# A Neural Network-Based Hybrid Framework for Least-Squares Inversion of Transient Electromagnetic Data

Muhammad Rizwan Asif, Thue S. Bording, Pradip K. Maurya, Bo Zhang, Gianluca Fiandaca, Denys J. Grombacher, Anders V. Christiansen, Esben Auken, and Jakob Juul Larsen, *Senior Member, IEEE*

***Abstract*— Inversion of large-scale time-domain transient electromagnetic (TEM) surveys is computationally expensive and time-consuming. The calculation of partial derivatives for the Jacobian matrix is by far the most computationally intensive task, as this requires calculation of a significant number of forward responses. We propose to accelerate the inversion process by predicting partial derivatives using an artificial neural network. Network training data for resistivity models for a broad range of geological settings are generated by computing partial derivatives as symmetric differences between two forward responses. Given that certain applications have larger tolerances for modeling inaccuracy and varying degrees of flexibility throughout the different phases of interpretation, we present four inversion schemes that provide a tunable balance between computational time and inversion accuracy when modeling TEM datasets. We improve speed and maintain accuracy with a hybrid framework, where the neural network derivatives are used initially and switched to full numerical derivatives in the final iterations. We also present a full neural network solution where neural network forward and derivatives are used throughout the inversion. In a least-squares inversion framework, a speedup factor exceeding 70 is obtained on the calculation of derivatives, and the inversion process is expedited ∼36 times when the full neural network solution is used. Field examples show that the full nonlinear inversion and the hybrid approach gives identical results, whereas the full neural network inversion results in higher deviation but provides a reasonable indication about the overall subsurface geology.**

***Index Terms*— Forward modeling, inverse modeling, Jacobian matrix, neural networks, transient electromagnetics (TEM).**

Manuscript received January 12, 2021; revised March 16, 2021 and April 20, 2021; accepted April 25, 2021. Date of publication May 11, 2021; date of current version December 29, 2021. This work was supported by the Innovation Fund Denmark through the MapField Project. *(Corresponding authors: Muhammad Rizwan Asif; Bo Zhang.)*

Muhammad Rizwan Asif, Thue S. Bording, Pradip K. Maurya, Denys J. Grombacher, and Anders V. Christiansen are with the Department of Geoscience, Aarhus University, 8000 Aarhus, Denmark, and also with the Aarhus University Center for Water Technology (WATEC), Aarhus University, 8000 Aarhus, Denmark (e-mail: rizwanasif@geo.au.dk).

Bo Zhang is with the College of Geo-Exploration Sciences and Technology, Jilin University, Changchun 130026, China (e-mail: em_zhangbo@163.com).

Gianluca Fiandaca is with the Department of Earth Sciences, University of Milan, 20122 Milan, Italy.

Esben Auken is with the Geological Survey of Denmark and Greenland, 1350 Copenhagen, Denmark.

Jakob Juul Larsen is with the Department of Electrical and Computer Engineering, Aarhus University, 8000 Aarhus, Denmark (e-mail: jjl@ece.au.dk).

Digital Object Identifier 10.1109/TGRS.2021.3076121

## I. Introduction

TRANSIENT electromagnetics (TEM) is a well-proven method to map the conductivity structure of the subsurface. Typical applications of TEM include mapping of groundwater resources and mineral exploration [1]. In TEM, a primary magnetic field is established from a coil. When this primary field is turned off, eddy currents are induced and propagates into the Earth. These decaying eddy currents generate a secondary magnetic field from which the subsurface resistivity is inferred through inverse methods. A key component of inversion is the forward problem, represented as in the following equation:

$$\mathbf{d} = F(\mathbf{m}) \tag{1}$$

where $F$ is the forward operator, $\mathbf{m}$ is a vector of model parameters, and $\mathbf{d}$ holds the forward data.

The forward problem computes the data $\mathbf{d}$ given $\mathbf{m}$ and $F$. However, if only $\mathbf{d}$ and $F$ are known and the model $\mathbf{m}$ is to be predicted, this process is called inversion. During the inversion, the aim is to find one or more models $\mathbf{m}$ that minimize the objective function, $\phi(\mathbf{m})$, using the following equation:

$$\phi(\mathbf{m}) = \underbrace{\|\mathbf{Q_d}(\mathbf{d_{obs}} - F(\mathbf{m}))\|_{L_2}^2}_{\text{data misfit}} + \underbrace{\left\|\mathbf{Q_p R_p m}\right\|_{L_2}^2}_{\text{smoothness constaints}} \tag{2}$$

where $F(\mathbf{m})$ is the predicted data, $\mathbf{d_{obs}}$ is the observed data, and $\mathbf{Q_d}$ holds the inverse of the data variance. $\mathbf{Q_p}$ specifies the variability associated with the constraints described by $\mathbf{R_p}$ that is the roughness matrix that calculates the model parameters $\mathbf{m}$ in the neighboring depth layers.

