dataset.

It is observed from Figure 17 that the curves attain their maxima at MA filter span = 3 samples for all the cases and also, the ARD Matern 3/2 and ARD Matern 5/2 kernels with both linear and quadratic basis functions give comparable average accuracy on the training and testing dataset, in terms of 2D correlation coefficient. Individual accuracies of each seismogram may be different using these two kernels. Hence the 1D Pearson correlation coefficient based on the individual smoothed seismograms have also been calculated after the initial model screening, on both the training and testing dataset using the ARD Matern 3/2 and 5/2 kernels with both the linear and quadratic basis functions. Although the overall predictive accuracy (in terms of 2D correlation) on the training and testing dataset are similar for the ARD Matern 3/2 and 5/2 kernels with both linear and quadratic basis, as shown in Figure 14, the number of relatively poor predictions or outliers present in the predictions are actually different. Therefore, from the first stage screening from the pool of machine learning algorithms using 2D correlation coefficient on all time instants and samples, following the schematic in Figure 8, we carry out a further second stage selection of the best algorithm that yields minimum number of outliers in its predictions. We define a predicted data-point as outlier if the 1D Pearson correlation coefficient between a particular simulated and the corresponding predicted smoothed seismogram becomes negative i.e. $R_{1D}<0$. The goal here is to minimise such extreme predictions, although most of the predicted seismograms show fairly high accuracy.

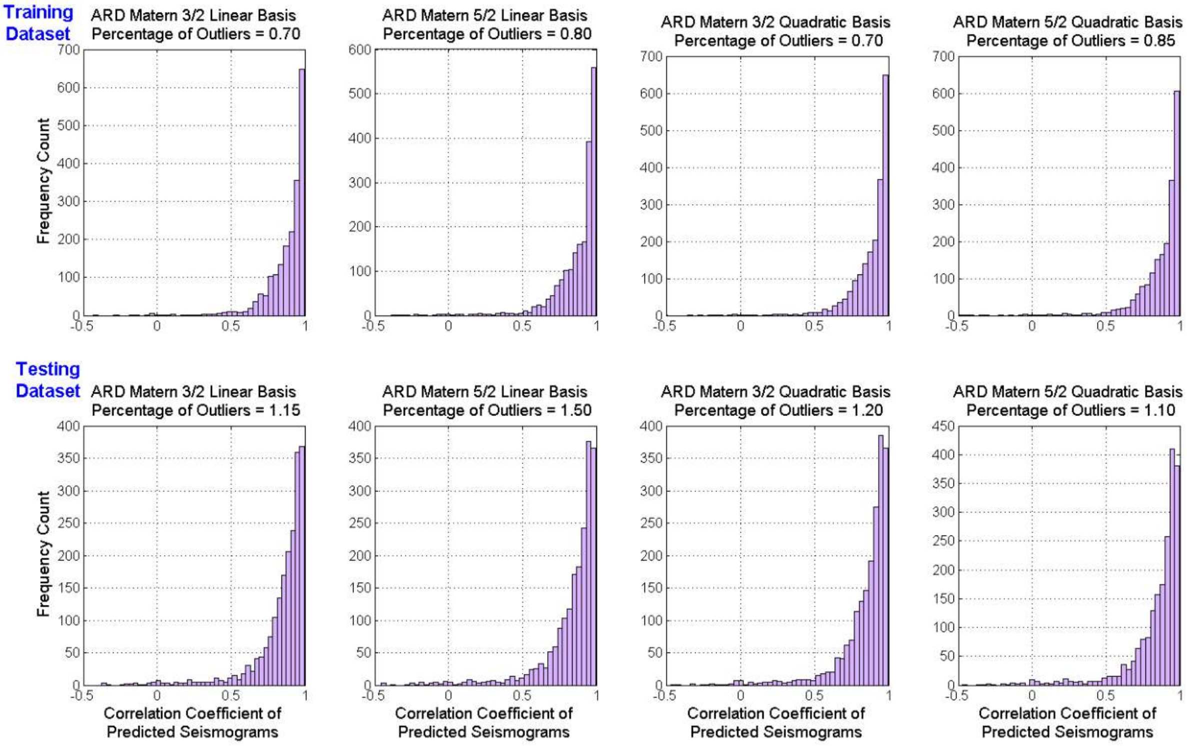


Figure 20: Histograms of 1D correlation coefficients between ground-truth vs. smoothed predicted seismograms for the central receiver using the training and testing dataset. The percentage outliers in different models are mentioned in the titles.

Figure 20 shows that in all the cases for both training and testing dataset, most of the samples give a good predictive accuracy, as revealed from the peaks near $R \approx 1$. In all the cases there is a small decaying left tail, indicating a drop in the predictive accuracy. Counting these outliers as a fraction of the total sample size below a fixed threshold R_{1D}<0 can identify the best surrogate meta-model structure with minimum number of outliers. It is apparent from Figure 20 that the ARD Matern 3/2 kernel with both linear and quadratic basis functions produce the same lowest number of outliers (0.7%), on the training dataset. The same kernel with linear basis produces 0.05% less outliers having negative correlation, over that with the quadratic basis and hence chosen in remainder of the paper for further analysis. The presence of few predicted outliers can also be viewed from the cross-plots in Figure 21, after applying the tuned smoothing filter on the predicted seismograms. In general the ARD Matern 3/2 kernel with linear basis gives a trade-off between high average predictive accuracy (as revealed from the cross-plots on training/testing dataset in Figure 21) and minimum number of outliers (represented by the left tail of the histograms in Figure 20).

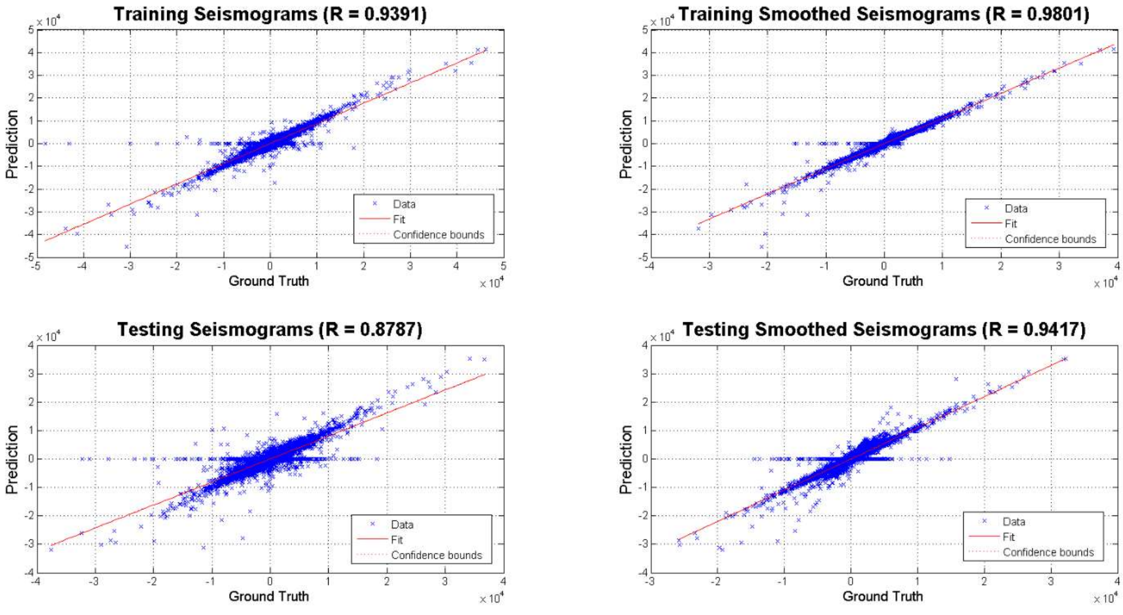


Figure 21: Cross-plots of the training and testing dataset with and without smoothing using the GP quadratic basis Matern 3/2 kernel. Data is reshaped in 1D array to calculate the optimum least square line and correlation coefficient R.

6. Prediction Performance on All the 23 receivers

6.1. Selection of the Best Regression Model for All the 23 Receivers

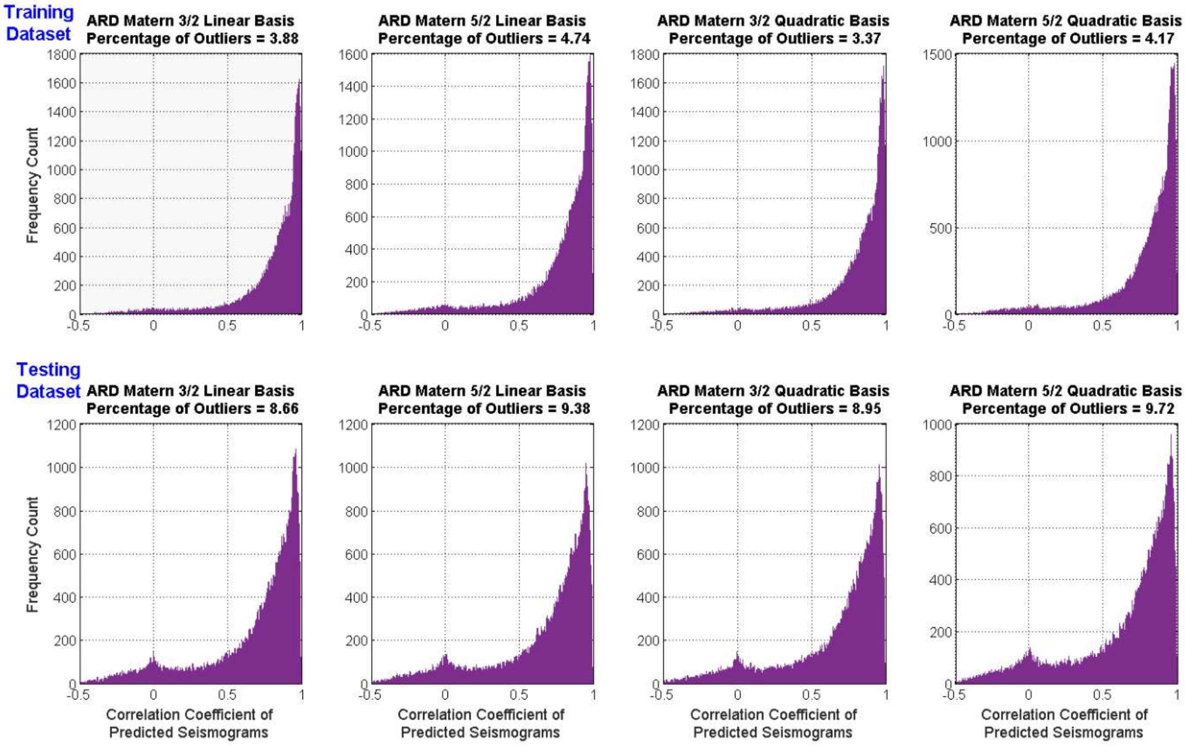


Figure 22: Histograms of 1D correlation coefficients of each predicted seismograms for all the 23 receivers using the training and testing dataset. The percentage outliers in different models are mentioned in the titles of subplots.

