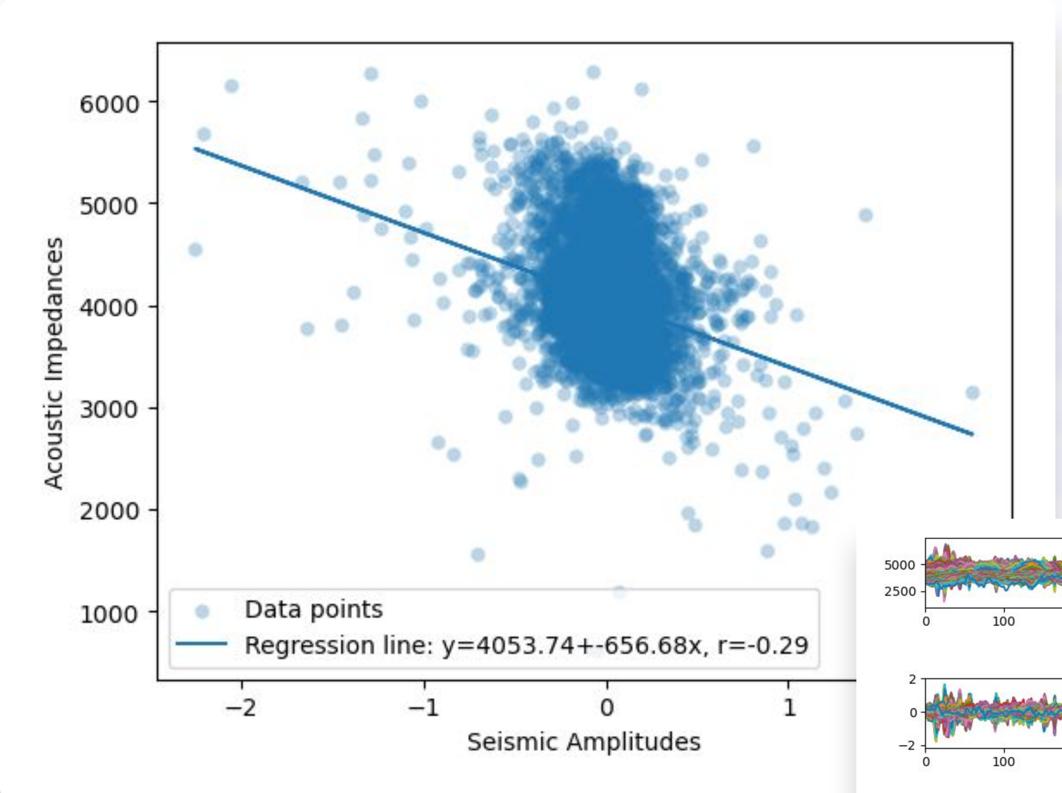
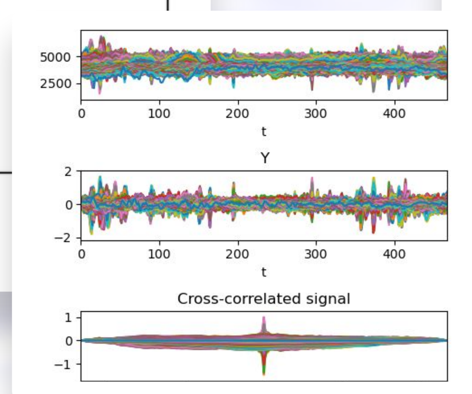
IFP School 2023
BABAYEV Toghrul
TAGHIYEV Vagif

Post-Stack Data-Driven Seismic Inversion

Description

The objective is to estimate the acoustic impedances, from the post-stack amplitudes, not using the conventional seismic inversion but with a data-driven approach. A ML/NN will be used and will learn on a predefined area, from the simultaneous knowledge of the amplitudes and impedances obtained by classical inversion. Then, the network will be launched to invert all the data. The results will be compared to those of the classical inversion.





