Research, Society and Development, v. 13, n. 5, e7213545797, 2024
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v13i5.45797

A Comparison of WaveNet and XGBoost for traditional direct wave propagation and seismic inversion using horizontal layer models

Uma comparação entre WaveNet e XGBoost para propagação direta de ondas tradicional e inversão sísmica usando modelos de camada horizontal

Una comparación de WaveNet y XGBoost para la propagación directa de ondas tradicional y la inversión sísmica utilizando modelos de capas horizontales

Received: 05/05/2024 | Revised: 05/15/2024 | Accepted: 05/16/2024 | Published: 05/19/2024

Juli Sergine Tavares Teixeira Saldanha
ORCID: https://orcid.org/0000-0002-5784-4445
Federal University of Rio Grande do Norte, Brazil
E-mail: julisergine7@gmail.com
Jaime Andres Collazos Gonzalez
ORCID: https://orcid.org/0009-0002-1539-8018
Federal University of Rio Grande do Norte, Brazil
E-mail: collazos.jaime.andres@gmail.com
Paulo Douglas Santos de Lima
ORCID: https://orcid.org/0000-0002-7353-536X
Federal University of Rio Grande do Norte, Brazil
E-mail: paulodouglaslima@gmail.com
Hugo Alexandre Dantas do Nascimento
ORCID: https://orcid.org/0000-0003-1690-1201
Goias Federal University, Brazil
E-mail: hadn@ufg.br
João Medeiros de Araújo
ORCID: https://orcid.org/0000-0001-8462-4280
Federal University of Rio Grande do Norte, Brazil
E-mail: joaomedeiros@fisica.ufrn.br

Abstract

The application of machine learning in geophysics has steeply increased in the last decade, with the quality of its results varying according to the type of seismic problem in focus and the employed computational method. Deep Learning methods are achieving impressive results in this area, but we note that there is still a lack of certainty on whether classical machine learning methods can provide similar results. In the present paper, the objective was to attempt to fill part of that gap, by comparing a well-known non-DL machine-learning method with a DL method for the direct wave propagation and the seismic inversion problems for 2D horizontally-layered models. Both methods are evaluated in different scenarios, but under similar conditions, so that it is possible to understand the effect of parameter configuration on their final results. The dataset has 20,000 samples, each consisting of three vectors: a velocity vector with 236 values (representing a vertical profile of a randomly generated 2D layered model), a reflective vector with 600 values obtained directly from the velocity vector, and the associated seismogram vector with 11 traces, each containing 600 values. The overall results show that the WaveNet produces a lower Mean Squared Error between predicted and correct outputs than that of XGBoost. One challenge yet not dealt with is that the WaveNet can train well in GPU, but we did not succeed in doing the same with the XGBoost, due to the amount of data to be processed.

Keywords: WaveNet; XGBoost; Inversion; Seismic.

Resumo

A aplicação de aprendizado de máquina em geofísica aumentou vertiginosamente na última década, com a qualidade de seus resultados variando de acordo com o tipo de problema sísmico em foco e o método computacional empregado. Os métodos de aprendizagem profunda estão alcançando resultados impressionantes nesta área, mas notamos que ainda há uma falta de certeza sobre se os métodos clássicos de aprendizagem de máquina podem fornecer resultados semelhantes. No presente artigo, o objetivo foi tentar preencher parte dessa lacuna, comparando um método bem conhecido de aprendizado de máquina não DL com um método DL para a propagação direta de ondas e os problemas de inversão sísmica para modelos 2D em camadas horizontais. Ambos os métodos são avaliados em cenários diferentes, mas sob condições semelhantes, para que seja possível compreender o efeito da configuração dos parâmetros nos seus resultados finais. O conjunto de dados tem 20.000 amostras, cada uma consistindo em três vetores: um vetor de velocidade com 236 valores (representando um perfil vertical de um modelo em camadas 2D gerado aleatoriamente), um vetor reflexivo

com 600 valores obtidos diretamente do vetor de velocidade e o vetor associado sismograma com 11 traços contendo 600 valores cada. Os resultados gerais mostram que o WaveNet atinge um MSE inferior entre os resultados previstos e corretos do que os resultados do XGBoost. Um desafio ainda não enfrentado é que o WaveNet consegue treinar bem em GPU, mas não conseguiu fazer o mesmo com o XGBoost, devido à quantidade de dados a serem processados.
Palavras-chave: WaveNet; XGBoost; Inversão; Sísmico.

Resumen

La aplicación del aprendizaje automático en geofísica ha aumentado drásticamente en la última década, y la calidad de sus resultados varía según el tipo de problema sísmico en cuestión y el método computacional empleado. Los métodos de aprendizaje profundo están logrando resultados impresionantes en esta área, pero observamos que todavía no hay certeza sobre si los métodos clásicos de aprendizaje automático podrían proporcionar resultados similares. En el presente artículo, el objetivo era intentar llenar parte de ese vacío, comparando un método de aprendizaje automático no DL bien conocido con un método DL para la propagación directa de ondas y los problemas de inversión sísmica para modelos 2D de capas horizontales. Ambos métodos se evalúan en escenarios diferentes, pero bajo condiciones similares, de modo que es posible comprender el efecto de la configuración de parámetros en sus resultados finales. El conjunto de datos tiene 20.000 muestras, cada una de las cuales consta de tres vectores: un vector de velocidad con 236 valores (que representa un perfil vertical de un modelo en capas 2D generado aleatoriamente), un vector reflectante con 600 valores obtenidos directamente del vector de velocidad y el vector de velocidad asociado. Vector de sismograma con 11 trazas que contienen 600 valores cada una. Los resultados generales muestran que WaveNet alcanza un MSE más bajo entre las salidas previstas y correctas que el resultado de XGBoost. Un desafío aún no abordado es que WaveNet puede entrenarse bien en GPU, pero no logró hacer lo mismo con XGBoost, debido a la cantidad de datos a procesar.
Palabras clave: WaveNet; XGBoost; Inversión; Sísmico.

1. Introduction

During the last five years, the number of scientific publications using Deep Learning methods (DL) in geophysics has steeply increased. Among the geophysics problems investigated, the simulation of wave propagation and the seismic inversion have the highest demand of computational resources (Krebs *et al.*, 2009; She *et al.*, 2019). Therefore, they have received greater attention from the Deep Learning community and are the focus of the current paper. In general, the scientific works in this field can be grouped according to the dimension of the geological model being used (2D or 3D) and the type of DL method employed.

Regarding the nature of the data, 3D velocity/density models with complex geological features express well the earth structure in practical cases. Nevertheless, due to the large amount of computation resources to deal with such models, 2D models have often been employed in academia. 2D models allow the theoretical validation of new concepts and methods, which can later be extended to more complex problems. In fact, even very simple geological structures are commonly employed in geophysics studies with Deep Learning, including layered models. Wu *et al.* (2018) used horizontally-layered models having a geological fault for studying the application of a convolutional neural network (CNN) for the inversion problem. Junior *et al.* (2019) applied a Physics-Informed Neural Network (PINN) to solve the elastic wave equation in a three-layered velocity model for the estimation of petroelastic properties. Moseley *et al.* (2020) studied the wave propagation and inversion problems with a WaveNet network and an auto-encode network design for both plain horizontally-layered models and similar models that have a geological fault.

Despite the impressive results obtained by the use of DL in this area, there is still a lack of certainty on whether or not classical machine learning methods (not based on deep learning) can provide similar results. The advantages of using classical ML methods are many: they are more widely available in libraries for several programming languages; they have less configuration parameters and are, therefore, easier to tune; their strengths and weaknesses are more well-known; and there are more advanced machine-learning explainable strategies for them.

In the present paper, we attempt to fill part of that gap, by comparing a well known non-DL machine-learning method with a DL method for the direct wave propagation and the seismic inversion problems for 2D horizontally-layered models. The chosen standard machine learning method is XGBoost. The DL method is the WaveNet algorithm employed by Moseley *et al.* (2020), which consists of one of the most recent investigations into that type of model. Both methods, WaveNet and XGBoost, are evaluated in different scenarios, but in similar conditions, so that it is possible to establish the effect of parameter configurations on the training time and the quality of their outcome.

The remainder of this paper is organized as follows: Section 1 provides a background about the WaveNet and XGBoost methods and discusses their use in order to solve a given seismic problem. Section 2 presents the approaches with experimental setups for comparing both methods, including the description of the employed dataset. Section 3 provides our results and discussions. Finally, Section 4 carries out the conclusion about the work.

1.1 Machine Learning Methods

1.1.1 WaveNet

WaveNet was introduced by Oord *et al.* (2016) as a deep neural network for generating raw audio waveforms. It was based on the PixelICNN architecture and was originally applied to the generation of raw speech signals and music, and to audio recognition.