The exploration reported in the previous section shows fine-tuning of the proxy models on the central receiver when the event can roam around within the subsurface volume underneath. This mapping for the event location to receiver's response is not necessarily similar for different receivers (in Figure 3), due to the heterogeneity of the velocity model. We now verify the performance of the proxy or surrogate meta-model on all 23 receivers using the best set of models that produced good predictive accuracy on the central receiver i.e. Gaussian process regression with linear and quadratic basis having ARD Matern 3/2 and ARD Matern 5/2 kernels. Amongst these four class of models, the smoothing filter-tuning and outlier detection has been carried out in the same way for the multiple receivers' case, as shown in the earlier sections. The other choices of smoothing filter window size apart from 3 samples are found to be worse, as also shown before for the central receiver. Here, individual receiver responses are analysed separately instead of comparing aggregated predictions on the 23 receivers. With the MA smoothing filter having a window size of 3 samples, the 1D Pearson's correlation coefficient (R_{1D}) between the predicted vs. original seismogram on training and testing data for all 23 receivers are shown in Figure 22, using the best four GP proxy meta-models. It is also evident from Figure 22 that although the ARD Matern 3/2 kernel with quadratic basis in GP produces 0.51% fewer outliers with R_{1D}<0 on the training dataset, on the testing dataset the same kernel with linear basis produces 0.29% fewer outliers and is hence chosen for the rest of the analysis and the likelihood calculation.

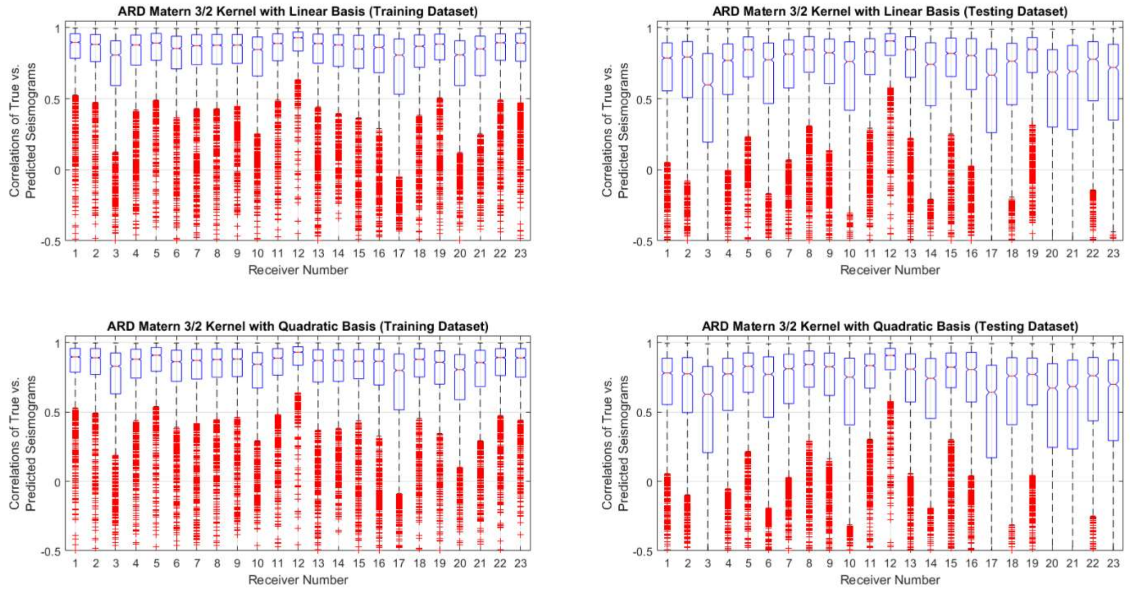


Figure 23: Box-plots of the receiver-wise correlation coefficient between the original vs. predicted seismograms on the training and testing dataset using ARD Matern 3/2 kernel with linear (top panel) and quadratic basis (bottom panel). The red crosses indicate outliers in the prediction on individual receivers.

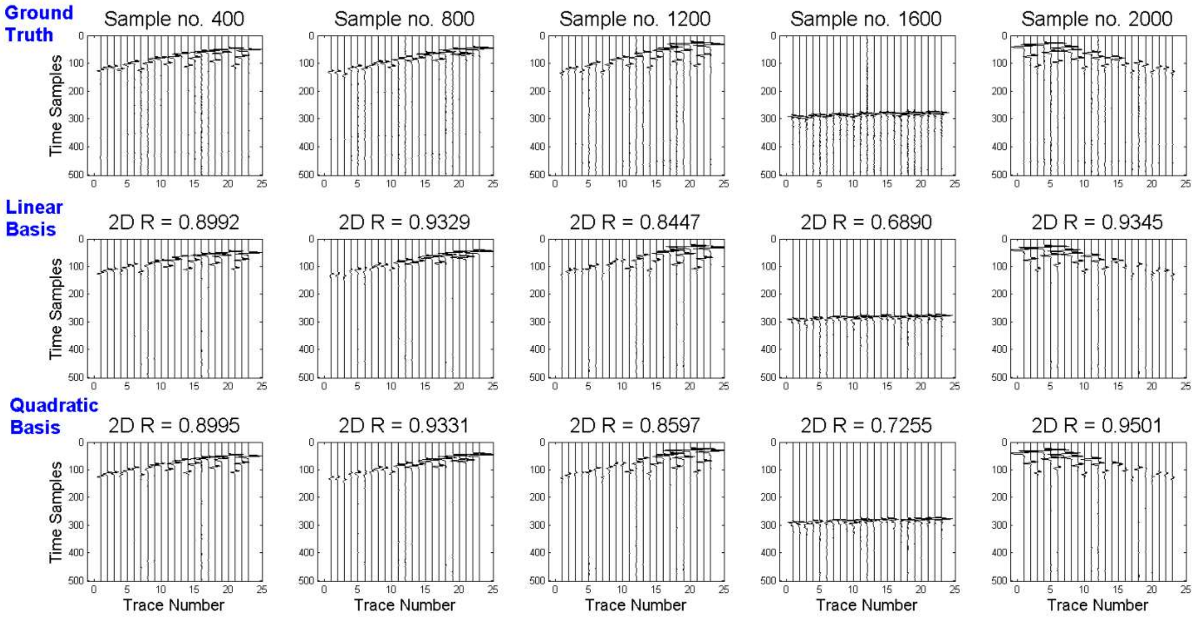


Figure 24: Predicted seismogram wiggle plots using the ARD Matern 3/2 kernel on the training dataset (top) ground truth, (middle) with linear basis, (bottom) with quadratic basis. Corresponding R_{2D} on 23 receivers are reported in the subplot titles.

Receiver-wise prediction performances are shown in Figure 23 on the 2000 training and 2000 testing datasets using the top 2 surrogate meta-models using ARD Matern 3/2 kernel with linear and quadratic basis functions. It is apparent from Figure 23 that the central receiver (R-12) produces the best prediction accuracy amongst all the 23 receivers. Also, amongst these 23 receivers in Figure 3, R-5, R-8, R-15, R-19 are found to have the next best responses whereas R-3, R-10, R-17, R-21 contain relatively more outliers. The presence of outliers in certain channels does not necessarily represent unusable predictions, as the 1D correlation coefficient R_{1D} essentially compares the full morphology of the spiky seismograms. In most cases, the arrival times and the polarity of first arrival of the seismic waves are predicted accurately, containing most of the useful information (Tarantola 2005). We have also shown 5 representative examples of true vs. predicted seismograms on all the 23 receivers from both the training and testing datasets in Figure 24 and Figure 25 respectively using the ARD Matern 3/2 kernel with linear/quadratic basis (in the two bottom rows), where the arrival times and morphology of the seismic response using the proxy meta-models are identified almost accurately with the original solutions of the expensive elastic PDE solver (represented in the top row).