Introduction

What is stacking in seismic mean?

Post-stack seismic data refers to seismic data that has been processed and analyzed after stacking, which is a technique used to improve the signal-to-noise ratio of seismic data. In the post-stack processing stage, the seismic data is typically filtered, corrected for velocity, and transformed to produce images of the subsurface geology. Post-stack seismic data is often used for interpretation and analysis of subsurface geological structures and formations in the oil and gas industry, as well as in other geophysical applications.

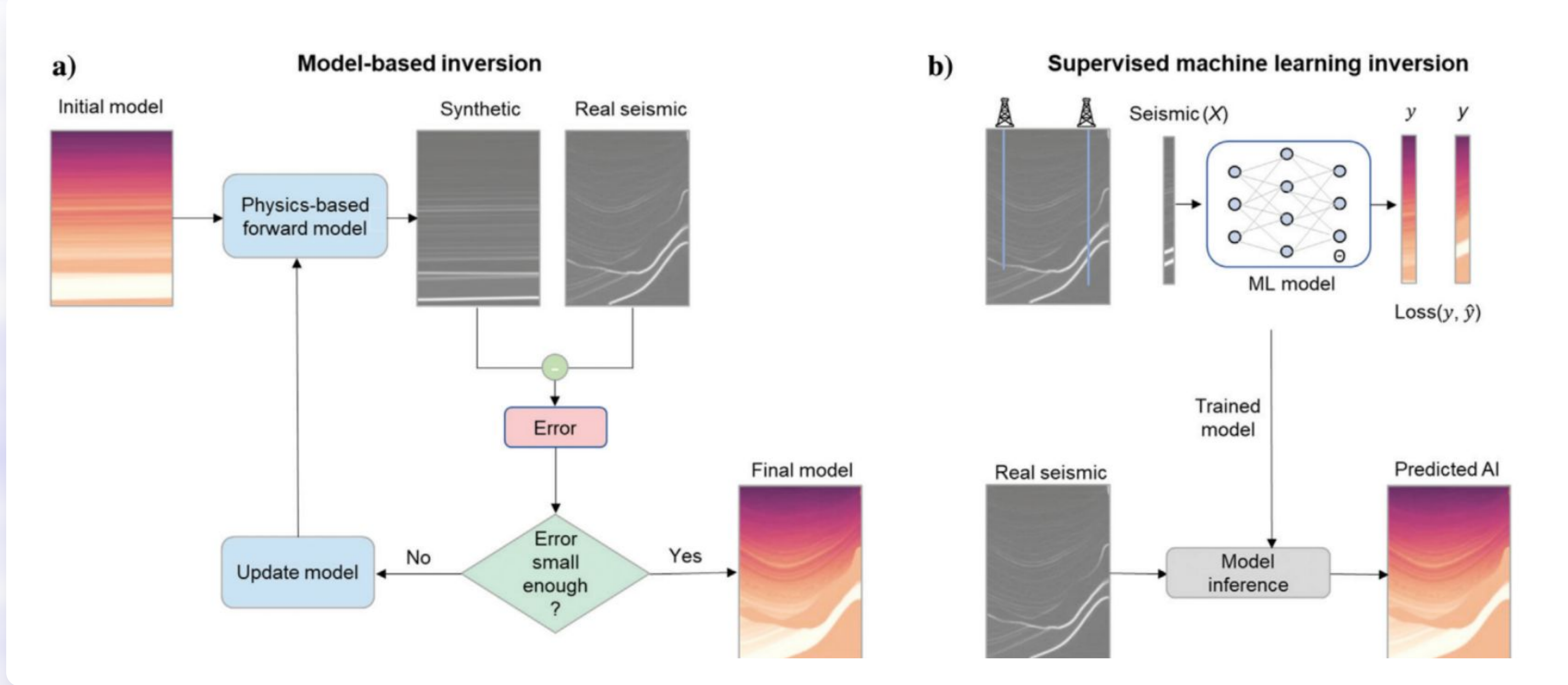
What is seismic inversion?

