**SPEKTRA: Jurnal Fisika dan Aplikasinya**
Volume 10, Issue 2, August 2025
Homepage: doi.org/10.21009/SPEKTRA

DOI: doi.org/10.21009/SPEKTRA.102.04

# One-Dimensional Modeling of Magnetotelluric Data using Convolutional Neural Network-Gated Recurrent Unit Based Inversion and Its Implementation on Field Data

Achmad Aulia Fikri[1], Nurhasan[1,*], Marshanda Adisti Rahmadini[1], Auza Naufal Abraar[1], Hamzah Firoos Fauzi[1], Dini Fitriani[2]

[1]*Earth Physics and Complex Systems Research Group, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung 40132*
[2]*Department of Geophysics, Faculty of Mathematics and Natural Sciences, University of Padjadjaran, Bandung 45363, Indonesia*

*Corresponding Author Email: nurhasan@itb.ac.id

**Received**: 13 May 2025
**Revised**: 28 July 2025
**Accepted**: 12 August 2025
**Online**: 30 August 2025
**Published**: 31 August 2025

**SPEKTRA**: Jurnal Fisika dan Aplikasinya
p-ISSN: 2541-3384
e-ISSN: 2541-3392



## ABSTRACT

The magnetotelluric method is a geophysical method that utilizes natural variations in the electromagnetic field to map the resistivity distribution beneath the surface. In this method, inversion is the primary process used to estimate the resistivity structure from field data. This study proposes a deep learning-based approach for one-dimensional magnetotelluric inversion, combining Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) as an alternative inversion method. The dataset consists of 20 layers with resistivity and thickness values randomly selected within a specific range at 4000 meters depths and a probability value. Apparent resistivity and phase are obtained through forward modeling based on selected resistivity and thicknesses as input, while the resistivity structure is used as output, with a large data sample. The dataset was standardized and normalized using a logarithmic scale and the MinMax method to map values into the 0-1 range. The dataset was used to train the proposed CNN-GRU model, which is capable of mapping the resistivity distribution in the subsurface. The results show that the CNN-GRU model could map the resistivity distribution model and predict its thicknesses with small error based on the apparent resistivity and phase data, indicating that it can be used for one-dimensional inversion in magnetotellurics. Nevertheless, the model performed quite well on several field datasets, showing a good fit between predicted and true values.

**Keywords**: one-dimensional inversion, magnetotellurics, convolutional neural network, gated recurrent unit

## INTRODUCTION

Magnetotelluric (MT) is a geophysics method that utilizes passive electromagnetic (EM) sources, involving the measurement of fluctuations in the natural electric and magnetic fields in orthogonal directions into the earth to determine the conductivity structure of the earth from a few tens of meters to several hundreds of kilometers [1]. In general, this method was used for volcanoes analysis [2, 3] geothermal exploration [4-6], fault detection [7-13], hydrocarbon exploration[14, 15], etc. In its application, magnetotelluric data analysis involves two primary modeling techniques, forward modeling and inversion.

Forward modelling is the process of calculating the model response that theoretically would be observed at the Earth's surface based on the set of model parameters in the subsurface. In contrast, inversion is calculating model parameters of subsurfaces based on the observed model data. This process will give good accuracy if the response models from the model parameters fit the observed model data[16]. In magnetotellurics implementation, model response or the observed model refers to apparent resistivity ($\rho_a$), phase ($\phi$), and frequencies. Whilst model parameters refer to the resistivity value and the thicknesses of each layer in subsurfaces.

The development and application of Deep Learning as part of Machine Learning has grown rapidly in various fields. Deep Learning has been widely used for many systems such as image processing/ computer vision, Natural Language Processing (NLP), data regression, data prediction, speech recognition, etc [17]. The fundamental and commonly used in deep learning is Artificial Neural Network (ANN), which is inspired by and mimics how the human brains work [18]. Furthermore, ANN is modified by researchers to be another type of neural network, namely Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). The Recurrent Neural Network itself has two types that are currently developed: Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) [19].

In addition, deep learning has been increasingly applied to various geophysical problems. For instance, Tao Ren et al. [20] employed a Convolutional Neural Network (CNN) to estimate seismic severity for earthquake early warning. Huang et al. [21] implemented a Deep CNN with a Gramian Angular Field to reconstruct subsurface resistivity from Airborne Electromagnetic data, while Liurong Tao et al. [22] enhanced seismic impedance inversion using a Fully Convolutional Network (FCN). In magnetotellurics (MT), Khatami and Grandis [23] developed a one-dimensional inversion using CNN, and Xiaolong Liao et al. [24] applied a CNN-LSTM architecture for one-dimensional MT inversion. Collectively, these studies demonstrate that deep learning provides promising solutions for a wide range of geophysical applications.