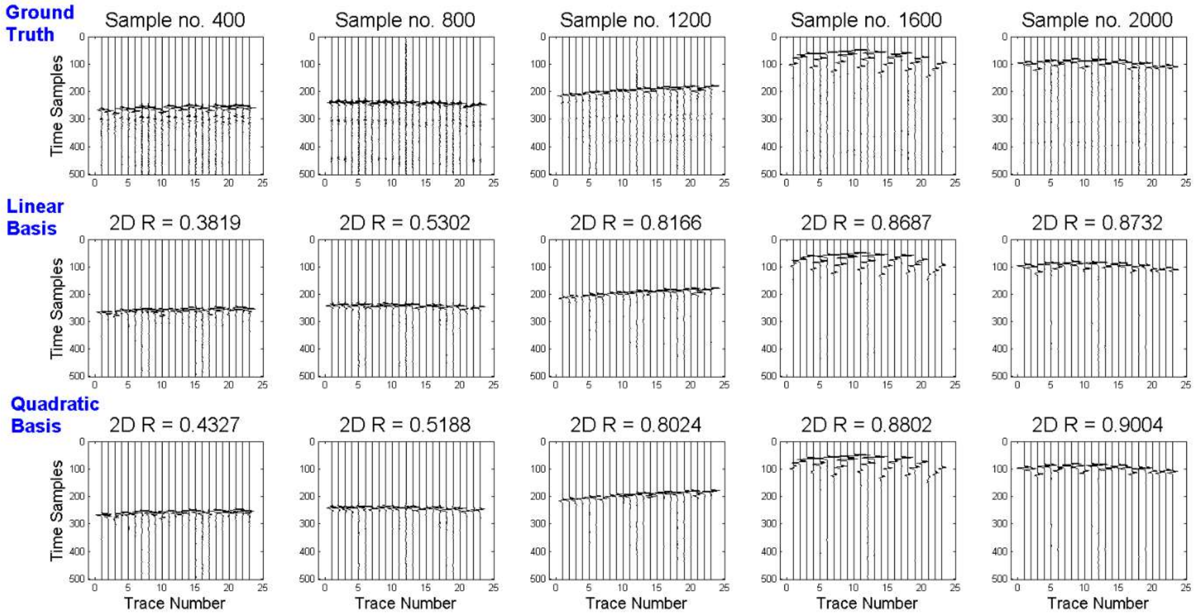


Figure 25: Predicted seismogram wiggle plots using the ARD Matern 3/2 kernel on the testing dataset (top) ground truth, (middle) with linear basis, (bottom) with quadratic basis. Corresponding R_{2D} on 23 receivers are reported in the subplot titles.

6.2. Comparison of the Computation Time for Surrogate Proxy Meta-Models vs. Solving the Full Elastic Wave Equation

In this subsection, we compare the run time saving due to the use of proxy or surrogate meta-models instead of the full elastic PDE solver for obtaining the seismic response at the 23 receivers at specified locations due to explosive microseismic events at random locations in the subsurface. As discussed in the introduction section, the purpose of surrogate meta-modelling is to reduce the computational time for fast generation of approximate template seismic events and hence facilitate a likelihood-based inversion approach where such fast noiseless template data generation is required in batches of thousands of speculative event locations.

Figure 26 shows that using the GP linear basis with both ARD Matern 3/2 and 5/2 kernel, the peak of the run time distribution is around 2 sec and for the quadratic kernels the peak run time is around 2.8 sec on a standard 4-core 64-bit Windows desktop PC with 16 GB memory and Intel I5, 3.3 GHz processor. Whereas for a single shot seismic simulation, the original elastic wave propagation on a 12-core Linux PC with K20 GPU card with 5.5 GB memory and 1.1 GHz processor, the peak run time is 1063 sec ≈ 17.7 min. Therefore, to simultaneously compute the seismic response at the 23 receivers, the surrogate regression meta-models produce a 531-fold acceleration using the linear kernel and 380-fold acceleration using the quadratic kernel. This speed up for the forward simulation when called from the likelihood function comes at the cost of initial simulation for training data generation and required training efforts of GP regression meta-models, but this is needed only once for a fixed velocity model. There is also a small inaccuracy incurred due to the compressed domain regression modelling in comparison with the original elastic wave simulation. In many real microseismic monitoring applications such approximate templates are sufficient for probabilistic event parameter estimation problems, since the measurements are often buried under significant amounts of noise, thus making the effect of such small modelling uncertainties due to the proxy negligible.

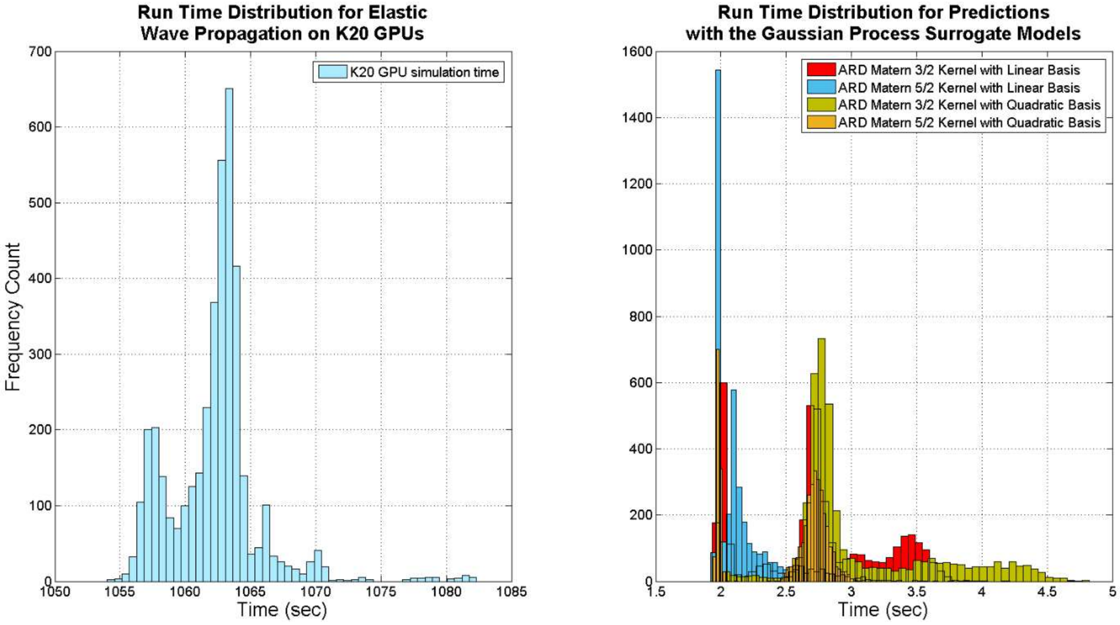


Figure 26: Comparison of run time distributions between elastic wave propagation on K20 GPU card and surrogate proxy meta-model predictions on CPU for all the 23 receivers.

7. Fast Computation of the Likelihood Function Using the Trained Surrogate Meta-Models

7.1. Formulation of the Likelihood Function for Detecting Microseismic Events

In this section, we use the best surrogate meta-model i.e. the GP with linear basis and ARD Matern 3/2 kernel, followed by a MA smoothing filter with a span-size of 3 samples, for fast computation of template seismic response in the likelihood computation. In many geophysical inverse problems, fast calculation of the likelihood is necessary in order to get the maximum likelihood (ML) or maximum a-posteriori (MAP) estimates or calculation of the evidence to enable model comparison. Representing the true noiseless template seismic response due to a microseismic event as $\hat{Y}$ and a measured noisy response as Y, the Gaussian likelihood function can be calculated as (17):

$$L = \frac{1}{\sqrt{(2\pi)^N |C|}} \exp\left[-\frac{1}{2}\left(Y-\hat{Y}\right)^T C^{-1}\left(Y-\hat{Y}\right)\right]. \tag{17}$$

Here, C is the covariance matrix of the noise on the measured data Y, and N is the number of observed data points. Often the likelihood is represented in log-scale for convenience in Bayesian analysis and is given by (18) considering either a full or only diagonal covariance matrix:

$$\begin{aligned}&\log L=-\frac{N}{2}\log\left(2\pi\right)-\frac{\log\left|C\right|}{2}-\frac{1}{2}\left(Y-\hat{Y}\right)^{T}C^{-1}\left(Y-\hat{Y}\right),\\&\log\left|C\right|=\begin{cases}N\log\left(\sigma^{2}\right) & \text{for diagonal covariance}\cdot\\2\times\sum_{i}\log\left(diag\left(C_{i}^{Cholesky}\right)\right) & \text{for full covariance}\end{cases}\end{aligned}\tag{18}$$

In the log-likelihood calculation involving the full covariance matrix, the Cholesky decomposition is commonly used for numerical stability and increased speed, whereas for diagonal covariance the log determinant of covariance (log|C|) can be easily computed using the common variance (σ^2) as in (18).