Least-squares inversion aims at producing one optimal model, minimizing $\phi(\mathbf{m})$, by iteratively updating the model parameters $\mathbf{m}$, for instance, through an iterative Levenberg–Marquardt minimization algorithm, presented in the following equation:

$$\begin{aligned}\mathbf{m}_{n+1} = \mathbf{m}_n &+ \left[\mathbf{G}_n^T \mathbf{C}_{\text{obs}}^{-1} \mathbf{G}_n + \mathbf{R}_p^T \mathbf{C}_c^{-1} \mathbf{R}_p + \lambda \mathbf{I}\right]^{-1} \\ &\cdot \left[\mathbf{G}_n^T \mathbf{C}_{\text{obs}}^{-1} (\mathbf{d}_{\text{obs}} - F(\mathbf{m}_n)) + \mathbf{R}_p^T \mathbf{C}_c^{-1} \left(-\mathbf{R}_p \mathbf{m}_n\right)\right]\end{aligned} \tag{3}$$

where $\mathbf{G}_n$ is the Jacobian matrix for the $n$th model parameter of $\mathbf{m}_n$, $\mathbf{C}_{\text{obs}}^{-1} = \mathbf{Q}_d^T \mathbf{Q}_d$ is the covariance matrix holding the data uncertainties, $\mathbf{C}_c^{-1} = \mathbf{Q}_p^T \mathbf{Q}_p$ defines the strength of the smoothness constraints, $\mathbf{I}$ is the identity matrix, and $\lambda$ is the Marquardt damping parameter. For more details regarding (2) and (3), the readers are referred to [2] and [3].

The entries of the Jacobian matrix $\mathbf{G}$ for a model vector $\mathbf{m}$ are the partial derivatives in logarithmic space computed by

$$\mathbf{G}_{\rm st} = \frac{\partial \log(d_s)}{\partial \log(m_t)} = \frac{m_t}{d_s}\frac{\partial d_s}{\partial m_t} \tag{4}$$

for the $s$th parameter in the data vector $\mathbf{d}$ and the $t$th parameter in the model vector $\mathbf{m}$.

The Jacobian matrix $\mathbf{G}$ can also be written as

$$\mathbf{G}_t = \frac{m_t}{\mathbf{d}}\frac{\partial \mathbf{d}}{\partial m_t}. \tag{5}$$

Most of the computation time in a least-squares algorithm lies within the calculation of $\mathbf{G}$. If the $\mathbf{G}$ matrix holds $M$ parameters, $M+1$ forward calculations are required to populate $\mathbf{G}$. The model parameters $\mathbf{m}$ are continuously updated until the forward response of a model fits the observed data within some specified bounds.

In order to avoid the computationally hefty inversion process, artificial neural networks (ANNs) can be trained to directly determine the resistivity structure of the Earth [4]–[10]. This approach is tempting, but it has several drawbacks, including the limitations in including data noise explicitly, loss of lateral and spatial coherence in the inverted models, and the need for network retraining when changing the geophysical system or any of its settings. For these reasons, we focus on using neural networks to calculate the derivatives to be used in the least-squares inversion framework. A similar approach has been attempted in [11]; however, this study is restricted to a synthetic study of few layer models and requires prior knowledge about the subsurface structures.

Neural networks can be used to predict sufficiently accurate TEM forward responses [12], [13], but the precision of state-of-the-art networks is still insufficient for computing derivatives, because the uncertainty of the predicted response is random and not systematically biased. Therefore, we train and use a separate network for computing derivatives. Our neural network is trained on the derivatives for multilayer models based on impulse responses, and the system transfer function is convolved post-network [14]. This makes it possible to work with almost system-independent neural networks to invert field data. In addition, by keeping the least-squares inversion framework, the coherence between the inverted models can be retained and *a priori* information and/or smoothness constraints can be applied without retraining the network. One disadvantage is that the evaluation of the matrix–matrix, matrix–vector products, and inverse matrix adds a computational overhead [15]. Our neural network is named "jNet," short for "JacobianNet" since it is used directly to compute the Jacobian matrix. We investigate the performance benefits for a towed TEM system [16] and compare several iterative inversion schemes. We examine accuracy and computation time between these.