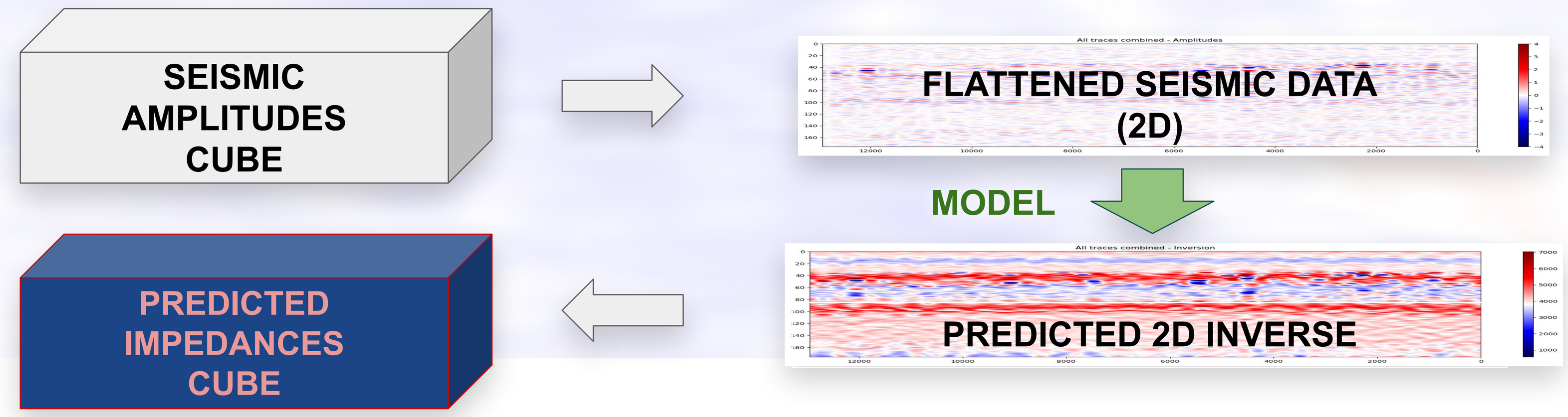
Seismic inversion is a method used in geophysics to estimate the properties of the subsurface rocks and fluids based on seismic data. Seismic waves that travel through the subsurface are affected by the properties of the rocks and fluids they encounter, such as their density, velocity, and acoustic impedance. By analyzing the seismic data and using mathematical algorithms, geophysicists can invert the seismic data to estimate these subsurface properties. Seismic inversion can provide valuable information about the composition and structure of subsurface geological formations, which is useful in many applications such as oil and gas exploration, geothermal energy exploration, and groundwater studies. There are different types of seismic inversion techniques, but the most conventional is use physics-based forward modeling, but it requires computationally complex steps, such as initial model building and wavelet extraction. Noise in the seismic volume also may lead to poor results.

Which ML methods do we use and why?

Advances in machine learning have the potential to overcome some of the challenges faced by conventional seismic inversion techniques. By using machine learning algorithms, geophysicists can potentially improve the accuracy of subsurface property estimations and reduce the time required for model building.