Given that a seismic wave is similar to a sound wave, it is intuitive to think of using WaveNets for the processing of seismic data. Moseley *et al.* (2020) explored this idea by proposing the application of a WaveNet for learning the wave propagation process and also the seismic inversion, focused on 2D models with simple horizontal layers. The authors trained the WaveNet on a data set consisting of 50,000 randomly generated cases. Each case had a velocity profile of 256 points (that, when expanded, would represent a 200 x 256 velocity matrix) and a seismogram with eleven traces (11 receivers x 600 time steps). The seismograms were generated by calculating a 2nd-order acoustic Finite Difference (FD) modeling with CPML attenuation at the borders. Before usage, the velocity profiles were converted to their corresponding reflectivity signals (with the same dimension of the traces, i.e, 600 data points for each seismogram trace).

The WaveNet was implemented with a TensorFlow library, for the wave propagation problem, and had 9 convolutional layers, 256 hidden channels and a filter size of 3. The input and output data pairs were the reflectivity signals and the seismograms, respectively. Moseley *et al.* used a learning rate of 10^{-5}, a batch size of 20 training examples and 300,000 epochs. The Adam stochastic gradient descent algorithm was employed with a L2 misfit function given by

$$L2 = \frac{1}{N} \| G \cdot (\hat{Y} - Y) \|_2^2, \tag{1}$$

where $\hat{Y}$ is the simulated seismogram response from the WaveNet, Y is the real seismogram produced by a FD modeling, G is a gain function with the fixed form G(t) = t_g, with $g = 2.0$, t = 1,2,…,600, and N the number of training cases in each batch.

Moseley *et al.* (2020) showed that the WaveNet was capable of reconstructing the seismogram for a set of 1,000 unseen randomly-generated cases, achieving high accuracy when compared numerically and visually to a 1D convolutional wave propagation model.

For the inversion problem, the same structure of the WaveNet was used with some minor changes: the layers of the network were inverted to reflect the nature of the problem and it had 128 hidden channels instead of 256. The input data and output data consisted of the seismogram and the reflectivity signals, respectively. The function to be minimized by the network was similar to Equation 1, but was based on the reflectivity signals as shown below:

$$L = \frac{1}{N} \| \hat{R} - R \|_2^2 \tag{2}$$

where $\hat{R}$ is the predicted reflectivity and R is the real one.

The authors also reported good results of the WaveNet for the inversion problem when compared to a dilated convolutional network.

1.1.2 XGBoost

Extreme Gradient Boosting (XGBoost) is a machine learning technique based on gradient tree boosting. It was developed to work with a second-order Taylor expansion on the loss function, allowing distributed training. XGBoost implements various optimisation strategies, which makes it much faster to train and more scalable than previous approaches. This is because XGBoost was developed to work with a second-order Taylor expansion on the loss function, allowing distributed training, doing faster than its predecessor, as described by Chen and Guestrin (2016) and Zou *et al.* (2020). It has been successfully used in Kaggle competitions for solving different problems.

Following the notations of Chen and Guestrin (2016), the general optimization problem embedded in XGBoost can be described as presented next.

For a given data set *D* with n examples and *m* features,

$$D \;=\; \{(X_i, y_i)\}\;(|D| \;=\; n, X_i \,\epsilon\, \mathbb{R}^m, y_{i\square} \epsilon\;\; \mathbb{R}), \tag{3}$$

the tree ensemble model can be written as

$$\hat{y}_i \;=\; \phi\,(X_i\,) \;=\; \textstyle\sum_{k=1}^{k} \square\, f_k\,(X_i\,), f_k\;\,\epsilon\, F, \tag{4}$$

where *k* is the quantity additive function to predict the output $\hat{y}_i$, and f_k belong to the space of regression trees. To find a solution to equation (3), it is necessary to minimize the loss and regularization objectives describe by Mitchell and Frank 2017 as:

$$Obj = \textstyle\sum_i^{\square} \square\;\; L\,(y_i\;\; , \hat{y}_i\,) + \;\textstyle\sum_i^{\square} \square\;\; \Omega\,(\,f_k), \tag{5}$$

with *L* the loss function, and *Ω* a regularization term that measures how complex the tree model is and helps to avoid the overfitting problem. The regularization term can be defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda||w||^2. \tag{6}$$

In this equation, f_k is a tree of the model, *γ* and *λ* are configurable regularization coefficients, *T* is the number of leaf nodes of the tree, and *w* is a vector of weights of the leaves. The first term *γT* penalizes for adding tree leaf and the second term penalizes when the *w* is extreme. After applying a mathematical transformation describe by Zhang *et al.* (2019), we obtain a new expression for the objective function, that measures the quality of the tree model:

$$Obj = -\frac{1}{2}\sum_{t=1}^{T} \frac{(\sum_{i\in I_{T-t}} g_{m,i})^2}{\sum_{i\in I_t} h_{m,i} + \lambda} + \gamma T, \tag{7}$$

where $g_{m,i}$ and $h_{m,i}$ are the first and second derivatives of the objective function.

1.2 XGBoost In Seismic

XGBoost is a powerful machine learning algorithm that has gained widespread popularity in recent years due to its ability to handle large datasets with high accuracy and efficiency. In the seismic area, XGBoost has proven to be an effective tool for predicting earthquake occurrences and analyzing seismic data.

Seismic data can be complex and challenging to analyze due to its high dimensionality and variability. XGBoost is well-suited to these challenges because it can effectively handle large feature spaces, deal with missing data, and automatically detect non-linear relationships between variables. This makes it a suitable tool for identifying patterns and relationships within seismic data, enabling researchers to better understand seismic activity and predict future events.

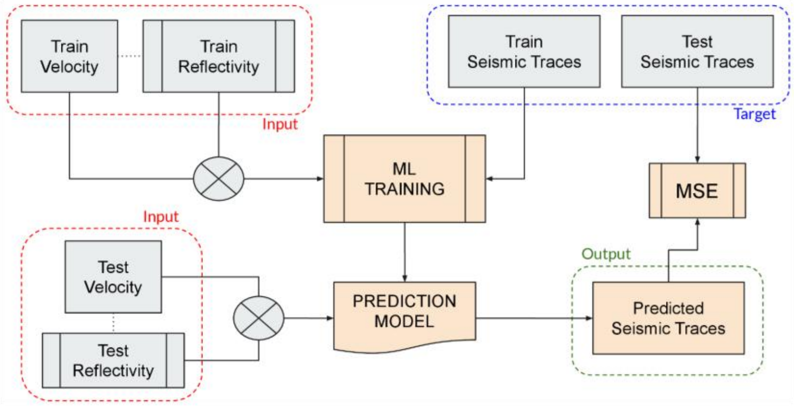
One of the first reported studies of the use of XGBoost in seismic was described by Priezzhev et al. (2019). The problem solved in that study was the improvement of the seismic characterization of a fluvial-deltaic reservoir in the Zapotal field, in the Talara basin. Seismic characterization is important for understanding the distribution of fluids in the subsurface and the identification of possible hydrocarbon reservoirs, being essential for the exploration and production of oil and gas. The output of the study was a better seismic characterization of the fluvial-deltaic reservoir in question.

2. Methodology

As previously mentioned, our aim is to compare the WaveNet and the XGBoost approaches on two groups of tasks, related to direct wave propagation and seismic inversion, using 2D layered models. For carrying out such a comparison, we used the dataset[1] built by Moseley *et al.* to test the WaveNet in one of their preliminary works (Moseley *et al.*, 2018). The dataset has 20,000 samples, produced via a process similar to the one described in Section 2.1. Each sample consists of three vectors: a velocity vector with 236 values (representing a vertical profile of a randomly-generated 2D layered model), a reflective vector with 600 values directly obtained from the velocity vector, and the associated seismogram vector with 11 traces containing 600 values each. Due to the simple horizontal-layered structure assumed, only one profile in depth is sufficient to represent a velocity model.

For the present research, we split the dataset into a set for training with the first 18,000 samples, and a set for testing with the remaining 2,000 samples. A further separation of the training samples was performed in order to support the calibration of the XGBoost, as explained later. The seismograms were amplified by applying the gain function *G* (mentioned in Equation 1) on every trace, as a preprocessing phase.

Figure 1 - Flowchart detailing the input, output, and metrics data in the machine learning methods used.



[1] The dataset is available at https://github.com/benmoseley/seismic-simulation-wavenet. Source: Authors (2024).

For the direct wave propagation tasks, we followed the framework shown in Figure 1. Firstly, a machine learning model (ML) is trained using the reflectivity data (computed from the velocity data) and the related seismic traces from the training samples. Then, the trained model runs. It inputs the reflectivity data from the test set and outputs predicted seismic traces. Finally, the Mean Squared Error (MSE) between the predicted seismograms and the test (correct) seismograms are computed. The output consists of MSE measures (minimum, maximum, mean and standard deviation of the MSE calculated on the 2,000 test samples), that, combined with the required time for training, represents the performance of the machine learning method.