In this research, a combination of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) hybrid network was applied for one-dimensional magnetotelluric (1D MT) inversion. The CNN was used to extract spatial features from the input data, while the GRU captured sequential dependencies to support the prediction of inversion results. Compared to LSTM, GRU is simpler and more effincient, offering a faster yet effective alternative to previous CNN or CNN-LSTM-based approaches [23, 24].

## Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of Artificial Neural Network. This algorithm was categorized as supervised learning. This algorithm processes one-dimensional data or two-dimensional data. In addition, this algorithm is named Convolutional Neural Network because it applies a convolutional operation on its input, as illustrated in FIGURE 1. CNN is comprised of three main parts, which are the Convolutional Layer, Pooling Layer, and Fully Connected Layer [25, 26].

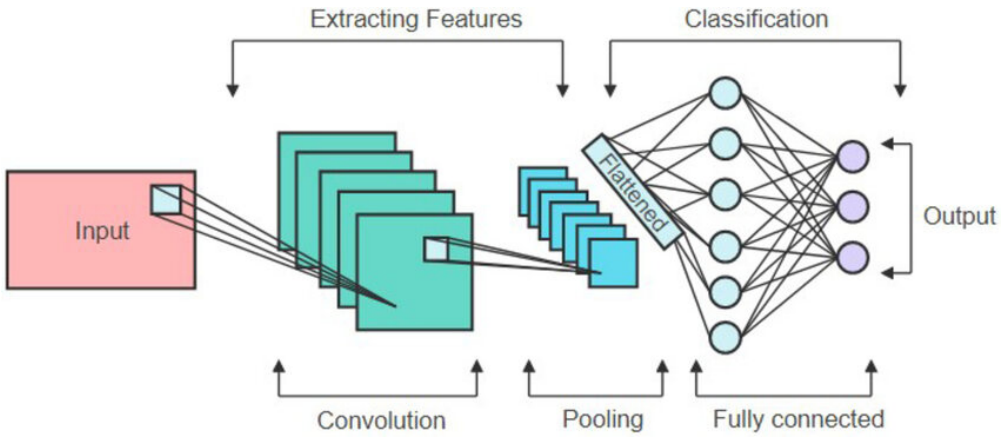


**FIGURE 1.** The illustration of basic Convolutional Neural Network architecture [27].

### ***Convolutional Layer***

The convolutional layers are responsible for performing feature extraction tasks. They carry out this process by applying convolution operations to the input data, and the resulting output is passed to the next layer. Briefly, this operation involves sliding (or shifting) a small filter called a kernel across the entire input and performing a dot product between the kernel values and the input values at each location. These kernel values are learned during training, starting from random initialization. The convolution operation is influenced by parameters such as the number of filters, kernel size, padding, and stride. After applying an activation function such as ReLU, the output becomes a feature map [26].

### ***Pooling Layer***

The pooling layer is the part of a Convolutional Neural Network (CNN) that is designed to reduce the size and dimensionality of the feature map [27]. The output from the pooling layer represents the feature map that will be fed into the next layer. Standard pooling methods for finding the representative of feature maps include MaxPooling, which selects the largest value, and Average pooling, which calculates the average value of the feature map. The benefits of the pooling layer include speeding up computational times, making the training process faster, and enabling robust feature detection.

***Fully Connected Layer***

The last part of a Convolutional Neural Network (CNN) is the Fully Connected layer, similar to the layer found in an Artificial Neural Network (ANN). This layer processes the complex data resulting from the convolutional and pooling layers. Before being passed into the Fully Connected Layer, the feature maps will be flattened into a one-dimensional matrix. The data will be learned and used for a fully connected layer for predicting or classifying data output. In addition, the activation function is used to determine the output of the neural network based on the dataset.

**Gated Recurrent Unit**

Gated Recurrent Unit (GRU) is one of the modified results of Recurrent Neural Network (RNN). GRU consists of two gates in each neuron: update gates and reset gates, as depicted in FIGURE 2.