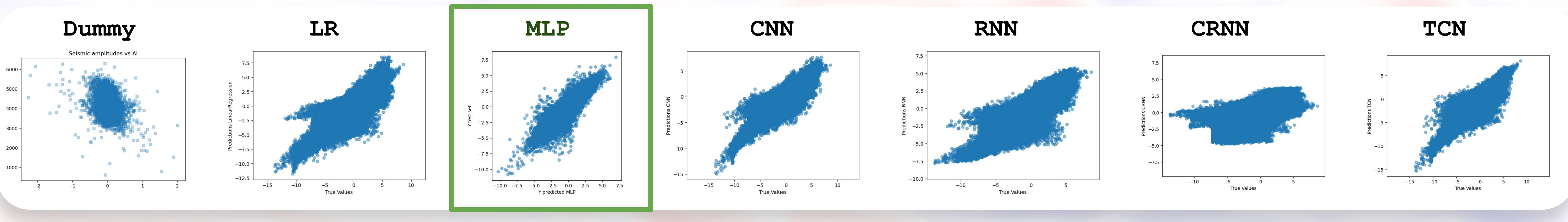
Linear regression is a simple and widely used statistical method for modeling the relationship between variables. It is useful when the relationship between variables is linear and can be used as a baseline model for comparison with more complex models. MLP is a widely used and simple neural network architecture that is good for solving many problems. CNNs are effective in image recognition tasks and have been used in a variety of fields such as medicine, biology, and computer vision. RNNs are useful for tasks that involve sequential data such as language processing and speech recognition. CRNNs combine the advantages of CNNs and RNNs, and have shown good performance in audio classification and speech recognition tasks. TCNs are a newer type of neural network architecture that has shown good performance in sequential data tasks, and can handle long-term dependencies better than RNNs.