For calculating the log-likelihood in (18), given some speculative microseismic event locations {*x*, *y*, *z*} the noiseless predicted seismic data can be obtained using the trained proxy meta-model in (19) following the steps shown in the schematic diagram Figure 8:

$$Y=F_{proxy}\left(X\right)=F_{proxy}\left(x,y,z,d\right).\tag{19}$$

The covariance matrix *C* required in the likelihood (17) can be calculated from the measured noisy data (*Y*) using (20), considering a diagonal covariance or uncorrelated noise for the sake of simplicity:

$$\begin{aligned}&C=\mathbb{E}\left[\left(Y-\overline{Y}\right)^{T}\left(Y-\overline{Y}\right)\right],\quad \overline{Y}=\mathbb{E}\left[Y\right],\\&Y=\hat{Y}+\mathcal{N}\left(0,C\right),\quad C=\sigma^{2}I.\end{aligned}\tag{20}$$

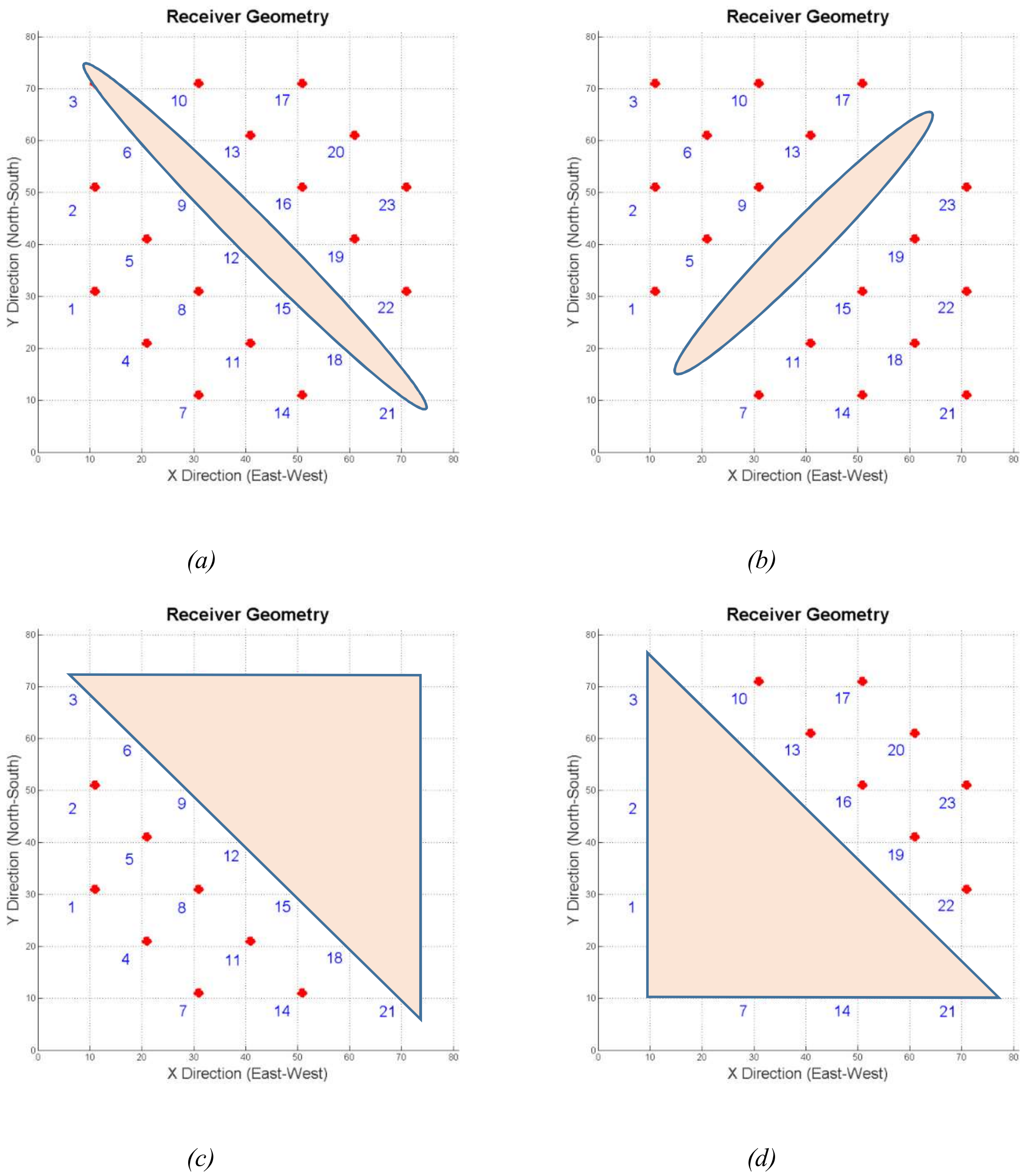
Here, σ^2 is the common variance of the data, reshaped as 1D vector *Y* in the multi-receiver case, with the assumption of no correlation amongst them and $\overline{Y}$ represent the mean of the measured data, while $\mathbb{E}$ being the mathematical expectation operator. Here in (20), the noise has been considered to have a Gaussian distribution with zero mean and a specified variance σ^2, however any expert choice of the noise covariance can also be incorporated in the likelihood function (17).

Next we calculate and visualize the likelihood as an inverse problem for the microseismic event locations (Tarantola 2005)(Aster et al. 2011), in different cases viz. using only the central receiver's data, seismograms along the principal and anti-diagonals, in the upper/lower triangular parts or using all the 23 receivers. For computation of the likelihood, template seismic responses corresponding to single microseismic events at random positions are calculated first using the fast proxy/surrogate meta-models and independent white Gaussian noise (wGn) of two different standard deviations σ = {100, 250} which are added on the noiseless seismic data to generate some realistic corrupted dataset. The noise free data is assumed to be generated due to a microseismic event at the grid point (31, 25, 158), as a representative example for the log-likelihood calculation. The signal to noise ratio (SNR) has been calculated on the single/multiple receivers using the ratio of average energy calculated through the sum of squared signal amplitudes and represented in the decibel scale as in (21):

$$SNR = 10\log_{10}\left(\sum_i A_{signal,i}^2 \Big/ \sum_i A_{noise,i}^2\right) \text{dB}. \tag{21}$$

It is understandable that in a relatively less noisy or high signal to noise ratio (SNR) case, the likelihood function will be manifested as a narrow delta function in the event parameter space which may be harder to detect. In the case of higher noise or in other words low SNR levels, the likelihood function gets softened which may help navigating towards the maximum likelihood regions by standard optimization or sampling algorithms like Markov Chain Monte Carlo (MCMC) etc. Here we focus on obtaining the ML estimates of the event parameters for a single microseismic source, with a specified noise variance by gradually increasing the number of receivers.

The most likely event positions are visualized using the scatter diagrams for different receiver combinations. For this purpose, here we use the top 90 percentile of all the log-likelihood values out of the 4000 uniformly distributed LH samples. Out of these 4000 samples one of them is the ground truth voxel which is expected to have the highest likelihood value. In order to verify this, we calculate the maximum likelihood estimate of the microseismic event location and the norm difference of the event positions from the ground truth location (31, 25, 158), with two different SNR levels using various combinations of receiver positions which can be found in the supplementary material. In the next sub-section, the joint distribution of the event parameters in the 2D scatter plots are shown using the top 90 percentile of the likelihood values where the higher likelihood values are represented by bubbles with a darker shade. In all the cases the legends show the log-likelihood values.



(a) *(b)*

(c) *(d)*

Figure 27: Receiver subset selection for the likelihood calculation (a) principal diagonal, (b) anti-diagonal, (c) upper-triangular region, (d) lower triangular region.

7.2. Different Receiver Subset Selection and Its Effect on the Likelihood Function

Here we explore 6 different cases of the receiver subset out of the 23, in order to calculate the likelihood using the LH samples. The positions of the central receiver (R-12) and all the 23 receivers have been shown in Figure 3. Out of the 23 receivers, we now select a subset of receivers along the principal-diagonal (total 7) and anti-diagonals (total 5) as shown in the top row of Figure 27 (a and b). In order to show the effect of the heterogeneity in the velocity model of Figure 1, here the 23 receivers are divided in the upper and lower triangular parts as shown in the bottom row of Figure 27 (c and d).

The corresponding noiseless and noisy seismograms in these different geometries are shown in Figure 28. To better understand the spiky or narrow spatially localised nature of the likelihood function for multiple receivers, the scatter diagrams of the randomly sampled likelihoods are provided in Figure 29.

Moreover, there are small errors incurred between the true PDE simulated vs. the proxy predicted seismic data in all the receivers which get combined within the likelihood calculation. The propagation of these small modelling uncertainties due to the use of proxies or surrogates for different receivers on the final parameter estimates in the inversion process may be explored more systematically in a future work. As per the previous reports of geophysical inversion e.g. in (Tarantola 2005), incorporating more receivers' data should make the estimates more accurate and the non-vanishing high likely regions should ideally shrink towards a smaller region within the volume under scanning which is also observed here. However, addition of higher noise level decreases the SNR and consequently softens the likelihood in all the cases. The gross natures of the likelihood are not drastically altered for the same receiver subsets but different noise levels.