The ML method could be either a WaveNet or a XGBoost. For the first option, we adopted the WaveNet with the same parameters specified by Moseley *et al.* (2018) and briefly described in Section 1.1.1. When choosing the XGBoost, we explored and then fixed three parameters: the number of estimators (number of trees), the maximum depth of the trees and the learning rate.

Another difference between the methods is that the WaveNet outputs multiple values, while the XGBoost regressor generates just one value at a time. Therefore, we have to embed the XGBoost in a MultiOutput Regressor approach, which fits one regressor per target, in order to produce results with the correct dimension[1].

The data flow for running seismic inversion tasks was similar to that used for direct wave propagation. The main difference consisted in swapping the inputs and outputs of the ML method. In addition, the internal parameters of the WaveNet for inversion had minor changes, as previously explained. The XGBoost was also set up with the proper parameters tuned for inversion tasks.

All codes were implemented in Python v3.7 and set up for exploiting parallelism when running the machine learning methods. The WaveNet was configured to run on a GPU NVidia Tesla K40, installed on a machine with 2x CPU Intel Xeon(R) Ten-Core E5-2650 v3 of 2.3 GHz, totalizing 20 visible cores, 25 MB of cache and 128 GB of DDR4 2133 DIMM RAM. We also tested a standard GPU implementation of the XGBoost from the Python repository, but it did not work well on that machine, halting with “segmentation fault” for some amounts of training cases. Therefore, the XGBoost exploited only multi-core CPU parallelism, using a machine with 2x CPU Intel Xeon(R) Sixteen-Core E5-2698 v3 of 2.3 GHz, totalizing 64 visible cores with hyperthreads, 40MB of cache and 256 GB of DDR4 2133 DIMM RAM.

The data flow for running seismic inversion tasks was similar to that used for direct wave propagation. The main difference consisted in swapping the inputs and outputs of the ML method. In addition, the internal parameters of the WaveNet for inversion had minor changes, as previously explained. The XGBoost was also set up with the proper parameters tuned for inversion tasks.

All codes were implemented in Python v3.7 and set up for exploiting parallelism when running the machine learning methods. The WaveNet was configured to run on a GPU NVidia Tesla K40, installed on a machine with 2x CPU Intel Xeon(R) Ten-Core E5-2650 v3 of 2.3 GHz, totalizing 20 visible cores, 25 MB of cache and 128 GB of DDR4 2133 DIMM RAM. We also tested a standard GPU implementation of the XGBoost from the Python repository, but it did not work well on that machine, halting with “segmentation fault” for some amounts of training cases. Therefore, the XGBoost exploited only multi-core CPU parallelism, using a machine with 2x CPU Intel Xeon(R) Sixteen-Core E5-2698 v3 of 2.3 GHz, totalizing 64 visible cores with hyperthreads, 40MB of cache and 256 GB of DDR4 2133 DIMM RAM.

[1] For this project, we performed preliminary experiments with both a MultiOutput Regressor approach and a Regressor Chain approach. The Regressor Chain computes predictions in a sequential order using the available original input data extended with the predictions of the earlier models in the chain. Preliminary experiments showed that the Regressor Chain is much slower than the MultiOutput Regressor and that it did not provide significant improvements. Therefore, we adopted the MultiOutput Regressor as our standard approach.

3. Results and Discussion

3.1 Direct Propagation

We now describe comparative experiments between the WaveNet and the XGBoost methods for direct wave propagation. We start in Section 4.1.1 by analyzing the fine-tuning process of the XGBoost method. Then, the WaveNet and the XGBoost are compared in Section 4.1.2 in terms of training time and accuracy (the mean MSE) as the training set increases in size. Finally, in Section 4.1.3, we analyze the outputs of the WaveNet and the XGBoost for two test samples.

3.1.1 Parameter Tuning

In order to tune the XGBoost method, we used the first 3000 samples of the dataset. From them, the first 1000 were employed as a training set, and the remainder formed the validation set. We explored different values for the Number of Estimators, Learning Rate and Max Depth parameters, as shown in Table 1, consisting of 84 combinations. Each combination was evaluated independently and resulted in two metrics: the training time of the XGBoost and the mean of the MSE values for the 2000 samples.

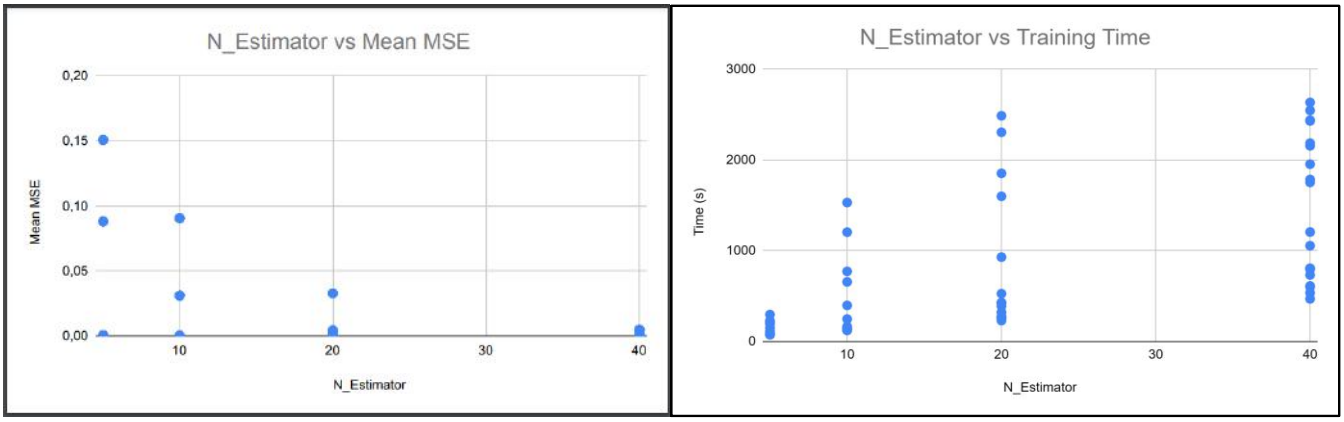
Table 1 - Parameters for tuning the XGBoost for the direct wave propagation.

Parameters	Values
Number of Estimators	5, 10, 20, 40
Max. Depth	5, 10, 20, 40, 50, 100, 200
Learning Rate	0.005, 0.01, 0.05

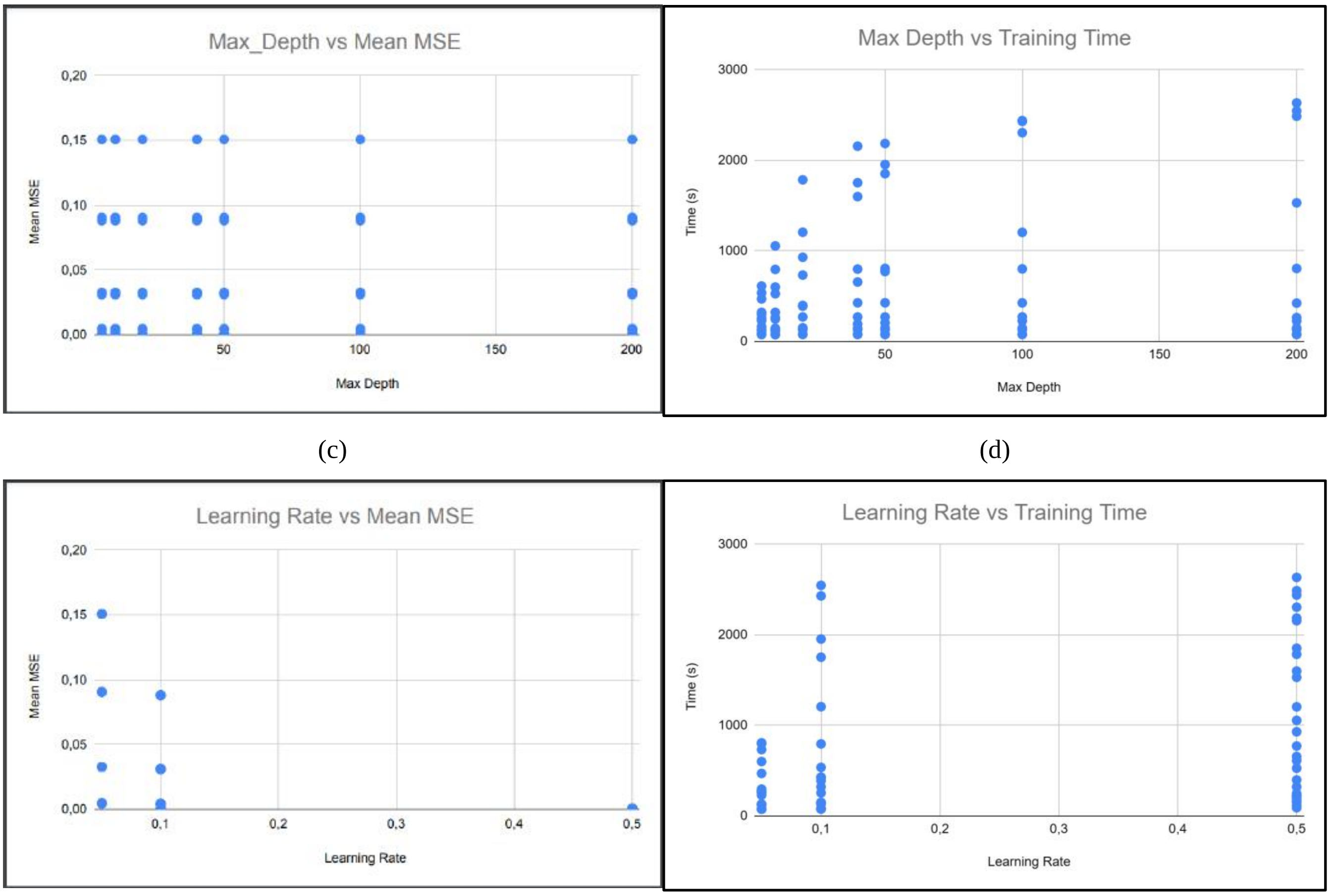
Source: Authors (2024).