This article is structured in the following order. In Section II, we define the inputs and outputs for the neural network and discuss the data preprocessing workflows and the network architecture. Inversion results on field data are shown in Section III. In Section IV, we discuss the limitations and prospects of the proposed scheme. We give the conclusion in Section V.

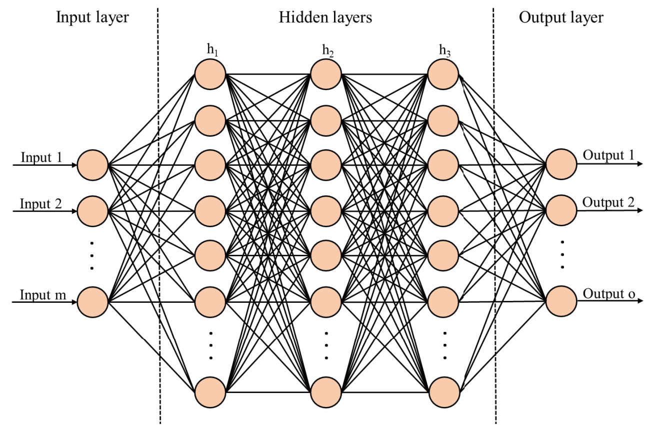



Fig. 1. Typical structure of a three-hidden-layer deep neural network.

## II. Proposed Methodology

ANNs are computing systems that are inspired by biological neural networks aiming to simulate the way that a human brain analyzes and processes information. ANNs have self-learning capabilities to perform specific tasks by considering a set of examples [17]. These systems are composed of processing units called artificial neurons, aiming to loosely model the biological brain. Generally, neurons are organized in layers, and the neurons in one layer are connected to the neurons of neighboring layers. The connections between neurons are called weights and biases. A typical structure of ANN is shown in Fig. 1.

An iterative formula (6) is used to move forward through the network to calculate each neuron in the next layer, called as a forward pass

$$f_i^l = \sigma\left(\sum_j w_{ij}^l f_j^{l-1} + b_i^l\right) \tag{6}$$

where $w_{ij}^l$ is the weight for the connection from the $j$th neuron in the $(l-1)$th layer to the $i$th neuron in $l$th layer, $b_i^l$ represents the bias for the $i$th neuron in The $l$th layer, and $f_i^l$ is the activation of the $i$th neuron in the $l$th layer. Finally, $\sigma$ is the activation function that adds nonlinearity to the system.

The expression (6) can be rewritten in a matrix form as in the following equations:

$$\mathbf{z}^l = \left(\mathbf{w}^l\mathbf{f}^{l-1} + \mathbf{b}^l\right) \tag{7}$$
$$\mathbf{f}^l = \sigma\left(\mathbf{z}^l\right). \tag{8}$$

Given the final output of (8) depending on the number of layers $l$, the weights and the biases are optimized for a cost function $C$. This process is called a backward pass. To update the network weights $\mathbf{w}$ and biases $\mathbf{b}$, a backward propagation algorithm [18] based on (9)–(11) is used

$$\frac{\partial \mathbf{C}}{\partial \mathbf{w}^l} = \frac{\partial \mathbf{C}}{\partial \mathbf{f}^l}\frac{\partial \mathbf{f}^l}{\partial \mathbf{z}^l}\frac{\partial \mathbf{z}^l}{\partial \mathbf{w}^l} \tag{9}$$

$$\frac{\partial \mathbf{C}}{\partial \mathbf{b}^l} = \frac{\partial \mathbf{C}}{\partial \mathbf{f}^l}\frac{\partial \mathbf{f}^l}{\partial \mathbf{z}^l}\frac{\partial \mathbf{z}^l}{\partial \mathbf{b}^l} \tag{10}$$

$$\frac{\partial \mathbf{C}}{\partial \mathbf{f}^{l-1}} = \frac{\partial \mathbf{C}}{\partial \mathbf{f}^l}\frac{\partial \mathbf{f}^l}{\partial \mathbf{z}^l}\frac{\partial \mathbf{z}^l}{\partial \mathbf{f}^{l-1}}. \tag{11}$$

The weights and biases are then updated as given in the following equations:

$$\mathbf{w}^l = \mathbf{w}^l - \alpha\frac{\partial \mathbf{C}}{\partial \mathbf{w}^l} \tag{12}$$

$$\mathbf{b}^l = \mathbf{b}^l - \alpha\frac{\partial \mathbf{C}}{\partial \mathbf{b}^l} \tag{13}$$

where $\alpha$ is the learning rate. For detailed understanding of ANNs, the readers are referred to [19]–[22].