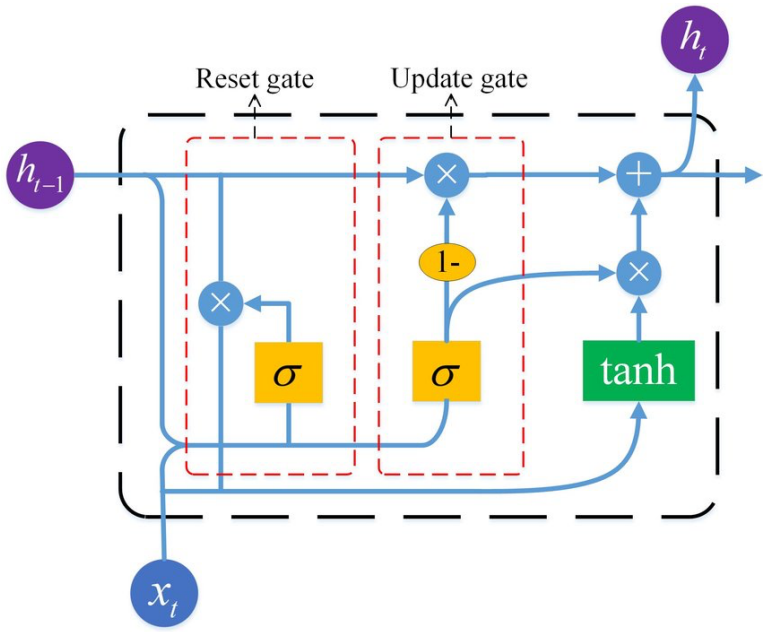


**FIGURE 2.** The illustration of arrangement and position of Update Gate and Reset Gate in Gated Recurrent Unit [28].

When input data enters a Gated Recurrent Unit (GRU), it is first processed by the update gate. The update gate determines how much information from the previous hidden state should be kept and passed to the next layer, using the sigmoid ($\sigma$) activation function. In parallel, the data is also processed by the reset gate, which decides how much of the previous hidden state should be forgotten. The result from the reset gate is then used to calculate the candidate state, using the tanh activation function. Finally, the GRU combines the previous hidden state and the candidate state to produce the new hidden state at the time step $t$, which is passed to the next time step or the next layer. The update gate $z_t$, reset gate $r_t$, candidate hidden state $\tilde{h}_t$, and final hidden state $h_t$ at time $t$, can be calculated using the following equations [28, 29]:

$$z_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b_u)\,, \quad (1)$$

$$r_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b_r)\,, \quad (2)$$

$$\tilde{h}_t = \tanh(W_h \cdot (r_t \odot h_{t-1} + W_h \cdot x_t + b_t)\,, \quad (3)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t\,, \quad (4)$$

where $h_{t-1}$ is the hidden state at the previous time; $x_t$ is the input at time $t$; $W_h$, $W_x$ represents weight matrices; $b_u$, $b_r$, and $b_t$ denotes the update, reset, and candidate state bias matrices, respectively.

## METHODS

### Forward Modelling

In Magnetotellurics, the resistivity model only varies with depth, which could be represented by a horizontal layer with model parameters. Forward modeling calculations for obtaining the one-dimensional model response can be performed using a recursive formula that relates the impedance in the two consecutive layers, as shown in the following equations [30].

$$Z_j = Z_{0j}\frac{1 - R_j \exp(-2k_j h_j)}{1 + R_j \exp(-2k_j h_j)}, \quad (5)$$

$$R_j = \frac{Z_{0j} - Z_{j+1}}{Z_{0j} + Z_{j+1}}, \quad (6)$$

$$Z_{0j} = \sqrt{i\omega\mu_0\rho_j}\,, \quad (7)$$

$$k_j = \sqrt{\frac{i\omega\mu_0}{\rho_j}}, \quad (8)$$

where $\rho_j$ is the true resistivity of the $j^{th}$ layer, $h_j$ is thickness of the $j^{th}$ layer, $\omega$ is the angular frequency, $\mu_0$ is magnetic permeability in free spaces ($4\pi \times 10^7\ H/m$), $R_j$ denotes reflectance coefficient of the $j^{th}$ layer, $Z_{0j}$ denotes intrinsic impedances of the $j^{th}$ layer, and $Z_j$ denotes the impedances of the $j^{th}$ layer.

From these equations, the model response, apparent resistivity, and phase value for every frequency can be obtained from the impedance on the first layer ($Z_1$) using the following equations:

$$\rho_a = \frac{1}{\omega\mu_0}|Z_1|^2\,, \quad (9)$$

$$\phi = \tan^{-1}\frac{Im\ Z_1}{Re\ Z_1}. \quad (10)$$

### Dataset Generation

A dataset was generated to train CNN-GRU model to learn the relationship between input and output data. Forward Modelling was used to create input data (apparent resistivity and phase) and output data (true resistivity). The dataset consists of 1,000,000 sample that consists of 20 layers, a total a total depth of 4000 m, and 69 frequencies ranging from $10^{-4}$ to $10^{4}$ Hz. True resistivity values range from 1-1000 Ω m. And true resistivity was taken from resistivity samples which is one-dimensional matrix from 1-1000 Ω m, and layer variations are controlled by random probability values between 0.4-1 [24].

The dataset was generated by looping through random probability and resistivity values. Layers retain the previous resistivity if the probability is outside the range; otherwise, a new value is taken randomly from resistivity samples matrix. Forward modelling then computes apparent resistivities and phase. The dataset was saved as CSV files, it was spliited into 800,000 data training and 200,000 validation samples, and scaled using logarithmic transformation and MinMax normalization to 0-1. The final input consists of 69-element apparent resistivity and phase, and the output consists of 20-element true resistivity values.