Figure 2 shows the effect of the parameters on the metrics, with each dot representing a configuration.

Figure 2 - Parameter tuning of the XGBoost for the forward problem.



(a) (b)



(c) (d)

(e) (f)

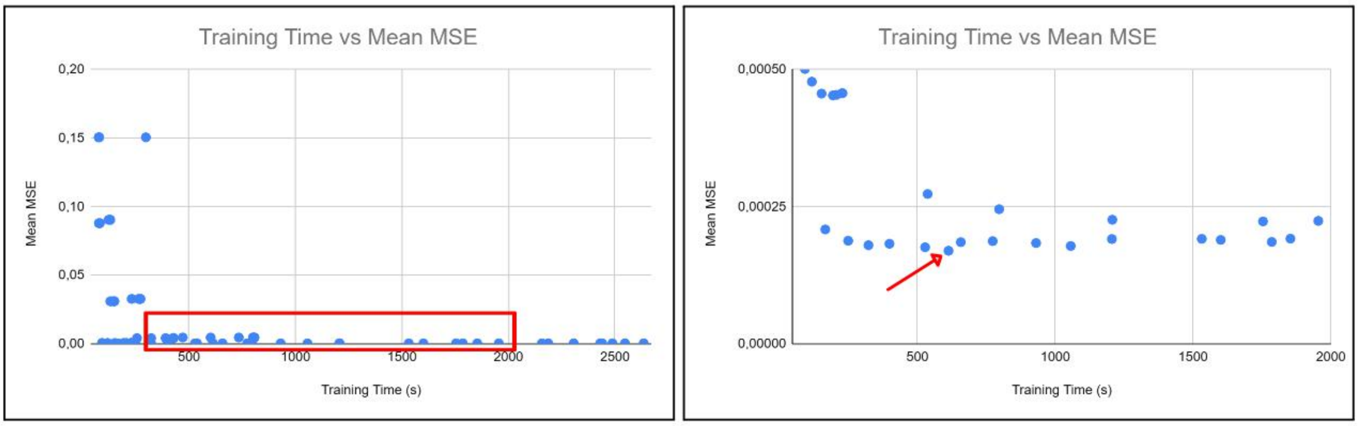
Source: Authors (2024).

In Figure 2(a), we can see that the spread of mean MSE values decreases as the Number of Estimators increases. However, for all Numbers of Estimators, there are configurations that provide very low mean MSE. In Figure 2(b), we have the opposite effect, with the training time (even the lowest ones) increasing with the Number of Estimators. Therefore, reducing the Number of Estimators as much as possible may be a better strategy. For the Max Depth parameter, shown in Figure 2(c) and (d), it can be seen that it has no much effect on the mean MSE but may affect the Training Time. Configurations with higher Max Depth had a larger spread of Training Time.

Finally, based on Figure 2(e) and (f), we can see that the Learning Rate had some effect on the spread and on the lowest value of both the mean MSE and the Training Time. The first Learning Rate (0.001) represents the worst case, while the latest (0.5) is associated with a slightly better set of combinations.

In order to choose a suitable combination, it is necessary to evaluate the trade-off between Training Time and mean MSE, which is possible via Figure 3.

Figure 3 - Tradeoff between the mean MSE and training time in the forward problem.



(a) (b)

Source: Authors (2024).

Figure 3(a) shows all configurations in a single picture. Above 300 seconds of training time, all configurations resulted in Mean MSE below 0.03. There were, however, configurations in that range with a much lower mean MSE. They are indicated in the red rectangle in the lower-left corner of the chart, and are zoomed in Figure 3(b). We manually selected the configuration setup in that figure with the lowest Mean MSE, indicated by an arrow, that has the Number of Estimators equals 20, Max. Depth of 5 and Learning Rate equals 0.5. These values were used in the next experiments of the XGBoost for the forward propagation problem.

3.1.2 Changes in Training Time and MSE Due to the Training Set Increase

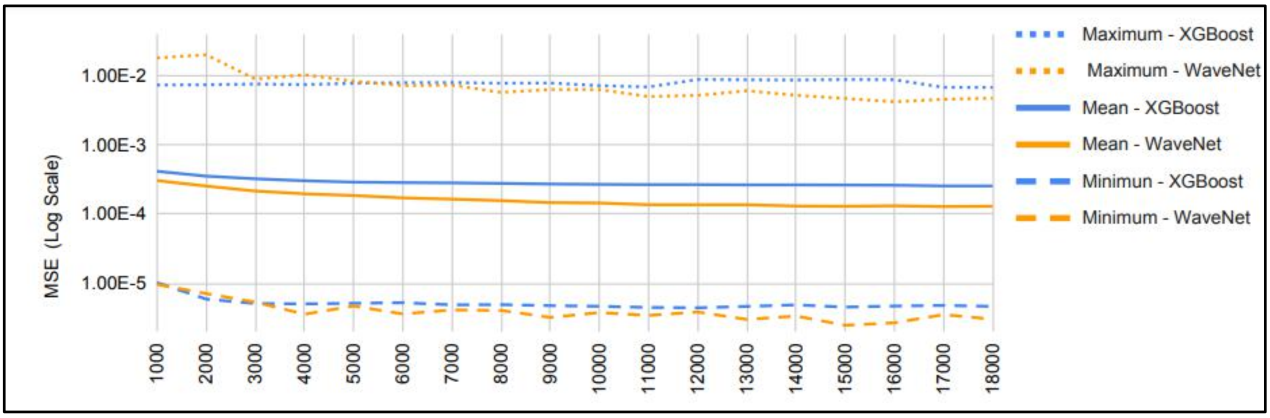
The effect of increasing the amount of training cases on the behavior of both machine learning methods was evaluated, as we can see in Figure 4.

Figure 4(a) illustrates how the number of training cases affects the MSE. The values are quite low, with a slight difference between them, but the WaveNet performed better than the XGBoost in all MSE measures. Furthermore, the WaveNet seems to improve its accuracy over time at a higher rate than the XGBoost.

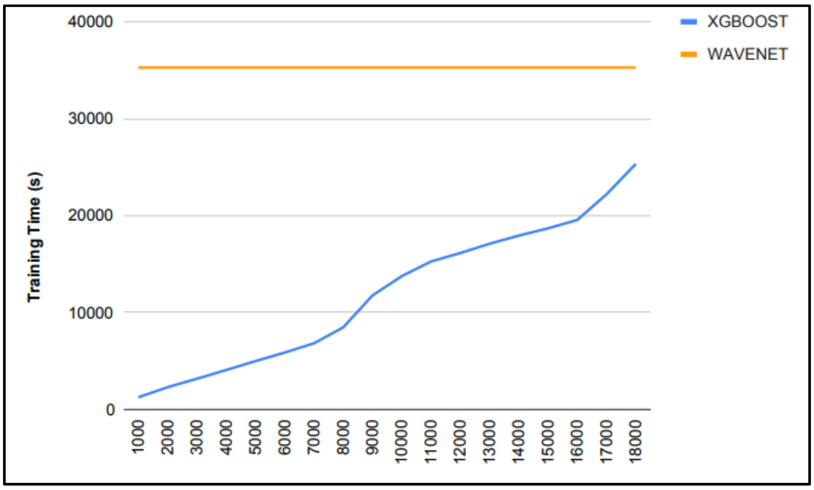
Figure 4(b) shows the changes in training time in seconds as the number of training cases varied. We recall that the WaveNet was run on a GPU, while the XGBoost ran on a multi-core CPU machine. The training time of the WaveNet did not change much as the number of the training cases increased (the number of epochs were kept the same). The training time of the XGBoost, on the other hand, increased steadily with the number of training cases. This behavior demonstrates that the WaveNet will become more efficient to train as the size of the training set keeps increasing. In fact, such turning point can be already reached by using more advanced GPU cards, which can significantly reduce the training time of the WaveNet (to less than one third of the current time).