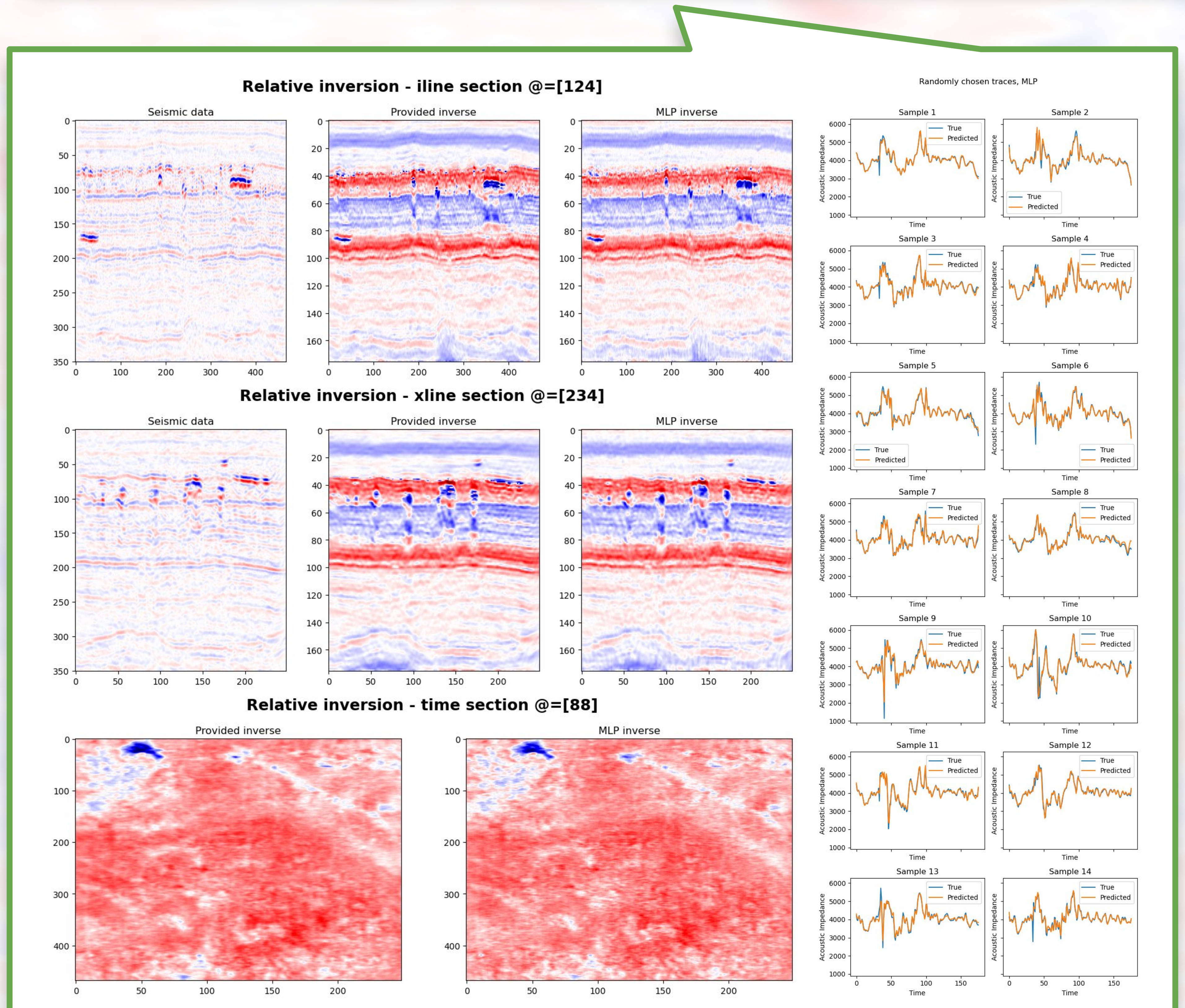


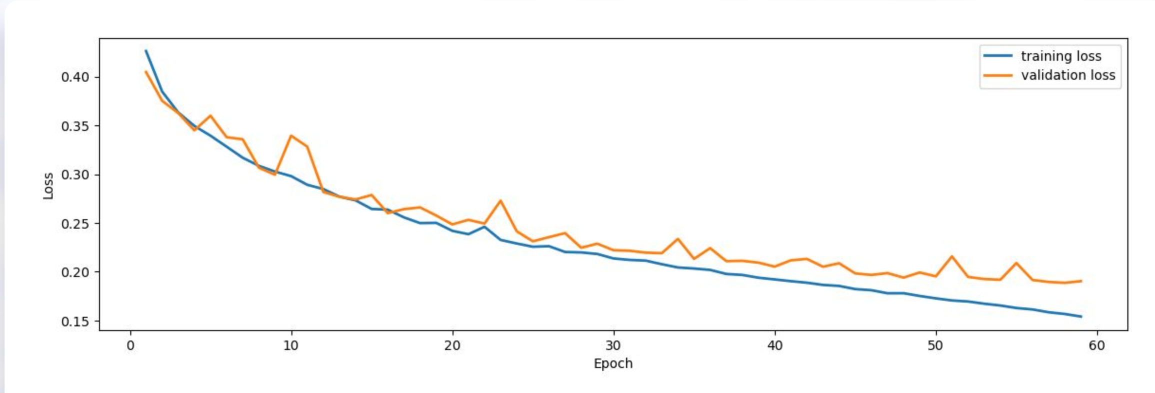


Workflow and Notebook organization

The notebook organization includes several steps for seismic data analysis and modeling using various Python libraries. The first step *(N.1,N.2)* involves a **quick-look** at the seismic cube and inverted cube to gain an initial understanding of the data and its geometry. The next (optional) step *(N.3)* involves **trace-by-trace inversion** using Pylops, which is a Python package for large-scale linear operators. This step involves loading the data, performing grid analysis, regridding, visualization, and generating synthetic data. The T-b-T inversion step (optional) involves regularization (N.4) using Pylops to improve the accuracy of the inversion.
The **original data is then converted** from *.segy* **to numpy** arrays format (N.5), and **exploratory data analysis** is performed. Time cut is then applied to the data to focus on specific **time interval** corresponding to that of Inverted cube, and then data is **reshaped to 2D and downsampled** for performance optimization.
The next step (N.6) involves **model development** using SkLearn and Tensorflow. The data is loaded and reshaped & downsampled, metrics analysis is performed to deduce best metric to evaluate the accuracy of different models. **Different models**, including Dummy (simple one-to-one linear relation), Linear Regression, MLP, CNN, RNN, CRNN, RNN and TCN are **trained** and **tested**. Results are plotted in multiple formats. Finally, the numpy results are **exported to *.segy* format** (N.7).
Several Python libraries are used throughout the analysis, including *Segyio, NumPy, Pandas, MatPLotLib, CuPy, PyLops, CuSignal, SciPy, SkLearn,* and *Tensorflow*. The use of these libraries ensures the analysis is conducted using standard scientific programming practices, making it replicable and scientifically sound.