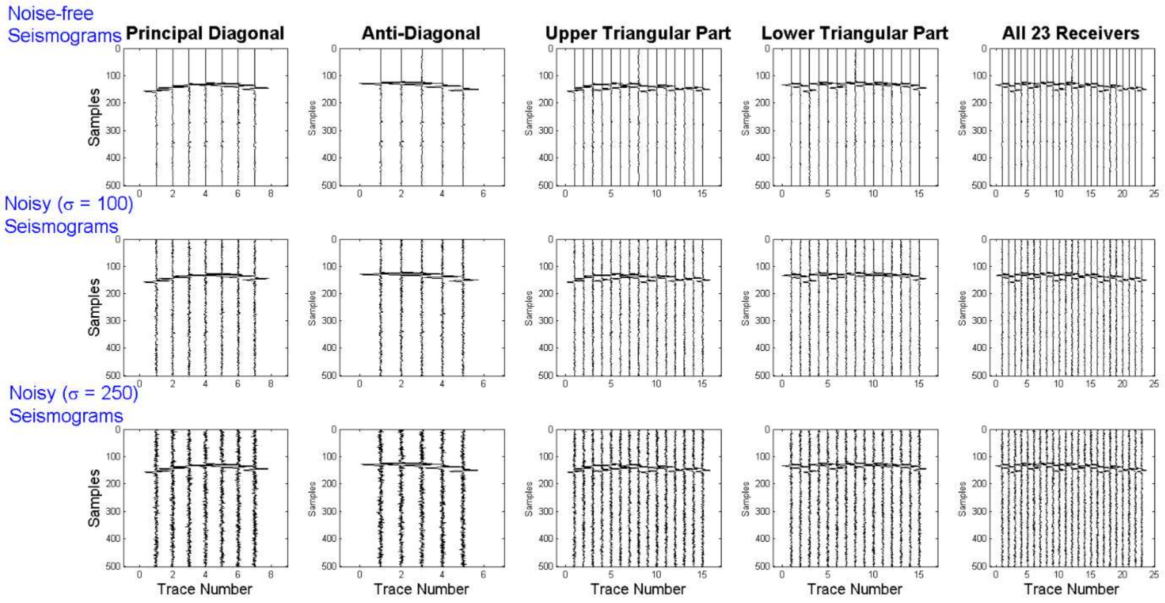


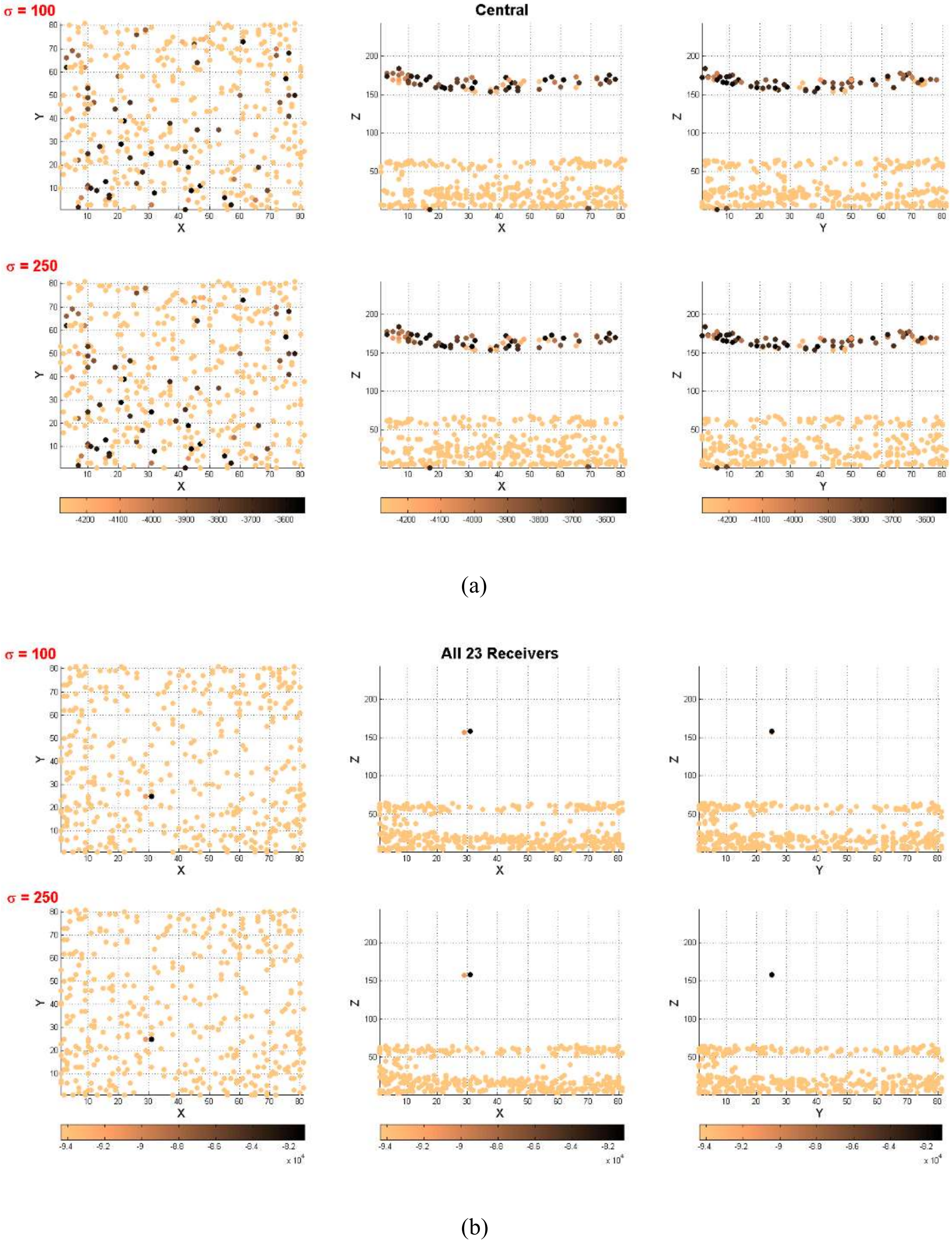
Figure 28: Seismograms used for calculating the likelihoods and maximum likelihood estimate of event position (top) noiseless (middle) wGn with σ = 100, (bottom) wGn with σ = 250.

It has been found that some of the regions have more high likelihood values where the data was originally generated from whereas in certain cases some other regions have more non-vanishing samples with high likelihood values. This is essentially a problem of resolution vs. smoothness trade-off of the likelihood, whereas an accurate detection should locate towards the highest likelihood voxel and its neighbouring regions. This may be an effect of the heterogeneity of the velocity model that neighbouring samples not always yield a smooth variation of the likelihood values. Alternatively, the likelihood values could have been calculated using some derived features of the seismic traces like the arrival times as shown in (Tarantola 2005) or some other feature like the polarity of first arrival etc. which may be pursued in a future research. The choice of the feature in such cases is crucial to yield a smooth variation of the likelihood values in the neighbouring voxels whereas here we focus on the raw seismic data based likelihood calculation only.

The non-vanishing most likely regions can also be summarized in terms of maximum likelihood point estimates, by bulk likelihood calculation using the LH samples. A systematic exploration would need a Bayesian sampling of the posterior distribution using a chosen likelihood function involving the raw seismic data itself or using some derived features (like arrival time or polarity) where the samples will gather more towards the mode of the posterior probability distribution which may be pursued in a future study. Here we explore the maximum likelihood values for convenience, corresponding to the bulk likelihood calculation at random locations. The detection error norm ($\|e\|$) for the event positions has been calculated as the Euclidean distance between the ground truth (x_o, y_o, z_o) and estimated ($\widehat{x_o}, \widehat{y_o}, \widehat{z_o}$) locations via maximum likelihood using (22):

$$\|e\| = \sqrt{\left(x_o - \widehat{x_o}\right)^2 + \left(y_o - \widehat{y_o}\right)^2 + \left(z_o - \widehat{z_o}\right)^2} \ . \tag{22}$$

The noise levels, the corresponding maximum likelihood estimate based detected voxels and detection error norm are reported in the supplementary material. It is evident that all the different cases can essentially capable of identifying the ground truth voxel with highest likelihood value. Depending on the noise level and receiver arrangements, some other voxels may also spuriously show high likelihood values which is explored next.



(a)

(b)

Figure 29: Scatter-diagram of top 10 percentile of likelihood values for two different noise levels and receiver combinations (a) central, (b) all 23 receivers. Similar plots with other receiver combinations are shown in the supplementary material.

As discussed before, the top 90 percentile of the likelihood values can also be visualised as scatter diagrams between the event location parameters as shown in Figure 29 for two different receiver combinations. It is evident that using just the central receiver (Figure 29a), although the ground truth voxel is revealed in darker shade, there are other spurious voxels producing similar but slightly less likelihood values. The number of these spurious voxels reduces in the case of 5 and 7 receivers along the diagonals as shown in the supplementary material. The ground truth voxel becomes more prominent when more receivers – 15 (upper or lower triangular as in the supplementary material) or 23 (all of them in Figure 29b) are used in the inversion. Use of more receivers shows the presence of many lower likelihood values at a different depth instead of the ground truth which is expected to shrink with a higher threshold on the likelihood values. In all the scatter diagrams in Figure 29, the colour/shade of the data points are proportional to its log-likelihood values shown in the legend.

The ML estimates of the event location parameters have been reported in the supplementary material, using voxel by voxel batch likelihood evaluation with all the 4000 LH samples. Here, the estimated location parameters ($\widehat{x_o}, \widehat{y_o}, \widehat{z_o}$) are obtained as the voxel returning the maximum log-likelihood value is found to be accurate in all the receiver combinations. A more systematic way could be to maximize the likelihood function using an optimizer or using Bayesian sampling methods with accelerated likelihood calculation using the trained surrogate or proxy meta-model. The purpose of the present work is to make the likelihood calculation faster and independent of the data under consideration, as here the proxy directly predicts the raw observables i.e. template seismic patterns and is different from the likelihood training approach in the BAMBI algorithm in (Graff et al. 2013; Graff et al. 2012), that needs retraining the surrogate meta-model when the dataset and consequently the nature of the likelihood changes.

A closer look at the scatter plots of the 2D joint distributions reveal that the high likelihood values change rapidly with small variation in the event location, particularly with less number of receivers e.g. only 1 receiver (Figure 29a). Incorporating more receivers reduces such variations as shown in the supplementary material using 15 for the upper and lower triangular parts to all 23 receivers (Figure 29b). Even though in a binned histogram, it may show more number of non-vanishing higher likelihood areas, the peak of the likelihood may lie in a different location i.e. the ground truth voxel for generating the data.