The prediction time of both methods were very low (around 13 seconds for the WaveNet, and between 15 and 16 seconds for the XGBoost) for the entire 2000 test sample, thus considered negligible at the current work.

Figure 4 - The effect of increasing the number of training cases on (a) the Maximum, Mean and Minimum MSE and on (b) the training time for the direct propagation. Data from the WaveNet are in orange (lighter color), while the ones for the XGBoost are in blue (darker color).



(a)



(b)

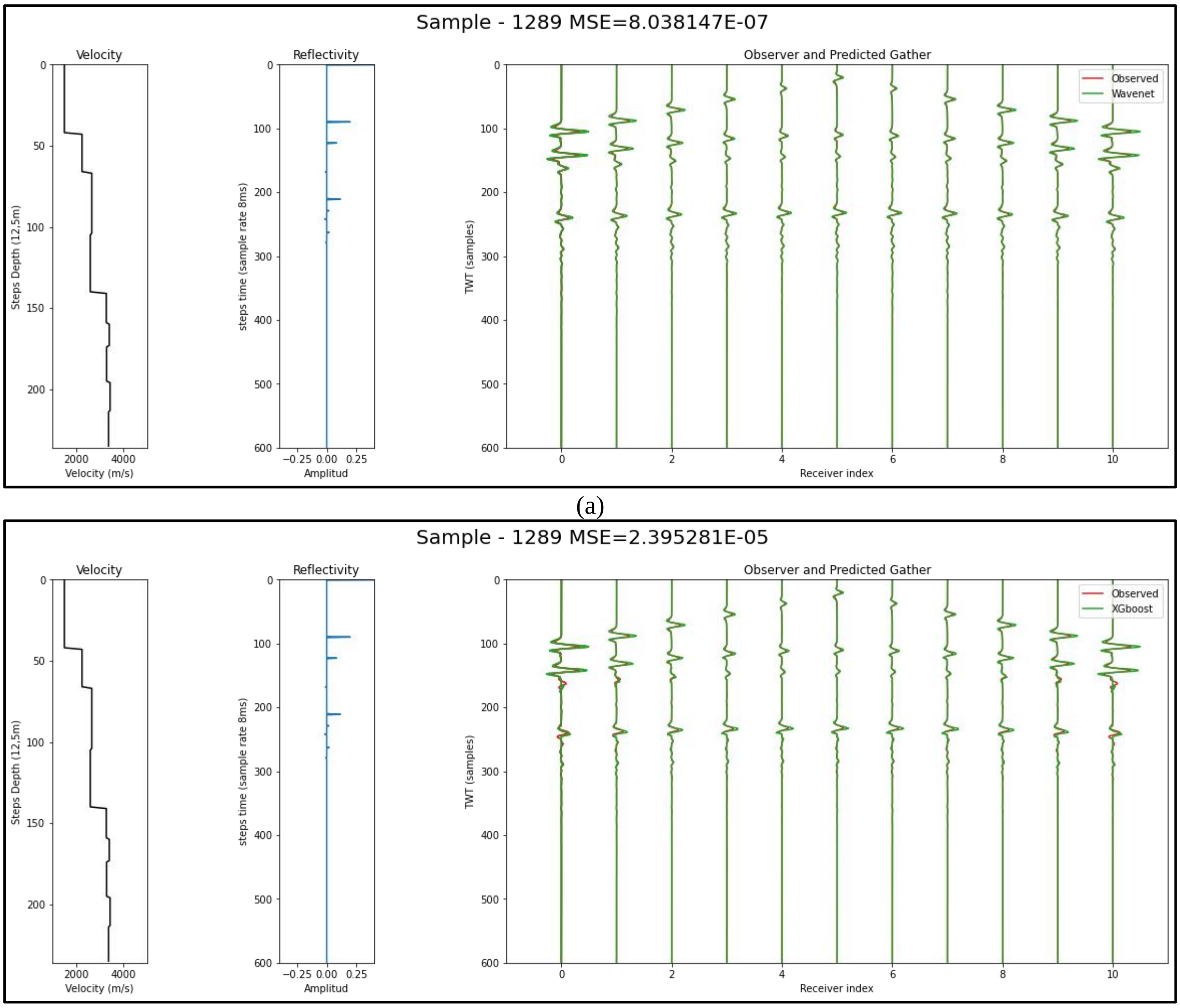
Source: Authors (2024).

3.1.3 Case Analysis

We now illustrate how the WaveNet and the XGBoost perform when producing good results, by means of the analysis of two cases.

Figure 5, with images (a) and (b), refers to Sample 1289, the one for which the WaveNet produced the best result (lowest MSE) among all samples. Figure 5(a) shows the input and the output of the WaveNet, while Figure 5(b) provides the input and the output of the XGBoost. The first two drawings on the left-hand side of these images are the velocity profile and its reflective data. They are the same for both machine learning methods. The right-hand side of the images contains the seismogram produced by the method (by the WaveNet on the top, and by the XGBoost at the bottom) compared against the observed (correct) seismogram.

Figure 5 - Comparison of the methods for the forward propagation using Sample 1289, the best case for the Wavenet. The images are: (a) the input data and the seismogram produced by the WaveNet (MSE=8.03x10^{-7}), and (b) the input and the seismogram of the XGBoost (MSE=2.39x10^{-5}).



(a)

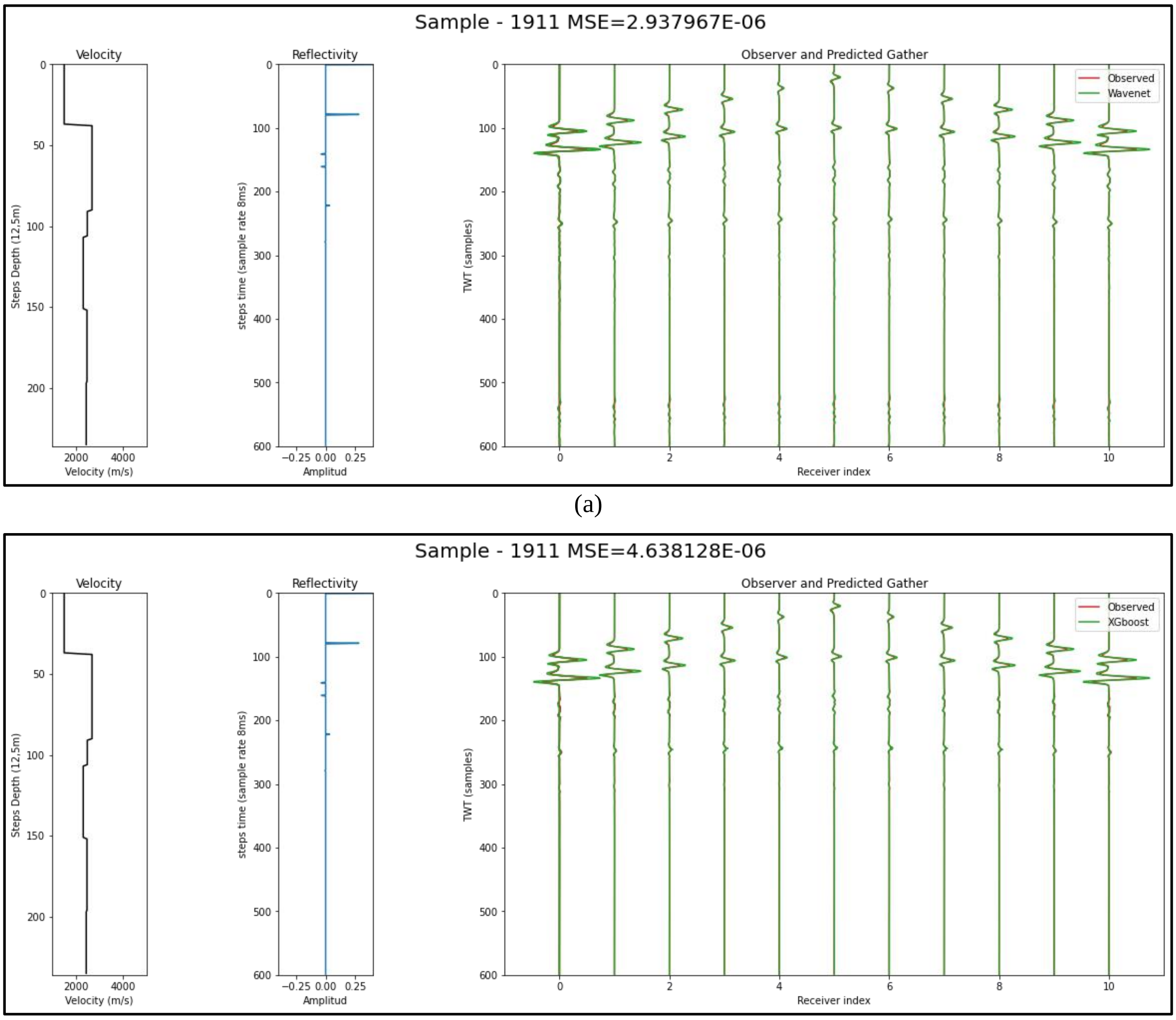
(b)

Source: Authors (2024).