Conclusion

The aim of this project was to develop and compare different ML methods for predicting acoustic impedance from amplitude data. The models used were Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Convolutional Recurrent Neural Network (CRNN), Temporal Convolutional Network (TCN), and Linear Regression. The models were trained and evaluated using mean squared error (MSE) as the performance metric.
MLP was found to be the most accurate model, with an RMSE of 0.1306. However, the results are not final as hyperparameter optimization was not performed. CNN had an RMSE of 0.1502 and TCN had an RMSE of 0.1904, other models had a much higher errors. Linear regression was also used and had an RMSE of 0.1497.
Disregarding numerical metrics, visually, the most accurate models were MLP, CNN, TCN methods.
Overall, MLP showed the best performance in this project. However, each model has its own advantages and disadvantages and can be further tuned to get better results.
In conclusion, this project demonstrates the effectiveness of different Machine Learning models in predicting acoustic impedance from amplitude data. The results can be used to guide the development of more accurate models for this task.

Model type	Train Time	Predict. Time	Mean Squared Error
T-by-T Inversion	NA	NA	Bad
Dummy	0.3s	0.1s	Terrible
Linear Regression	0.6s	0.2s	0.1497
MLP	11s	0.3s	0.1306
CNN	150s	6s	0.1502
RNN	8m	28s	0.2319
CRNN	5m	21s	0.5910
TCN	10m	51s	0.1904

Further study recommendations

The topic of project has garnered significant attention in the scientific community recently, with a prevailing belief that the TCN model outperforms other machine learning (ML) methods and even conventional model-based seismic inversion (Smith, 2022). However, we observed relatively lower performance of more complex methods in our project, which could be attributed to noise in the source data and subsequent inaccuracies in the provided inversion. Another factor could be that our models did not take into account spatial data, which we believe is a critical factor. We suggest that incorporating data from multiple wells to interpolate the region with rock properties could significantly improve the accuracy of the NN models. Simpler models, on the other hand, may not have significant improvements due to their inability to recognize spatial-temporal patterns as clearly. Additionally, changing the problem type from one-to-one trace to multidimensional input (e.g., 2D section) could improve the results but at a higher computational cost. It is also evident that all our models could perform better if we did not undersample our IL, XL, and time axes. Finally, it is essential to note that "model accuracy" refers to its similarity to the inversion provided by other software, not the actual data.

Reference List:
Robert Smith, Philippe Nivlet, Hussain Alfayez, and Nasher AlBinHassan, (2022), "Robust deep learning-based seismic inversion workflow using temporal convolutional networks," Interpretation 10: SC41-SC55. - https://doi.org/10.1190/INT-2021-0142.1
Fangshu Yang and Jianwei Ma, (2019), "Deep-learning inversion: A next-generation seismic velocity model building method," GEOPHYSICS 84: R583-R599. - https://doi.org/10.1190/geo2018-0249.1
Vishal Das, Ahinoam Pollack, Uri Wollner, and Tapan Mukerji, (2019), "Convolutional neural network for seismic impedance inversion," GEOPHYSICS 84: R869-R880. - https://doi.org/10.1190/geo2018-0838.1
Ravasi, M., & Vasconcelos, I. (2020). PyLops—A linear-operator Python library for scalable algebra and optimization. SoftwareX, 11, 100361. doi:10.1016/j.softx.2019.100361
Code examples:
segyio. (n.d.). segyio: Read and write SEG-Y files using Python. [GitHub repository]. GitHub. https://github.com/equinor/segyio
PyLops. (n.d.). PyLops/pylops. GitHub. https://github.com/PyLops/pylops

We would like to acknowledge the use of a language model, ChatGPT, which assisted in generating and refining our written content