### CNN-GRU Architecture, Parameter Model, and Training Process

The machine learning model used in this research combines a Convolutional Neural Network (CNN) with a Gated Recurrent Unit (GRU). CNN was used to extract spatial features, and GRU was used to extract sequential features. The architecture of the model was built using TensorFlow, a library provided by the Python programming language, and Google Colab. The architecture of the CNN-GRU that is used is illustrated in FIGURE 3. It used a functional API in TensorFlow that can create multiple outputs and a single input flexibly. In addition, the network parameter that are explicitly used is shown in TABLE 1.

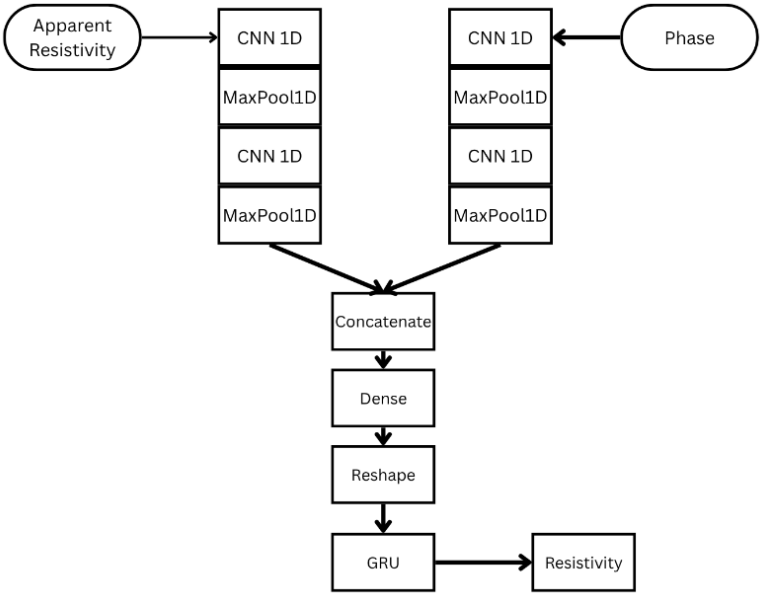


**FIGURE 3.** Illustration of CNN-GRU models, their inputs and outputs.

**TABLE 1.** The Parameters for CNN-GRU Model.

| Layer | Filter/Unit | Kernel Size/ Pool Size | Stride | Activation Function | Output Form |
|---|---|---|---|---|---|
| Conv1D_1 | 32 | 3 | 1 | relu | (69, 32) |
| Conv1D_2 | 64 | 3 | 1 | relu | (69, 64) |
| MaxPooling1D | | 2 | 2 | - | (34, 64) |
| Conv1D_3 | 128 | 3 | 1 | relu | (34, 128) |
| Conv1D_4 | 256 | 3 | 1 | relu | (34, 256) |
| MaxPooling1D_2 | - | 2 | 2 | - | (17, 256) |
| Flatten | - | - | - | - | (None,4352) |
| Conv1D_5 | 32 | 3 | 1 | relu | (69, 32) |
| Conv1D_6 | 64 | 3 | 1 | relu | (69, 64) |
| MaxPooling1D_3 | | 2 | 2 | - | (34, 64) |
| Conv1D_7 | 128 | 3 | 1 | relu | (34, 128) |
| Conv1D_8 | 256 | 3 | 1 | relu | (34, 256) |
| MaxPooling1D_4 | - | 2 | 2 | - | (17, 256) |
| Flatten | - | - | - | - | (None, 4352) |
| Concatenate | - | - | - | - | (None, 8704) |
| Dense | 256 | - | - | relu | (None, 256) |
| Dense_1 | 128 | - | - | relu | (None,128) |
| reshape | - | - | - | 0 | (1,128) |
| GRU | 64 | - | - | tanh | (1,64) |
| GRU_1 | 64 | - | - | tanh | (1,64) |
| GRU_2 | 32 | - | - | tanh | (1,32) |
| GRU_3 | 32 | - | - | tanh | (None,32) |
| Dense_2 | 20 | - | - | linear | (None, 20) |

The training process aims to successfully generate a machine learning model for one-dimensional inversion. It was done in Google Colab cloud system and using a CPU with high RAM. Optimizer used is Adam algorithm with a learning rate of 0.0001. The loss function used was Mean Squared Error. The training process was set for 200 epochs. To avoid overfitting, Early Stopping was used for 15 consecutive epochs with the loss function as the parameter. This process also loaded the best model with the lowest error using ModelCheckpoint for every epoch.

## RESULTS AND DISCUSSIONS

### Inversion Result for Synthetic Data

After the training process was done, the machine learning model was used to test several synthetic datasets that comprise six variations, respectively, with a homogeneous half-space model, a two-layer model, a three-layer model, and a four-layer model. Data training and data validation were also used to ensure the model's capability to predict resistivity in subsurfaces. In this inversion, Mean Absolute Percentage Error, $E_{MAPE}$ , was used to calculate the error between the true data and the predicted data. FIGURE 4 shows several results obtained from data training, data validation, and synthetic datasets.