We clearly see that the WaveNet generated a seismogram that most closely matches the observed data. Its MSE is two orders of magnitude lower than that of XGBoost. Interestingly, we see that, when the WaveNet misfitted the correct seismogram, it did so by overestimating its values. The XGBoost showed the same pattern for the first wave signals, but tended to underestimate the remaining data.

Figure 6 refers to Sample 1911, for which the XGBoost produced the lowest MSE among all test samples. In this case, the WaveNet still produced a better result, but its distance in MSE to the XGBoost was much shorter, of only 1.6 times. The mismatch patterns between predicted and correct seismograms observed in Figure 5 are also present in Figure 6 for both methods.

Figure 6 - Comparison of the methods for the forward propagation using Sample 1911, the best case for the XGBoost. The images are: (a) the input data and the seismogram produced by the WaveNet (MSE=2.93x10^{-6}), and (b) the input data and the seismogram of the XGBoost (MSE=4.63x10^{-6}).



(a)

(b)

Source: Authors (2024).

3.2 Seismic Inversion

We now compare the WaveNet and the XGBoost methods for the seismic inversion problem, following the same steps taken in the previous sections for the direct wave propagation problem. In Section 4.2.1, we describe a parameter tuning study of the XGBoost. In Section 4.2.2, we investigate the effect of the size of the training set on the training time and on the MSE for both ML methods. In Section 4.2.3, we compare the results of the methods (the outputted reflectivity/velocity profile) for two test samples. It is worth noting that, for the inversion problem, the WaveNet also ran on a GPU, and the XGBoost had to be run on a multicore CPU machine. However, the XGBoost required more RAM memory, thus using the same multi-CPU machine than before but with 512 GB of RAM.

3.2.1 Parameter Tuning

In order to select the best parameters for the XGBoost for the inversion problem, we used again the first 3000 case samples of the database. The first 1000 of them were separated for training, and the following 2000 ones formed the validation set. The main difference between this process and the previous one is that the input and the output data of the XGBoost were swapped here. We also explored parameters for turning the XGBoost, but now with more values, as shown in Table 2, consisting of 280 distinct combinations.

Table 2 - Parameters for tuning the XGBoost for the inverse problem.

Parameters	Values
Number of Estimators	50, 100, 150, 200, 250, 300
Max. Depth	5, 10, 20, 40, 50, 100, 200
Learning Rate	0.005, 0.01, 0.05, 0.1, 0.5

Source: Authors (2024).

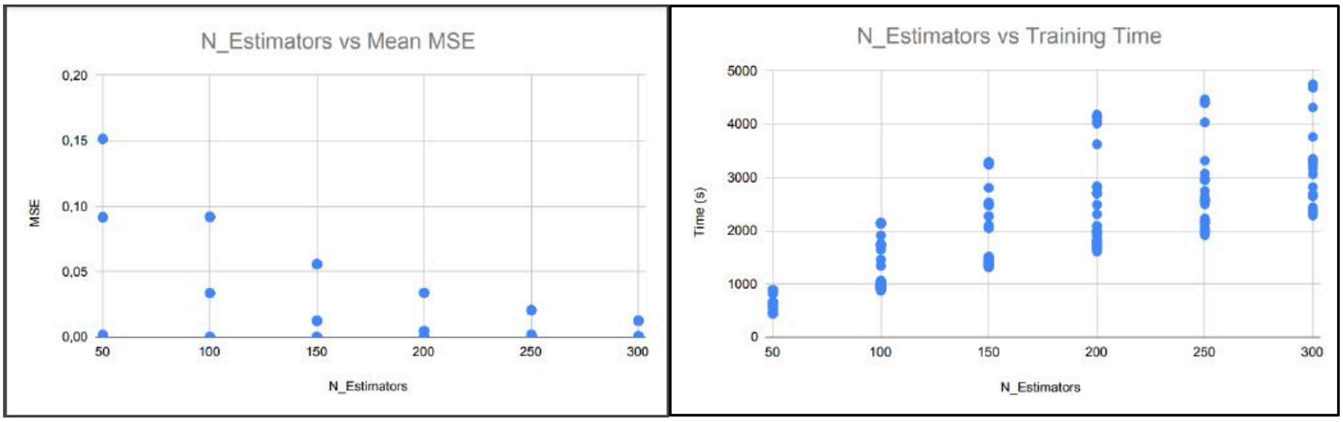
Figure 7 shows the effect of these parameters on the Mean MSE and on the Training Time considering all combinations.

In Figure 7(a), we can see an overall reduction of the MSE with an increase in the number of estimators. There are, nevertheless, combinations that provide low MSE values independently of the number of estimators. In Figure 7(b), we see the opposite effect, with training time increasing as the estimator increases. Therefore, choosing a low number of estimators seems a reasonable option for both MSE and Training Time.

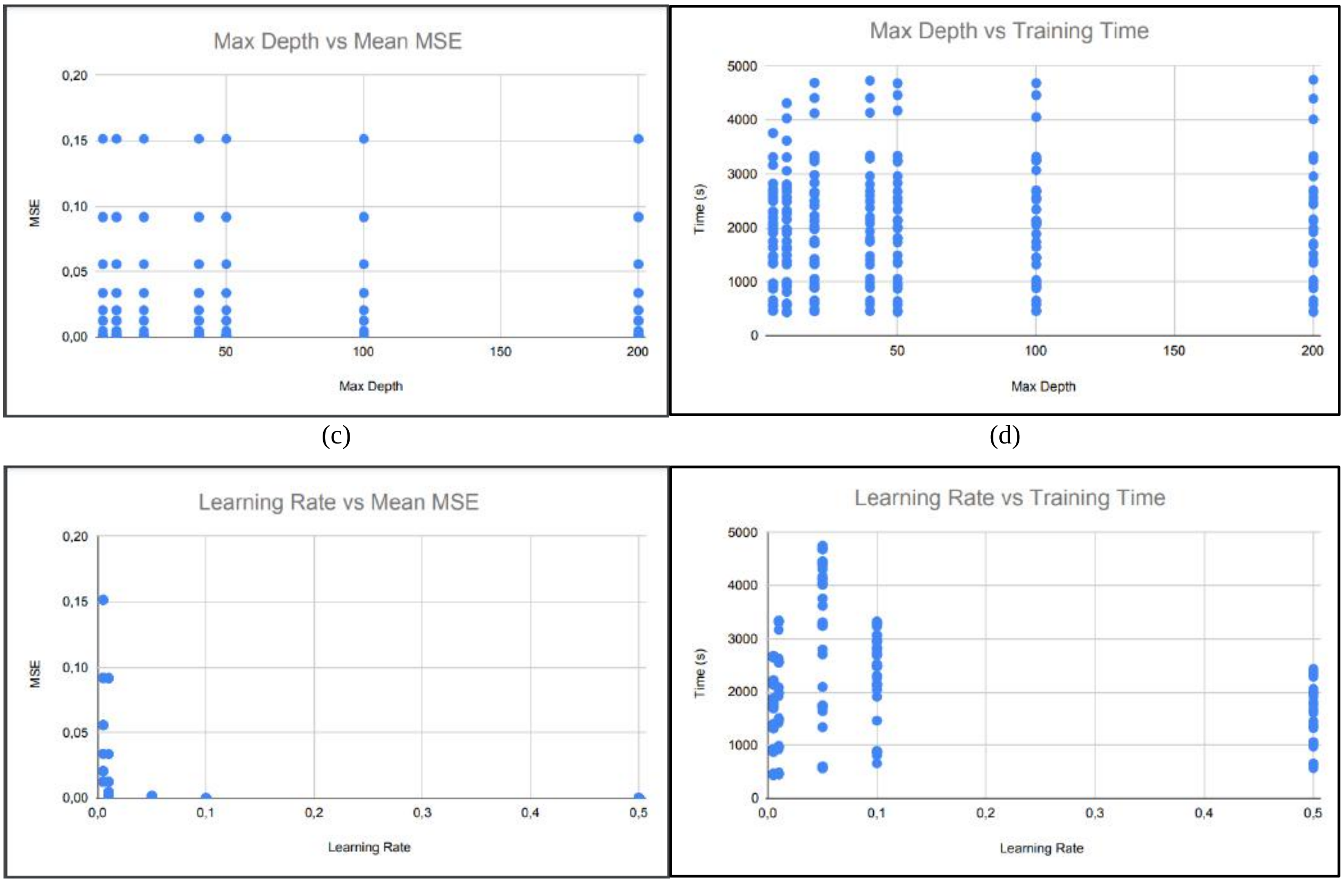
Figures 7(c) and (d) show that both MSE and training time are not significantly affected by the maximum depth parameter. Regarding the learning rate, Figure 7(e) shows that there is a slight decrease in the MSE as the value of this parameter increases. Small values of training time, however, can be achieved at extreme learning rates of 0.005 and 0.5, as demonstrated in Figure 7(f).