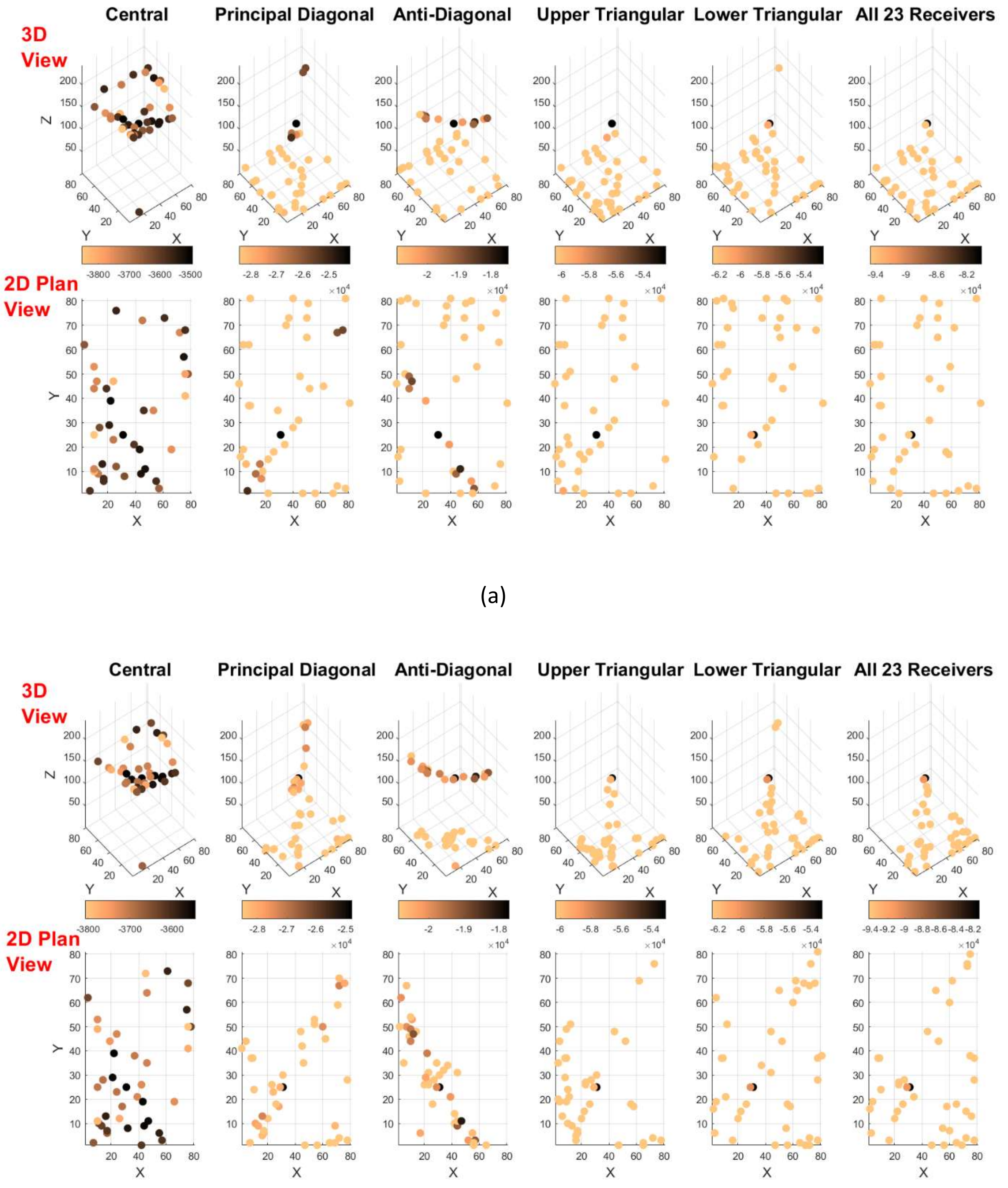
It is also important to note that in all of the above likelihood scatter diagrams, many islanded regions can be identified rendering such an event detection essentially a multi-modal inference problem. The adopted LH samples drawn throughout the 3D volume of the velocity model smoothly interpolates the true noiseless seismic responses in the forward problem but may be insufficient to accurately localise the events using bulk-scale likelihood calculation at these prospective locations while using the noisy seismic data in the likelihood calculation. Hence the Bayesian analysis techniques via MCMC or nested sampling family of algorithms may be useful here with a suitable choice of likelihood function by utilising the proposed method for fast data independent proxy meta-model to predict the observables, in order to get the localised event posterior distributions along with calculation of the marginal likelihood or evidence for comparing different models or carrying out hypothesis testing.

7.3. Maximum Likelihood Estimate Using Various Receiver Geometries in the Event Detection

In the previous sub-section, the top 90 percentile likelihood values have been shown in the scatter diagrams where some lower likelihood samples are gathered at a different depth compared to the ground truth. Now in this subsection, we show the maximum likelihood based most likely event location estimation, using a much higher threshold of top 99 percentile of all the likelihood values amongst the 4000 LH samples. This helps in graphically understanding the accurate localisation of the microseismic source using increasing number of receivers and different SNR levels, corresponding to the location estimates for the various receiver geometries. This also allows traditional '*dots in the box*' type visualization of the most-likely microseismic event locations (Kendall et al. 2011; Eisner et al. 2010), using the 6 different receiver sub-sets as explored in the earlier subsection.

Here, Figure 30 shows the event locations using the top 99 percentile of highest log-likelihood values using a noise standard deviation of $\sigma = 250$, whereas the less noisy case with $\sigma = 100$ is shown in the supplementary material. Both Figure 30 and its less noisy version in the supplementary material ($\sigma = 100$) compares the top 99 percentile of likelihood values between the true likelihood (from the elastic wave propagator) vs. the GP regression surrogate meta-model or proxy generated likelihoods. A closer look at these figures will reveal that the introduction of the external noise manifests many possible event locations with high likelihood values, particularly when the inversion is attempted using just the central receiver or multiple receivers across the principal diagonal (7 receivers) and anti-diagonals (5 receivers). With increasing number of receivers, the variation in the likelihood values for other possible locations gradually decreases and the true event location becomes quite prominent, as manifested in the form of a larger and darker bubble in the last 3 columns of Figure 30, using 15 and 23 receivers respectively. It is also evident that with more number of receivers, the true vs. proxy generated likelihood peaks are located at the same position, compared to that using less number of receivers. This shows employing 15 or 23 receivers, the maximum likelihood detection is invariant between the choice of expensive true likelihood vs. the cheap GP proxy-based likelihood.

During the proxy training it might seem that the near surface shallow sources introduce a bias due to their higher amplitude compared to the deep sources. In order to show that the proxy-based likelihoods are indeed unbiased, we have compared the true likelihood values vs. the proxy-based likelihood values and their difference in Figure 31. In the joint plane of depth vs. distance, the likelihood differences are found to be low compared to the original likelihood values and are almost uniform with variation in depth or distance from the central receiver. This indicates the efficacy of the proxy in generating fast likelihood values close to the original ones. Also, the likelihood surface is not smooth owing to the fact that the medium is heterogeneous, and the seismograms containing complex structures of both the P-wave and S-waves. The likelihood values from the full-physics simulation at the sampled 4000 locations vs. the likelihood obtained from the surrogate meta-model are compared in Figure 31, along with their differences. Our simulations show that there is very small difference between these two cases, due to the fact that the surrogate meta-model has learnt the data generation mechanism by the elastic wave propagator rather than the likelihood surface itself. Also, introduction of the surrogate does not increase the complexity of the likelihood structure and indeed retains its shape intact.



(a)

(b)

Figure 30: Most likely event locations using various receiver arrangement with added noise std σ = 250. Log-likelihood values are shown in the color-bars and the size of the data-points are proportional to the likelihood values: (a) True likelihood, (b) Proxy-based likelihood.

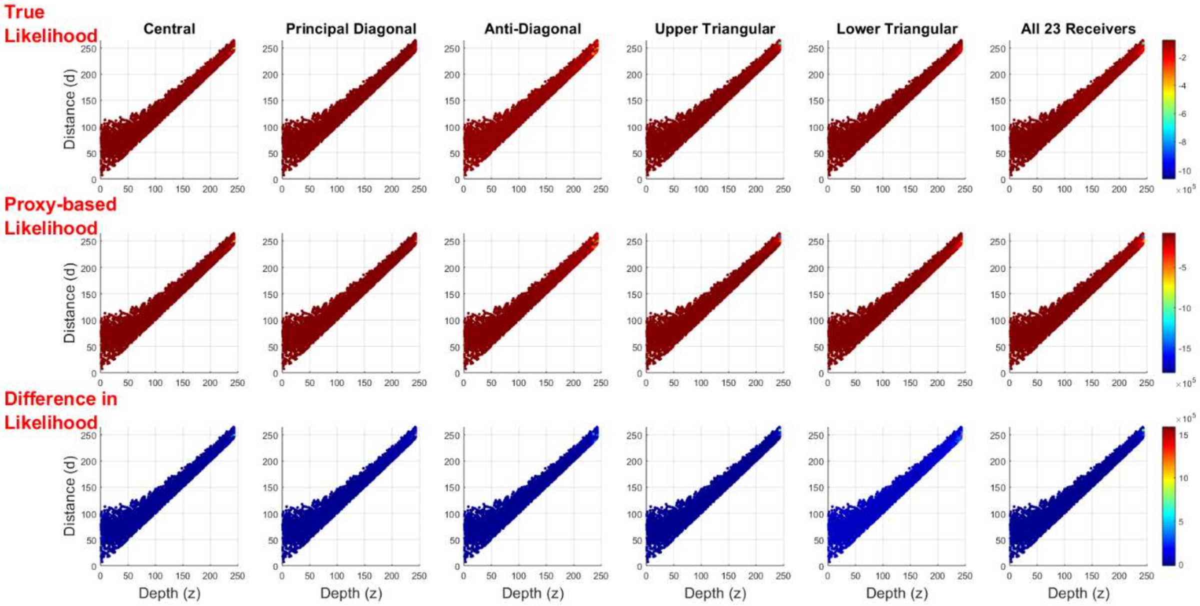


Figure 31: Comparison of the true likelihood vs. the proxy-based likelihood and their difference for noise std σ = 250. The legends represent the likelihood values or the difference.