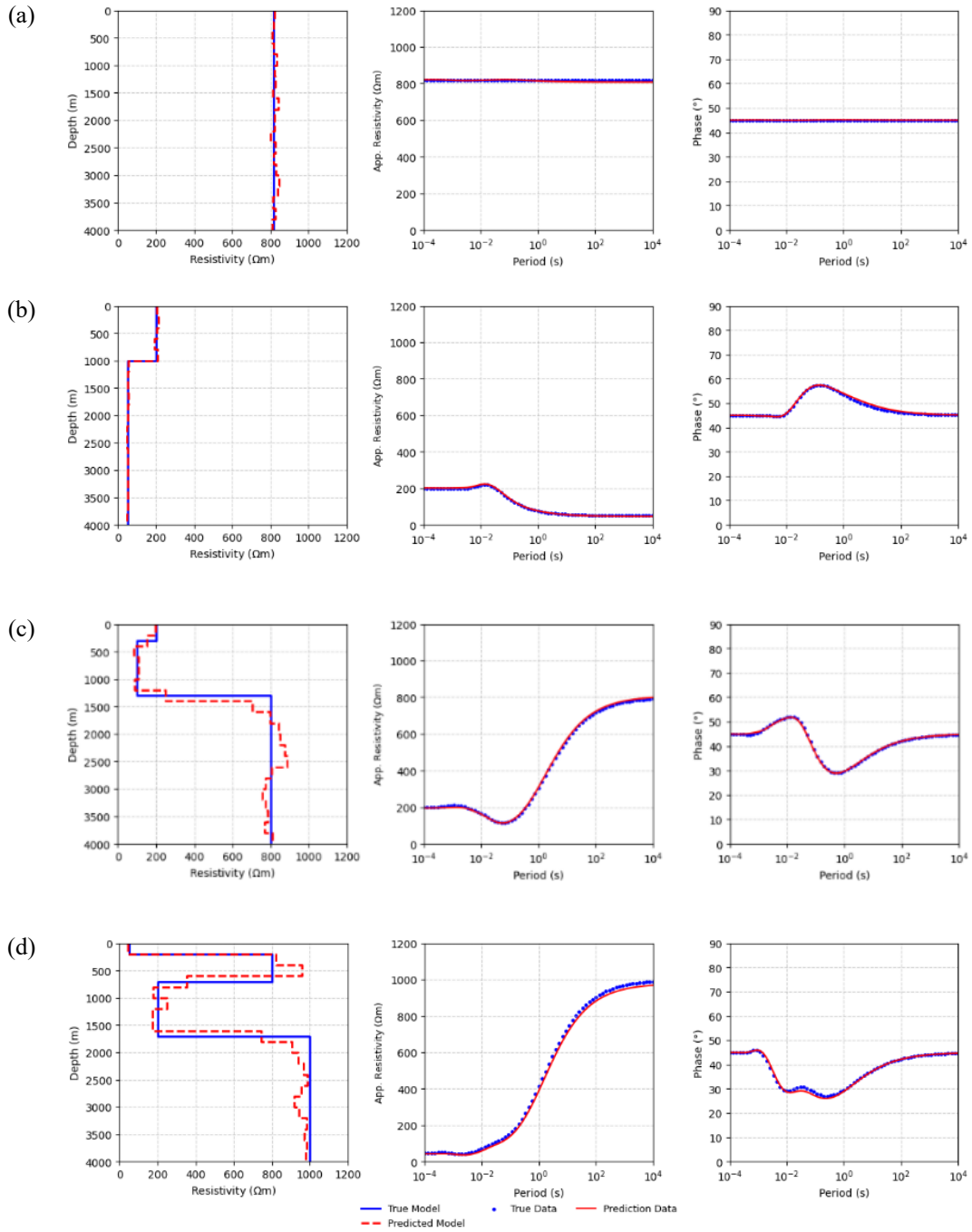


**FIGURE 4.** The Prediction Inversion Result of (a) Homogeneous Half-Spaces Model in Data Training, (b) Two-layer Model in Data Validation, (c) Three-layer Model from Synthetic Data, and (d) Four-layer Model from Synthetic Data.

Based on the results, the prediction of the inversion model by the CNN-GRU model shows quite well compared with the true model, indicating good performance. It was demonstrated that prediction models tend to fit the true model. Nevertheless, the predicted model still shows some anomalies with true model. The results of apparent resistivity and phase indicate that the true data aligns with the predicted data. The computational time needed for predicting data by the model is about 0.6-0.8 s, which means it has a fast computational time. The value of $E_{MAPE}$,

which represents the difference between the true and predicted data, is also low—ranging from 0.38% to 7.7% for apparent resistivity and from 0.11% to 1.33% for phase. It means that the predicted model response from the model prediction is fitting with the true model response. The value of $E_{MAPE}$ for the resistivity model is about 1.28% - 19.02%.