Finally, Figure 8(a) shows the relationship between training time and MSE for the studied configurations. We see a progressive reduction in MSE as training time increases. However, there are parameter combinations with low MSE and low training time, as highlighted in the red rectangle in the picture. Figure 8(b) shows an enlarged image of the highlighted area, with a red arrow indicating the chosen configuration. This configuration represents an intermediate option between low MSE and low training time, and consists of the following parameters: Number of Estimator equals 50, Max. Depth of 5 and Learning Rate equals to 0.5.

Figure 7 - Parameter tuning for the XGBoost in the inversion problem.



(a) (b)

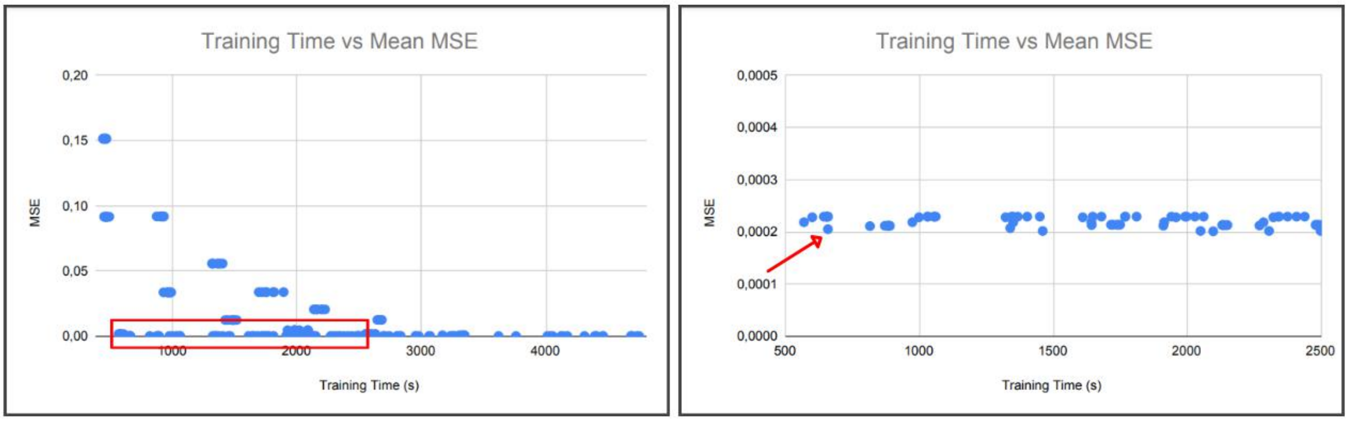


(c) (d)

(e) (f)

Source: Authors (2024).

Figure 8 - Tradeoff between mean MSE and training time in the inversion problem.



(a) (b)

Source: Authors (2024).

3.2.2 Changes in Training Time and MSE Due to the Training Set Increase

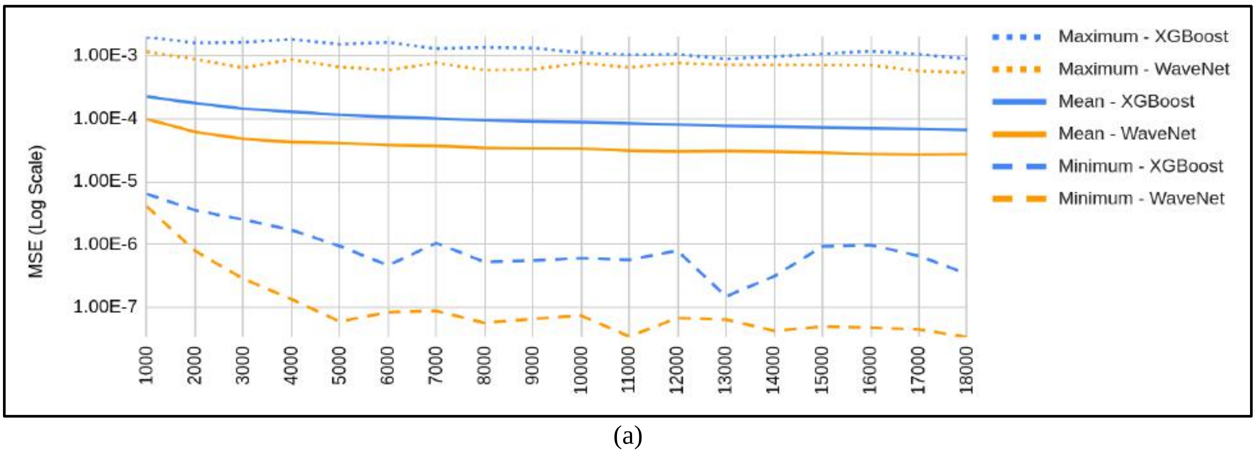
Similar to what was done for the forward propagation, we also analyzed the effect of increasing the size of the training set (number of training cases) on the mean MSE and on the training time of machine learning models.

As shown in Figure 9(a), both methods tend to improve their results as the training set grows, but the WaveNet method obtained lower minimum, mean and maximum MSE values than XGBoost. The difference between the minimum MSE values of WaveNet and XGBoost also varied greatly in the current scenario, showing that XGBoost was not stable in providing lower MSE values than the other method.

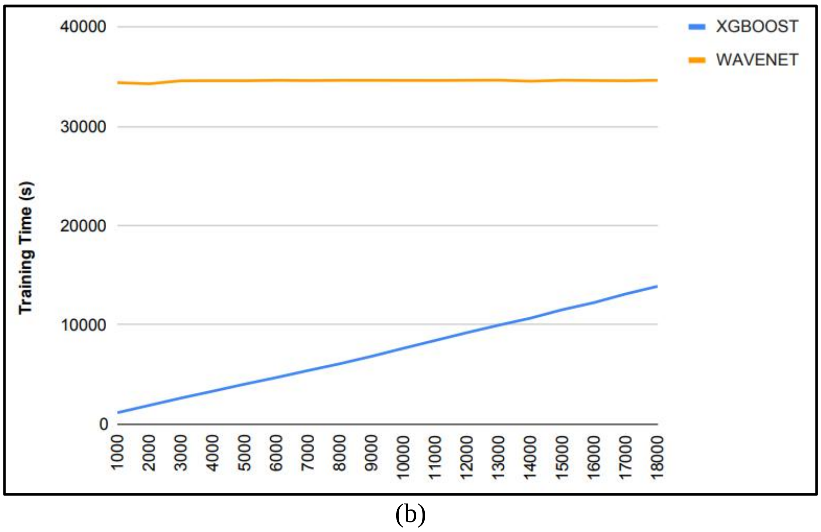
Figure 9(b) is similar to Figure 4(b), with the WaveNet training time being constant and XGBoost training time raising as the training set size increases. Nevertheless, a new aspect should be noted: the training time of XGBoost for solving the inversion problem is shorter and grows more regularly than that for the direct propagation. We believe this behavior is due to the fact that the XGBoost output size is much smaller in the inversion problem (256 points compared to 6600 points in the forward problem), thus requiring fewer instances of machine learning models in the MultiOutput Regressor approach (see discussion at the end of Section 3).

The prediction time for the 2000 test samples of the inversion problem was also very low for both methods (around 13 seconds for the WaveNet, and between 15 and 16 seconds for the XGBoost), therefore, considered negligible at the current work.

Figure 9 - The effect of increasing the number of training cases on (a) the Maximum, Mean and Minimum MSE and on (b) the training time for the inverse problem. Data from the WaveNet are in orange (lighter color), while the ones for the XGBoost are in blue (darker color).