8. Discussions

It is important to note here that the likelihood calculation has become relatively cheaper using the trained proxy meta-model, compared to the full elastic wave propagation solution, and therefore bulk calculation of voxel by voxel likelihood or a subset of LH or uniformly sampled voxel values may also help identifying the high likelihood regions for possible microseismic events. On larger velocity models or in higher dimensions, this approach of bulk likelihood calculation may be wasteful. Because using a suitable sampler may easily identify the highly likely event locations within fewer likelihood calls. Using the raw seismic data makes a relatively wilder variation of the likelihood values between neighbouring voxels for the microseismic source. Several derived features like arrival times, polarity of first arrival etc. can be used to calculate the likelihood instead as shown in (Tarantola 2005), which may produce a smoother likelihood function in the form of almost concentric circles (in 2D) or spheres (in 3D), for a single event. The trained proxy model can easily be used to derive any complex features out of the raw predicted signals and use them in the likelihood function which may be explored in a future research.

Moreover, in our likelihood formulation the ground truth signal (Y) has been generated from the elastic wave equation solver and then corrupted with specified noise level (σ) whereas the template seismic data ($\hat{Y}$) has been generated from the surrogate or proxy meta-model. Therefore, the likelihood contains the misfit due to both the measurement noise on the receivers as well as the inaccuracy due to the approximate seismic wave modelling with the proxy. We have shown through the above simulation results that even with both these two components of possible inaccuracy, a single microseismic event with known ground truth position can be reliably identified as the maximum likelihood point amongst 4000 randomly drawn source positions where the bulk likelihood calculation at possible source locations can be made extremely fast compared to the full elastic wave solution. It is rational that the variance of this likelihood analysis may be increased with the use of the surrogate/proxy model for fast template data generation apart from the specified measurement noise that goes in to the likelihood function. But the mode or the maximum likelihood point is unaffected by such an approximation due to the accurate surrogate model which is shown by the zero detection-error for the test cases using all the receiver combinations. A more elegant and accurate but massively computationally expensive solution is to calculate both the ground truth data (Y) and template data ($\hat{Y}$) in each likelihood evaluation by directly using the elastic PDE solution which is explored here in brief and as a proof of concept, on the sampled 4000 prospective locations which were used for training/testing of the surrogate meta-model.

In previous literature on microseismic monitoring, there are abundant use of physically simplified models instead of full-physics simulation with velocity model heterogeneity and elastic wave propagation. This is one of the viable solutions to reduce the computational cost compared to the proxy or surrogate meta-model based approach for fast likelihood calculation. However, for microseismic simulation, the mode conversion between P-wave and S-wave are predominant at the layer edges, even for explosive sources as described in this paper. Therefore, approximate methods like ray-tracing etc. that depends on separately calculating the P-wave and S-wave responses and then superimposing them may miss these aspects of the geophysical modelling. Rather we here took an alternative approach using the full elastic solution of the wave equation and then using the surrogate regression meta-models. Here, the proxy meta-models are trained to produce close approximations of the full elastic solution which is preferable than solving a reduced physics models for fast likelihood calculation.

This paper develops a methodology for learning wave propagation through heterogeneous medium. Given sufficient samples in the training process and from the convergence characteristics of the learning curve in Figure 14, it is apparent that the surrogate meta-model captures a close enough approximation of the true seismic wave, obtained through the numerical solution of elastic wave propagation. The velocity model we consider here is heterogenous compared to the layered ones and represented by voxelized grids with different values of density and P/S-wave velocity in each voxel. However, the vertical variations of the rock properties are larger than that in the lateral direction in our model. The heterogeneity along different directions of similar models can be seen in (Das et al. 2017). It is also worth noting that the surrogate meta-model is trained on a fixed velocity model. For other complex models, the same machine learning framework can be applied in principle but needs retraining using thousands of independently simulated seismograms which are dependent on the structure of the velocity model. In a more heterogenous case, the training process is likely to take more samples for the convergence of the learning curve, as shown in Figure 14. However, this paper aims at first developing the generic methodology and testing on different velocity models may be addressed in a future research.

9. Conclusion

Starting from a heterogeneous velocity model, we propose a technique to teach machine learning based surrogate regression meta-models to approximate elastic wave propagation solutions due to microseismic events which is computationally expensive even using state of the art GPU computing facilities. This allows calculation of thousands of batch evaluations of proxy-generated approximate template seismic responses with reduced physics modelling for rapid calculation of likelihood functions, for comparing with noisy dataset in a microseismic source inversion algorithm. The paper first develops a robust time domain compression method to reduce the number of observables in a sparse pressure wave-field generated by unit amplitude seismic events using a fixed heterogeneous velocity model. Then it compares 9 different families of surrogate regression models along with the details of their parameter tuning to obtain sufficient predictive accuracy on the learned seismogram patterns on multiple receivers. The machine learning algorithms essentially learn the mapping between the compressed domain sparse and spiky time series of the seismic waves as a function of event location parameters which can be decompressed next to get the full seismic waves with great saving of the computational cost compared to solving the full elastic PDEs with new event locations.

This paper also determines the achievable accuracy vs. the training time and storage requirement trade-offs using different flexible regression structures for synthetic template seismogram generation. The best results are achieved using the Gaussian process regression by fine tuning of the kernels and basis-functions, as it naturally incorporates a Bayesian regression framework instead of yielding only point estimates and hence provide superior performance as a smooth interpolator. Seismic data generation on 23 receivers using this proxy meta-model are found to be ~530 times faster than the GPU simulations for full elastic wave equation, at the cost of negligible reduction in quality of the signals, as revealed by the correlation analysis of the ground truth vs. predicted seismograms. However, the GP proxy meta-models in spite of its high predictive accuracy on smooth regression problems need more computational effort with growing sample size and number of receivers during the training period. For fast likelihood calculation, it is not intended to online train the proxy models but to train it only once as an offline process on a multi-core CPU, assuming the fact that in the real-fields the velocity model do not change over shorter span of time. However, with uncertain velocity model the seismic patterns, especially the arrival times, may be different, thus leading to inaccurate likelihood values which needs further investigation in future. A combined approach of incorporating seismic measurement noise and velocity model uncertainty together in the proxy models and hence in the likelihood function may also be investigated in future.

As discussed before, a similar proxy based fast multi-modal Bayesian inference technique has been previously proposed in the BAMBI algorithm (Graff et al. 2013; Graff et al. 2012) by directly learning complex likelihood functions which changes and need retraining for inference on different datasets. However, the present paper extends this concept by learning the raw observables instead i.e. the multi-receiver seismograms which does not need to be retrained if the data and consequently the likelihood values had changed. In addition, mapping of the useful information in sparse observables buried under few millions of data-points in the output i.e. the multi-receiver spiky time-series needed a robust compression method which this paper develops first, to frame it as a non-sparse regression problem. Here we also show the predictive accuracy vs. training time and storage requirements using 9 different family of regression models out of which Gaussian process families with ARD kernels outperform the rest. Future works may include extending the methodology for unknown number of microseismic events in the presence of background noise of different spatio-temporal characteristics and comparing different models using a Bayesian analysis with evidence calculation for hypothesis testing. It may also be worth exploring other compression methods e.g. wavelet compression, instead of the adopted time domain method, considering the full seismic wavefield rather than individual seismograms and test for the best regression model for this application. Also, modelling stress tensor components along with the event locations for non-explosive microseismic source mechanism is a challenging research topic and even more in geological models with higher complexity and uncertainty. Research in these directions are in progress and will be reported in our future works.

Acknowledgement

This work has been supported by the Shell Projects and Technology. The Wilkes high performance GPU computing service at the University of Cambridge has been used in this work.

Appendix

Additional analysis and high-resolution images for the simulation results are provided in the supplementary material.

References

Aster, R.C., Borchers, B. & Thurber, C.H., 2011. *Parameter Estimation and Inverse Problems*, Academic Press.

Auld, T. et al., 2007. Fast cosmological parameter estimation using neural networks. *Monthly Notices of the Royal Astronomical Society: Letters*, 376(1), pp.L11–L15.

Auld, T., Bridges, M. & Hobson, M., 2008. COSMONET: fast cosmological parameter estimation in non-flat models using neural networks. *Monthly Notices of the Royal Astronomical Society*, 387(4), pp.1575–1582.

Babaei, M., Alkhatib, A. & Pan, I., 2015. Robust optimization of subsurface flow using polynomial chaos and response surface surrogates. *Computational Geosciences*, 19(5), pp.979–998.

Babaei, M. & Pan, I., 2016. Performance comparison of several response surface surrogate models and ensemble methods for water injection optimization under uncertainty. *Computers & Geosciences*, 91, pp.19–32.

Babaei, M., Pan, I. & Alkhatib, A., 2015. Robust optimization of well location to enhance hysteretical trapping of CO2: Assessment of various uncertainty quantification methods and utilization of mixed response surface surrogates. *Water Resources Research*, 51(12), pp.9402–9424.

Barutçuouglu, Z. & Alpaydin, E., 2003. A comparison of model aggregation methods for regression. In *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003*. Springer, pp. 76–83.

Chapman, C., 2004. *Fundamentals of seismic wave propagation*, Cambridge University Press.

Collettini, C. & Barchi, M.R., 2002. A low-angle normal fault in the Umbria region (Central Italy): a mechanical model for the related microseismicity. *Tectonophysics*, 359(1), pp.97–115.

Das, S., Chen, X. & Hobson, M.P., 2017. Fast GPU-Based Seismogram Simulation from Microseismic Events in Marine Environments Using Heterogeneous Velocity Models. *IEEE Transactions on Computational Imaging*, 3(2), pp.316–329.