## Inversion Result for Field Data

The model of CNN-GRU that had been trained before was also implemented to predict the inversion result for field data in Aceh, Indonesia. Nevertheless, the CNN-GRU must be retrained with an appropriate frequency value based on field data and given a Gaussian error 5%. Several datasets show quite well results in several fields of data, as shown in the FIGURE 5.

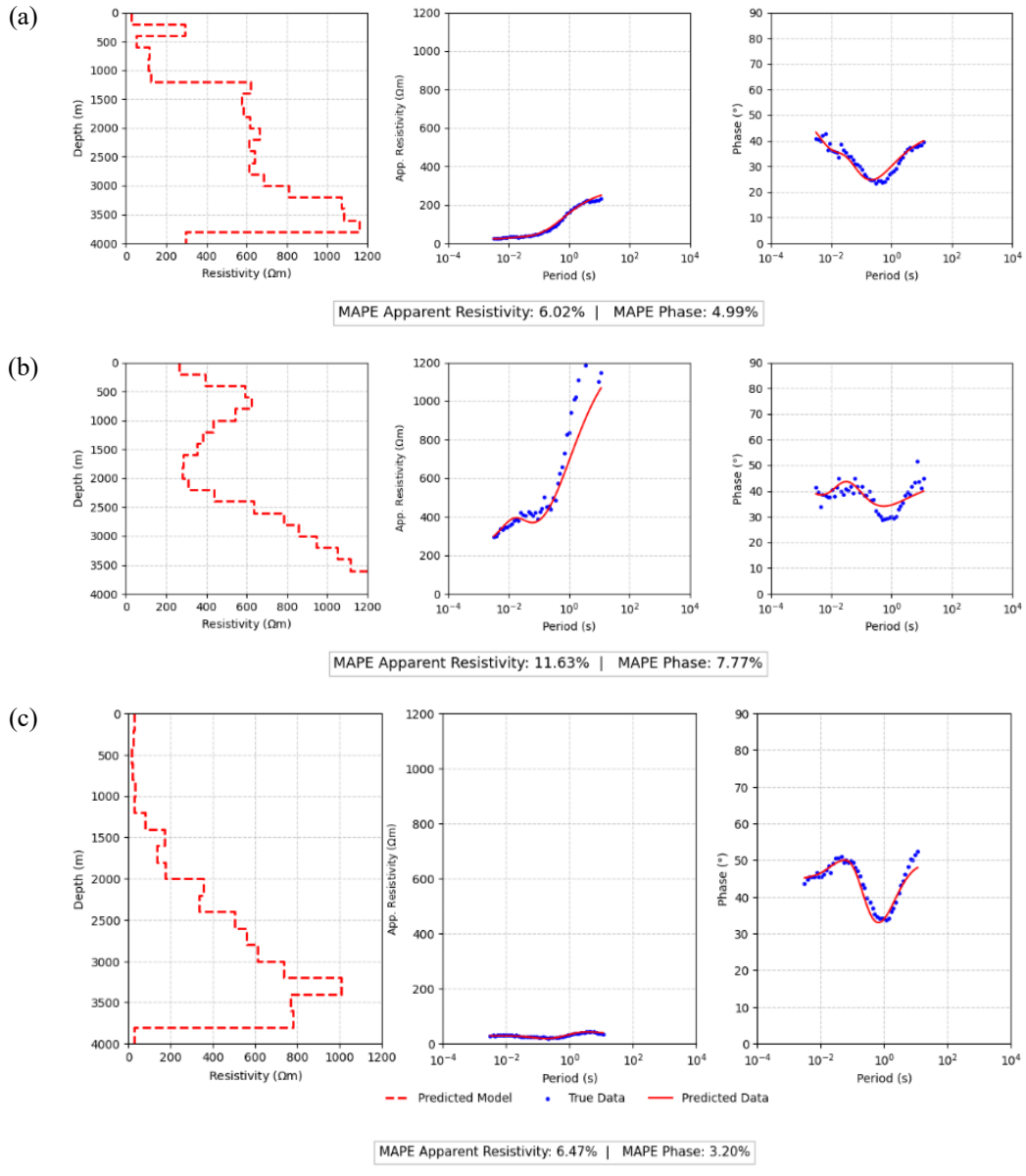


**FIGURE 5.** The Prediction of Inversion Result on Field Magnetotellurics Data in (a) Aceh 01, (b) Aceh 04, and (c) Aceh 05.