(a)



(b)

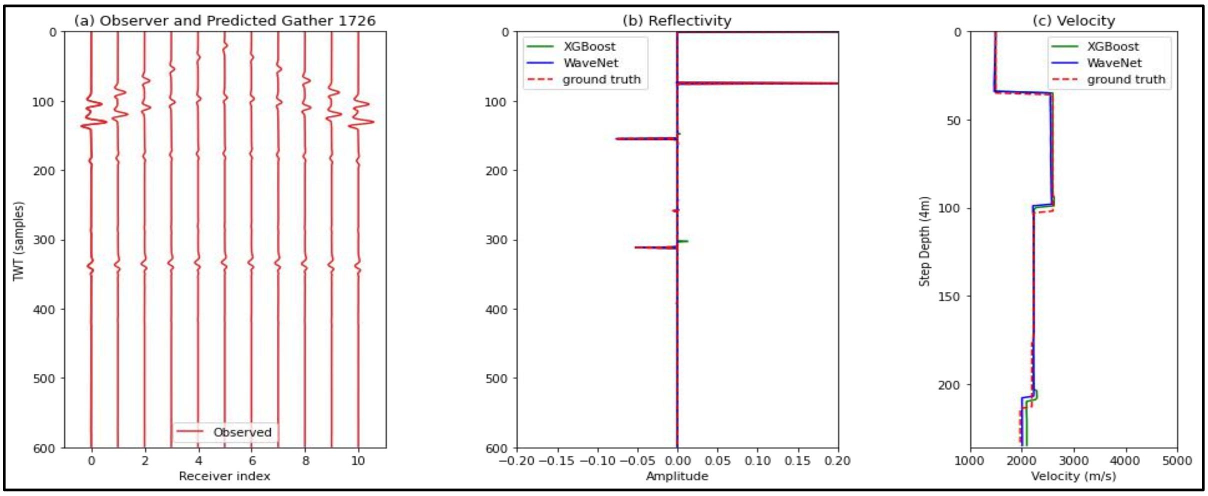
Source: Authors (2024).

3.2.3 Case Analysis

As in Section 4.1.3, the WaveNet and the XGBoost are compared again in terms of their outputs in two cases, now for the inversion problem.

Figure 10 refers to Sample 1726, for which the WaveNet produced the best result (lowest MSE). The first image is the seismogram, used as input to both WaveNet and XGBoost. The next image on the right shows the reflectivity signals generated by the methods and the ground truth reflectivity. The last image on the right-hand side represents the velocity profiles, generated from the reflectivity data. The WaveNet resulted in a reflectivity signal closer to the ground truth (with a lower MSE), although the machine learning methods are very familiar in the initial part of the output data.

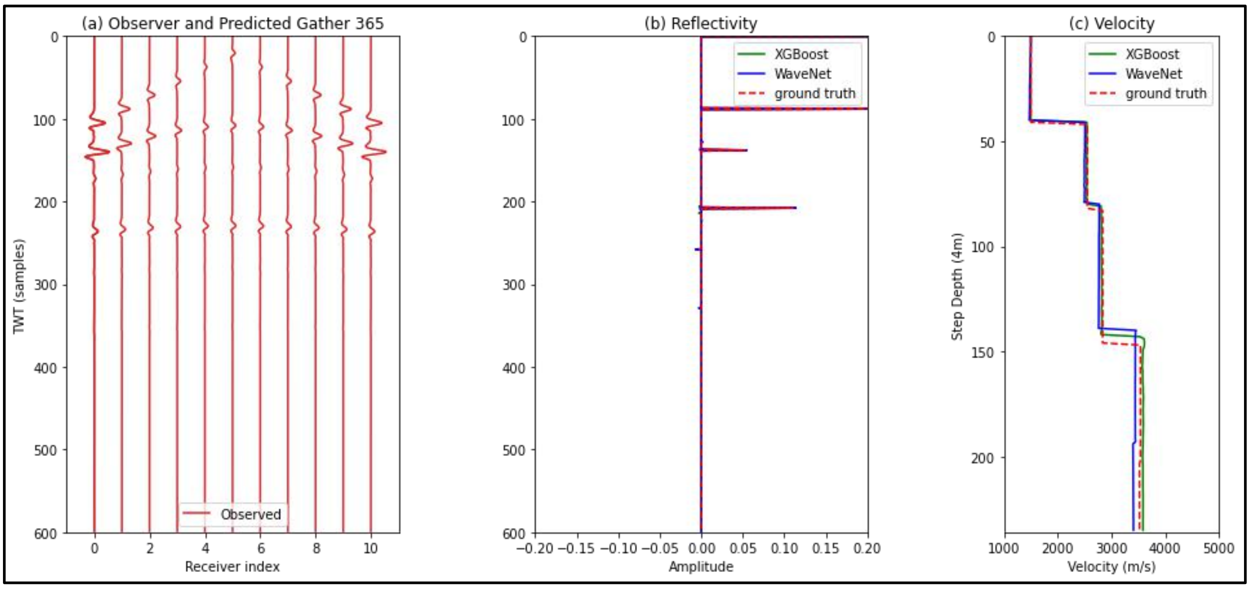
Figure 10 - Comparison of the machine learning methods for the inverse problem using Sample 1726, the best case for the WaveNet. The images are: (a) the input seismogram, (b) the reflectivity signals outputted by the methods (with MSE=7.78x10^{-8}, for the WaveNet, and MSE=1.01x10^{-6}, for the XGBoost), and (c) the associated velocity profiles.



Source: Authors (2024).

Figure 11, on the other hand, refers to Sample 365, for which the XGBoost produced the best result among the 2000 cases. As in the forward propagation, the WaveNet still produces the output with lowest MSE, but the gap between both methods are much shorter.

Figure 11 - Comparison of the machine learning methods for the inverse problem using Sample 365, the best case for the WaveNet. The images are: (a) the input seismogram, (b) the reflectivity signals outputted by the methods (with MSE=1.43x10^{-7}, for the WaveNet, and MSE=3.39x10^{-7}, for the XGBoost), and (c) the associated velocity profiles.



Source: Authors (2024).

4. Conclusion

A comparison between the WaveNet and XGBoost regression methods was performed for both directed propagation and seismic inversion using 2D velocity models with horizontal layers. The results showed that the WaveNet method produced lower MSE values overall, and that it can be competitive in training time as the training set increases in size. This result is important for geologists because it allows them to choose which method is appropriate for the problem at hand. Furthermore, the analysis methodology adopted in this article can be used to compare other machine learning methods.

As future work, we recommend exploring more attributes to investigate possible improvements in Wavenet and XGBoost results. For example, reflectivity and velocity data can be concatenated and used together as a single input/output to train machine learning models. As a classic method, the XGBoost can benefit from higher-level attributes, which provide more descriptive information than standard wave signals. Thus, a compact version of the velocity profile could be used, describing, for each layer, its velocity and depth value. Furthermore, given that WaveNet and XGBoost make mistakes in complementary aspects, a possible strategy is to combine them in an ensemble model, aiming to improve their results. Finally, studying how to efficiently run the XGBoost multiregression on multi-GPU architectures represents a useful contribution.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES), Finance Code 001. It was also supported by LaMCAD/UFG and NPAD/UFRN, which provided high-performance computing resources to the execution of the experiments. Finally, we thank Prof. Leslie Richard Foulds, from INF/UFG, for the detailed review of this article.

References

Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. *In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785-794).

Junior, A. C. N., Almeida, J. D. S., Quiñones, M. P., & de Albuquerque Martins, L. S. (2019, June). Physics-based machine learning inversion of subsurface elastic properties. *In 81st EAGE Conference and Exhibition 2019* (Vol. 2019, No. 1, pp. 1-5). European Association of Geoscientists & Engineers.

Krebs, J. R., Anderson, J. E., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A., & Lacasse, M. D. (2009). Fast full-wavefield seismic inversion using encoded sources. *Geophysics, 74(6), WCC177-WCC188.*

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, 3, e127.

Moseley, B., Markham, A., & Nissen-Meyer, T. (2018). Fast approximate simulation of seismic waves with deep learning. *arXiv preprint arXiv:1807.06873.*

Moseley, B., Nissen-Meyer, T., & Markham, A. (2019). Deep learning for fast simulation of seismic waves in complex media. *Solid Earth Discussions*, 2019, 1-23.

Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., & Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499.*

Priezzhev, I. I., Veeken, P. C. H., Egorov, S. V., & Strecker, U. (2019). Direct prediction of petrophysical and petroelastic reservoir properties from seismic and well-log data using nonlinear machine learning algorithms. *The Leading Edge*, 38(12), 949-958.

She, B., Fournier, A., Wang, Y., & Hu, G. (2019). Incorporating momentum acceleration techniques applied in deep learning into traditional optimization algorithms. *In SEG International Exposition and Annual Meeting* (p. D043S119R003). SEG.

Wang, T., Bian, Y., Zhang, Y., & Hou, X. (2023). Classification of earthquakes, explosions and mining-induced earthquakes based on XGBoost algorithm. *Computers & Geosciences*, 170, 105242.

Wu, Y., Lin, Y., & Zhou, Z. (2018). InversionNet: Accurate and efficient seismic waveform inversion with convolutional neural networks. *In SEG International Exposition and Annual Meeting* (pp. SEG-2018). SEG.

Zhang, R., Li, B., & Jiao, B. (2019). Application of XGboost algorithm in bearing fault diagnosis. *In IOP Conference Series: Materials Science and Engineering* (Vol. 490, p. 072062). IOP Publishing.

Zou, C., Zhao, L., Chen, Y., & Wang, Y. (2020). Machine learning based fluids and lithofacies prediction based on the integration of well logging data and seismic inversion. In SEG 2020 Workshop: 2nd SEG Borehole Geophysics, Virtual, 26–27 November 2020 (pp. 119-120). Society of Exploration Geophysicists.