There are several factors that need to be considered before training a neural network. First, it is important to specify the inputs and outputs. Second, to achieve maximum accuracy performance from a network, it is essential to normalize the data appropriately. Finally, it is important to select the optimal network configuration.

### A. Neural Network Inputs and Outputs

The jNet input consists of 32 variables. The first 30 input variables correspond to the 1-D subsurface resistivity model $\mathbf{m}$ with log-increasing thicknesses, a top layer thickness of 1 m, and a depth to last layer boundary at 120 m. The 120 m corresponds to the absolute maximum depth of investigation (DOI) for the particular TEM system. The layer thicknesses are fixed and are not considered as input parameters. Since our TEM instrument uses offset transmitter and receiver coils, the distance $z$ between these is also an input parameter. The final input variable is the index $t$ in the model vector $\mathbf{m}$ for which the partial derivative will be calculated as (5).

The network output is the partial derivative $\partial \mathbf{d}/\partial m_t$ for the corresponding inputs. We do not include $m_t/\mathbf{d}$ in the network output since $\mathbf{d}$ and $\mathbf{m}$ are constants in (5). The partial derivatives for network training are computed as symmetric differences of full nonlinear forward responses using $\pm 2\%$ perturbations with respect to the $t$th model parameter. All other parameters are held fixed. The responses are generated at 83 discrete time gates from 50 to 37 ms with exponentially increasing gate widths sampled at 14 gates/decade. Although the actual TEM data span a narrower time interval, a wider range is considered to obtain accurate response after convolution with the system response.

The derivative values at several early time gates are zero and contain limited information [see Fig. 2(a) and (b)]. The useful information begins with the first nonzero time gate, and the position of the first nonzero time gate moves toward later times with the perturbation at deeper layers, as shown in Fig. 2(c) and (d). To improve precision in the predicted derivatives, we also include the first nonzero time-gate index $g$ as an output variable. It is used in the postprocessing to force all the entries before the first nonzero time gate in the predicted derivatives to zero.

### B. Data Normalization

It is advantageous to normalize and map the input and output variables to a common scale for faster convergence prior to training. Since the forward response does not vary linearly with resistivity, we consider logarithmic variations in the subsurface models $\mathbf{m}$ that are then normalized between $[a, b]$ using the following equation:

$$\mathbf{m}_N = a + \frac{(b-a)\left(\log_{10}(\mathbf{m}) - \log_{10}(m_{\min})\right)}{\log_{10}(m_{\max}) - \log_{10}(m_{\min})} \tag{14}$$

where $\mathbf{m}_N$ is the normalized resistivity model of $\mathbf{m}$ and $m_{\min}$ and $m_{\max}$ denote the global minimum and maximum resistivity values acquired from the training model space, respectively.

The distance between the transmitter and receiver coils affects the TEM signal. Therefore, it is considered an input parameter and is normalized linearly by the min–max scaling by the following equation:

$$z_N = a + \frac{(b-a)(z - z_{\min})}{z_{\max} - z_{\min}} \tag{15}$$

where $z_N$ and $z$ are the normalized and actual center-to-center distance between the transmitter and receiver coil, respectively, and $z_{\min}$ and $z_{\max}$ are the minimum and maximum possible distances, 7.28 and 10.28 m, respectively.

Similarly, the model parameter index $t$, at which the perturbation is to be applied, is also normalized linearly to map it to a common scale similar to (15), where $t_{\min}$ and $t_{\max}$ are the minimum and maximum indexes in model parameter $\mathbf{m}$, respectively. Since we consider 30-layer resistivity structures, this value is restricted between [1, 30].