Although several figures demonstrate that the predicted apparent resistivity and phase closely match the true values, some predictions still exhibit notable discrepancies. These anomalies indicate that the CNN-GRU model does not always generalize perfectly across all datasets. Nevertheless, the model is still considered reasonably adequate for predicting inversion results, as it successfully fits the resistivity and phase data in many cases.

## CONCLUSION

Based on the experimental results, the CNN-GRU model has proven effective for one-dimensional magnetotelluric (MT) inversion. Data synthetic prediction models demonstrated this, with the predicted resistivity model closely matching the true model. The resistivity structure visualization also confirmed that the predicted data fit quite well with the true data, although it still has some anomalies. Low Mean Absolute Percentage Error supported these values for both apparent resistivity and phase as model responses. For details, the value of $E_{MAPE}$, which represents the difference between true and predicted data was low, ranging from 0.38% to 7.7% for apparent resistivity and 0.11% to 1.33% for phase. It means that the predicted model response from the model prediction is fitting with the true model response. The value of MAPE for the model is about 1.28% - 19.02%. In addition, the CNN-GRU model offered fast computational time for predicting inversion results, making it suitable for efficient modeling. Several variables significantly influenced the model's performance, including the dataset generation method, the number of layers, the number of filters/units, the kernel size, the stride, the activation functions, and the learning rate. The method used to generate the dataset also played an essential role in determining prediction quality.

## REFERENCES

[1] F. Simpson and K. Bahr, "Practical magnetotellurics," *Pract. Magnetotellurics*, vol. 9780521817, pp. 1–254, 2005, doi: 10.1017/CBO9780511614095.

[2] Nurhasan, D. Sutarno, W. Srigutomo, S. Viridi, and D. Fitriani, "Integrated geophysical measurements for subsurface mapping at Papandayan volcano, Garut, Indonesia (preliminary result)," 2012, pp. 154–157. doi: 10.1063/1.4730710.

[3] D. Diba, Nurhasan, D. Sutarno, and Y. Ogawa, "Subsurface Structure around Mas Crater of Papandayan Volcano based on Magnetotelluric and Geomagnetic Data," *J. Phys. Conf. Ser.*, vol. 1949, no. 1, p. 012013, Jun. 2021, doi: 10.1088/1742-6596/1949/1/012013.

[4] N. Nurhasan *et al.*, "Analysis of Subsurface Resistivity Distribution of the Kelud Volcano Using Magnetotelluric Method," *J. Phys. Conf. Ser.*, vol. 2733, no. 1, p. 012020, Mar. 2024, doi: 10.1088/1742-6596/2733/1/012020.

[5] E. J. Mustopa, Nurhasan, W. Srigutomo, and D. Sutarno, "Resistivtiy Structure in Tangkuban Parahu Area Drived from CSAMT Data," *J. Phys. Conf. Ser.*, vol. 877, p. 012055, Jul. 2017, doi: 10.1088/1742-6596/877/1/012055.

[6] E. J. Mustopa, D. Sutarno, Nurhasan, A. R. Naulia, M. L. Baskoro, and N. S. Asih, "Subsurface structure of Tangkuban Parahu area derived from CSAMT and gravity investigation," *J. Phys. Conf. Ser.*, vol. 1080, p. 012049, Aug. 2018, doi: 10.1088/1742-6596/1080/1/012049.

[7] M. R. Naufal and N. Nurhasan, "2D Magnetotelluric Resistivity Structure Modeling Using Finite Element Method Based on Vector Triangular Grid and Its Application to Lembang Fault MT Data," *J. Phys. Conf. Ser.*, vol. 2866, no. 1, p. 012067, Oct. 2024, doi: 10.1088/1742-6596/2866/1/012067.