Dieterich, J.H., Richards-Dinger, K.B. & Kroll, K.A., 2015. Modeling Injection-Induced Seismicity with the Physics-Based Earthquake Simulator RSQSim. *Seismological Research Letters*, 86(4), pp.1102–1109.

Domingos, P., 2012. A few useful things to know about machine learning. *Communications of the ACM*, 55(10), pp.78–87.

Eaton, D.W. et al., 2014. Scaling relations and spectral characteristics of tensile microseisms: Evidence for opening/closing cracks during hydraulic fracturing. *Geophysical Journal International*, p.ggt498.

Eisner, L. et al., 2010. Beyond the dots in the box: Microseismicity-constrained fracture models for reservoir simulation. *The Leading Edge*, 29(3), pp.326–333.

Forrester, A., Sobester, A. & Keane, A., 2008. *Engineering design via surrogate modelling: a practical guide*, John Wiley & Sons.

Forrester, A.I. & Keane, A.J., 2009. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1), pp.50–79.

Forrester, A.I., Sóbester, A. & Keane, A.J., 2007. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 463(2088), pp.3251–3269.

Friedman, J., Hastie, T. & Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1), p.1.

Friedman, J., Hastie, T. & Tibshirani, R., 2001. *The elements of statistical learning*, Springer, Berlin.

Goodwin, N., 2015. Bridging the gap between deterministic and probabilistic uncertainty quantification using advanced proxy based methods. In *SPE Reservoir Simulation Symposium*.

Goutte, C., 1997. Note on free lunches and cross-validation. *Neural Computation*, 9(6), pp.1245–1249.

Graff, P. et al., 2012. BAMBI: blind accelerated multimodal Bayesian inference. *Monthly Notices of the Royal Astronomical Society*, 421(1), pp.169–180.

Graff, P. et al., 2013. Neural networks for astronomical data analysis and Bayesian inference. In *2013 IEEE 13th International Conference on Data Mining Workshops*. pp. 16–23.

Groos, J. & Ritter, J., 2009. Time domain classification and quantification of seismic noise in an urban environment. *Geophysical Journal International*, 179(2), pp.1213–1231.

Guo, P., McMechan, G.A. & Guan, H., 2016. Comparison of two viscoacoustic propagators for Q-compensated reverse time migration. *Geophysics*, 81(5), pp.S281–S297.

Hobson, M. et al., 2014. Machine-learning in astronomy. *Proceedings of the International Astronomical Union*, 10(S306), pp.279–287.

Holland, P.W. & Welsch, R.E., 1977. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9), pp.813–827.

Huang, T.-M., Kecman, V. & Kopriva, I., 2006. *Kernel based algorithms for mining huge data sets*, Springer.

Ieong, 2012. xnfx - High order predictor matrix for regression analysis. Available at: https://uk.mathworks.com/matlabcentral/fileexchange/39144-xnfx-high-order-predictor-matrix-for-regression-analysis.

Igel, H., 2016. *Computational Seismology: A Practical Introduction*, Oxford University Press.

Igel, H., Mora, P. & Riollet, B., 1995. Anisotropic wave propagation through finite-difference grids. *Geophysics*, 60(4), pp.1203–1216.

James, G. et al., 2013. *An introduction to statistical learning*, Springer.

Kalantari-Dahaghi, A., Mohaghegh, S. & Esmaili, S., 2015. Coupling numerical simulation and machine learning to model shale gas production at different time resolutions. *Journal of Natural Gas Science and Engineering*, 25, pp.380–392.

Kendall, M. et al., 2011. Microseismicity: Beyond dots in a box—Introduction. *Geophysics*, 76(6), pp.WC1–WC3.

Lattimore, T. & Hutter, M., 2013. No free lunch versus Occam's razor in supervised learning. In *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*. Springer, pp. 223–235.

Leet, L.D., 1949. Microseisms. *Scientific American*, 180, pp.42–45.

Levy, C., Jongmans, D. & Baillet, L., 2011. Analysis of seismic signals recorded on a prone-to-fall rock column (Vercors massif, French Alps). *Geophysical Journal International*, 186(1), pp.296–310.

MacKay, D.J., 1997. Gaussian processes-a replacement for supervised neural networks? *Lecture notes for a tutorial at NIPS 1997*.

Modesto, D. & de la Puente, J., 2016. Exploring a Priori Reduced Order Models for Fast Seismic Simulations. In *78th EAGE Conference and Exhibition 2016*.

Mohaghegh, S.D., 2006. Quantifying uncertainties associated with reservoir simulation studies using a surrogate reservoir model. In *SPE Annual Technical Conference and Exhibition*.

Mosegaard, K. & Tarantola, A., 2002. 16 Probabilistic approach to inverse problems. *International Geophysics*, 81, pp.237–265.

Mosegaard, K. & Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth*, 100(B7), pp.12431–12447.

Mu, D., Chen, P. & Wang, L., 2013a. Accelerating the discontinuous Galerkin method for seismic wave propagation simulations using multiple GPUs with CUDA and MPI. *Earthquake Science*, 26(6), pp.377–393.

Mu, D., Chen, P. & Wang, L., 2013b. Accelerating the discontinuous Galerkin method for seismic wave propagation simulations using the graphic processing unit (GPU)—single-GPU implementation. *Computers & Geosciences*, 51, pp.282–292.

Neal, R.M., 1996. *Bayesian learning for neural networks*, Springer Science & Business Media.

Pan, I. et al., 2014a. A multi-period injection strategy based optimisation approach using kriging meta-models for CO 2 storage technologies. *Energy Procedia*, 63, pp.3492–3499.

Pan, I. et al., 2014b. Artificial Neural Network based surrogate modelling for multi-objective optimisation of geological CO 2 storage operations. *Energy Procedia*, 63, pp.3483–3491.

Pan, I. & Das, S., 2015. Kriging based surrogate modeling for fractional order control of microgrids. *IEEE Transactions on Smart Grid*, 6(1), pp.36–44.

Pandey, D.S. et al., 2016. Artificial neural network based modelling approach for municipal solid waste gasification in a fluidized bed reactor. *Waste Management*, 58, pp.202–213.

Phadke, S., Bhardwaj, D. & Dey, S., 2000. An explicit predictor-corrector solver with application to seismic wave modelling. *Computers & Geosciences*, 26(9), pp.1053–1058.

Plumb, A.P. et al., 2005. Optimisation of the predictive ability of artificial neural network (ANN) models: a comparison of three ANN programs and four classes of training algorithm. *European Journal of Pharmaceutical Sciences*, 25(4), pp.395–405.

Rasmussen, C.E. & Williams, C.K., 2006. *Gaussian Processes for Machine Learning*, MIT Press.

Rodriguez, A. et al., 2006. A multiscale and metamodel simulation-based method for history matching. In *ECMOR X-10th European Conference on the Mathematics of Oil Recovery*.

Rogers, S. & Girolami, M., 2015. *A first course in machine learning*, CRC Press.

Rutledge, J.T., Phillips, W.S. & Schuessler, B.K., 1998. Reservoir characterization using oil-production-induced microseismicity, Clinton County, Kentucky. *Tectonophysics*, 289(1), pp.129–152.

Samui, P. & Sitharam, T., 2010. Site characterization model using artificial neural network and kriging. *International Journal of Geomechanics*, 10(5), pp.171–180.

Sitharam, T., Samui, P. & Anbazhagan, P., 2008. Spatial variability of rock depth in Bangalore using geostatistical, neural network and support vector machine models. *Geotechnical and Geological Engineering*, 26(5), pp.503–517.

Slotte, P.A. & Smorgrav, E., 2008. Response surface methodology approach for history matching and uncertainty assessment of reservoir simulation models. In *Europec/EAGE Conference and Exhibition*.

Street, J.O., Carroll, R.J. & Ruppert, D., 1988. A note on computing robust regression estimates via iteratively reweighted least squares. *The American Statistician*, 42(2), pp.152–154.

Tarantola, A., 2005. *Inverse problem theory and methods for model parameter estimation*, SIAM.

Tarantola, A. & Valette, B., 1982. Inverse problems= quest for information. *Journal of Geophysics*, 50(3), pp.150–170.

Treeby, B., Cox, B. & Jaros, J., 2012. k-Wave A MATLAB toolbox for the time domain simulation of acoustic wave fields User Manual.

Treeby, B.E. et al., 2014. Modelling elastic wave propagation using the k-wave matlab toolbox. In *Ultrasonics Symposium (IUS), 2014 IEEE International*. pp. 146–149.

Treeby, B.E. & Cox, B.T., 2010. k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of Biomedical Optics*, 15(2), pp.021314–021314.

Weglein, A.B. et al., 2009. Clarifying the underlying and fundamental meaning of the approximate linear inversion of seismic data. *Geophysics*, 74(6), pp.WCD1–WCD13.

Wilson, K., Durlofsky, L.J. & others, 2012. Computational optimization of shale resource development using reduced-physics surrogate models. In *SPE Western Regional Meeting*.

Wilson, K.C. & Durlofsky, L.J., 2013. Optimization of shale gas field development using direct search techniques and reduced-physics models. *Journal of Petroleum Science and Engineering*, 108, pp.304–315.

Wolpert, D.H., 2002. The supervised learning no-free-lunch theorems. In *Soft computing and industry*. Springer, pp. 25–42.

Wood, L.C., 1974. Seismic data compression methods. *Geophysics*, 39(4), pp.499–525.

Zou, H. & Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2), pp.301–320.

Zubarev, D.I., 2009. Pros and cons of applying proxy-models as a substitute for full reservoir simulations. In *SPE Annual Technical Conference and Exhibition*.