[8] Nurhasan *et al.*, "Resistivity Distribution of Lembang Fault Based on Magnetotelluric Data," *J. Phys. Conf. Ser.*, vol. 2734, no. 1, p. 012014, Mar. 2024, doi: 10.1088/1742-6596/2734/1/012014.
[9] L. Y. Roodhiyah *et al.*, "Interpretation of a 3D Magnetotellurics Model of the Aceh and Seulimeum Segments of the Sumatran Fault Zone," *Appl. Sci.*, vol. 14, no. 23, p. 11335, Dec. 2024, doi: 10.3390/app142311335.
[10] Nurhasan *et al.*, "Resistivity structure of Sumatran Fault (Aceh segment) derived from 1-D magnetotelluric modeling," 2012, pp. 150–153. doi: 10.1063/1.4730709.
[11] Nurhasan, N. D. Setyo Risky, A. Pratama, I. Zahra, and Y. Ogawa, "Analysis of subsurface Resistivity Structures in Relation to the Level of Damage Caused by Earthquakes Using the Audio-Magnetotelluric (AMT) Method (Case study: the 2006 Yogyakarta Earthquake)," *J. Phys. Conf. Ser.*, vol. 2866, no. 1, p. 012072, Oct. 2024, doi: 10.1088/1742-6596/2866/1/012072.
[12] A. Susilawati *et al.*, "Resistivity and Density Structure of Limboto Lake—Pentadio, Gorontalo, Indonesia Based on Magnetotelluric and Gravity Data," *Appl. Sci.*, vol. 13, no. 1, p. 644, Jan. 2023, doi: 10.3390/app13010644.
[13] Nurhasan *et al.*, "Dimensionality analysis using phase tensor and induction vector on magnetotelluric data in Tomori Region, Central Sulawesi," 2023, p. 050003. doi: 10.1063/5.0164138.
[14] H. Zhou *et al.*, "Super-resolution magnetotelluric data inversion with seismic texture constraint," *GEOPHYSICS*, vol. 90, no. 3, pp. WA153–WA168, May 2025, doi: 10.1190/geo2024-0113.1.
[15] W. Wei, Z. Hu, Z. Zhang, H. Zhang, S. Xu, and R. Zhang, "Application of MT data in the study of ultra-deep basement structure of Precambrian in Tarim Basin," in *SEG 1st Tarim Ultra-Deep Oil & Gas Exploration Technology Workshop, Korla, China, June 3-5, 2024*, Aug. 2024, pp. 113–116. doi: 10.1190/Ultra-Deep2024-030.1.
[16] H. Grandis, *Introduction to Geophysical Inversion Modeling*. Bandung: Indonesian Association of Geophysicists (HAGI), 2009.
[17] J. Ding, H. Chen, Y. Feng, and T. Hossain, "Applications of Deep Learning Techniques," *Electronics*, vol. 13, no. 17, p. 3354, Aug. 2024, doi: 10.3390/electronics13173354.
[18] E. Grossi and M. Buscema, "Introduction to artificial neural networks," *Eur. J. Gastroenterol. Hepatol.*, vol. 19, no. 12, pp. 1046–1054, Dec. 2007, doi: 10.1097/MEG.0b013e3282f198a0.
[19] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," *Information*, vol. 15, no. 9, p. 517, Aug. 2024, doi: 10.3390/info15090517.
[20] T. Ren *et al.*, "Seismic severity estimation using convolutional neural network for earthquake early warning," *Geophys. J. Int.*, vol. 234, no. 2, pp. 1355–1362, Mar. 2023, doi: 10.1093/gji/ggad137.
[21] J. Huang *et al.*, "Implementation of Deep Convolutional Neural Network to Reconstruct Subsurface Resistivity Using Gramian Angular Field for Airborne Electromagnetic System," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 18, pp. 10374–10387, 2025, doi: 10.1109/JSTARS.2025.3550437.
[22] L. Tao, Z. Gu, and H. Ren, "Improving the Seismic Impedance Inversion by Fully Convolutional Neural Network," *J. Mar. Sci. Eng.*, vol. 13, no. 2, p. 262, Jan. 2025, doi: 10.3390/jmse13020262.
[23] M. I. Khatami and H. Grandis, "One-dimensional magnetotelluric (MT) data inversion modeling using convolutional neural network," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1227, no. 1, p. 012023, Aug. 2023, doi: 10.1088/1755-1315/1227/1/012023.
[24] X. Liao, Z. Zhang, Q. Yan, Z. Shi, K. Xu, and D. Jia, "Inversion of 1-D magnetotelluric data using CNN-LSTM hybrid network," *Arab. J. Geosci.*, vol. 15, no. 17, p. 1430, Sep. 2022, doi: 10.1007/s12517-022-10687-1.
[25] A. Saxena, "An Introduction to Convolutional Neural Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 943–947, 2022, doi: 10.22214/ijraset.2022.47789.

[26] J. Rala Cordeiro, A. Raimundo, O. Postolache, and P. Sebastião, "Neural Architecture Search for 1D CNNs—Different Approaches Tests and Measurements," *Sensors*, vol. 21, no. 23, p. 7990, Nov. 2021, doi: 10.3390/s21237990.
[27] W. N. Ismail, H. A. Alsalamah, M. M. Hassan, and E. Mohamed, "AUTO-HAR: An adaptive human activity recognition framework using an automated CNN architecture design," *Heliyon*, vol. 9, no. 2, p. e13636, Feb. 2023, doi: 10.1016/j.heliyon.2023.e13636.
[28] J. Chen, X. Huang, H. Jiang, and X. Miao, "Low-Cost and Device-Free Human Activity Recognition Based on Hierarchical Learning Model," *Sensors*, vol. 21, no. 7, p. 2359, Mar. 2021, doi: 10.3390/s21072359.
[29] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734. doi: 10.3115/v1/D14-1179.
[30] H. Grandis, "An alternative algorithm for one-dimensional magnetotelluric response calculation," *Comput. Geosci.*, vol. 25, no. 2, pp. 119–125, 1999, doi: 10.1016/S0098-3004(98)00110-1.