Hence, the network input for a common scale of $[-1,1]$ is defined as a $32 \times 1$ vector as in the following equation:

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{m}_N \\ z_N \\ t_N \end{bmatrix}. \tag{16}$$

The performance of neural networks is not only affected by the changes within a data curve but also by the degree of

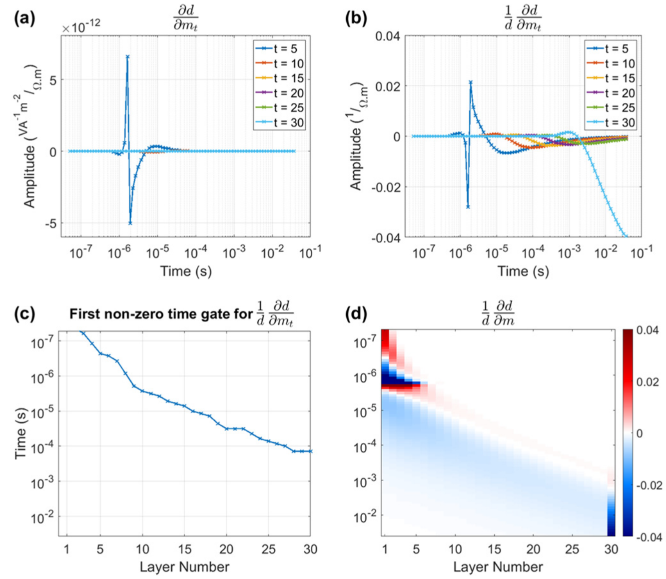



Fig. 2. Visualization of the partial derivatives for a 10-Ω·m half-space model. (a) Partial derivative (∂d/∂m) showing its large dynamic range. (b) ∂d/∂m normalized by the forward response resulting in more variation and reduced dynamic range. (c) First nonzero entry in the sensitivity matrix. (d) Pictorial visualization of the sensitivity matrix computed from (4).

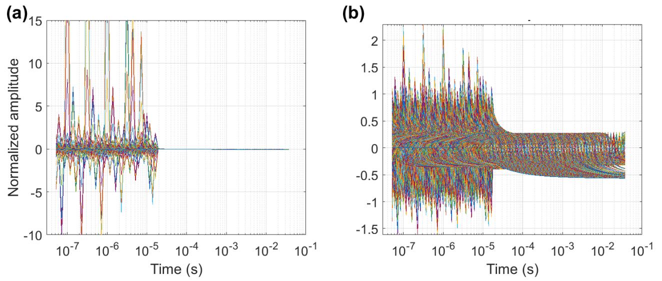



Fig. 3. Transformation of normalized partial derivatives of 700 logarithmically spaced half-space models between 1 and 1000 Ω·m. (a) Normalized partial derivatives. (b) Fifth root of the normalized partial derivatives.

variation between the observed data curves [12]. Therefore, instead of considering the partial derivatives directly, which have insignificant variation, i.e., the imperceptible difference between six curves [see Fig. 2(a)], we first normalize it by the forward response $\mathbf{d}$ of the unperturbed model to impart larger variation and reduced dynamic range [see Fig. 2(b)]. To have additional variation and to map to a common scale, we transform the normalized partial derivatives as in the following equation:

$$\mathbf{Y}_t = \sqrt[5]{\left(\frac{1}{\mathbf{d}}\frac{\partial \mathbf{d}}{\partial m_t}\right)}. \tag{17}$$

Root scaling effectively decreases the parameter span, and only considering odd roots ensures that the sign of the original data is kept. The output for any odd root results in a similar pattern. Heuristically, we apply the fifth root. Fig. 3 shows that the fifth root of the normalized partial derivatives results in significantly larger data variation than the normalized partial derivatives for 700 logarithmically spaced half-space models. This transformation results in better variation than the typically employed z-score method in neural networks.

Finally, the first nonzero time-gate index $g$ is also scaled to a common scale similar to (15). In this context, $g_N$ is the normalized index value and $g_{\min}$ and $g_{\max}$ are the minimum and maximum first nonzero gate-time indexes in the training set, 1 and 65, respectively.

Hence, the $84 \times 1$ network output stacked vector becomes

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_t \\ g_N \end{bmatrix}. \tag{18}$$

## C. Network Architecture

We use a five-layer fully connected deep neural network, as shown in Fig. 1. The first layer corresponds to the input variables and consists of 32 neurons. The next three layers, i.e., the hidden layers, have 384 neurons each. The final regression layer consists of 84 neurons. In our experience, not much improvement is gained when deeper network architecture is used. In addition, heuristic analysis by a logarithmic grid search of the number of neurons in the hidden layers shows that the selected number of neurons gives the best performance.

As an activation function, we utilize the hyperbolic tangent function since our inputs and outputs can take both positive and negative values. The hyperbolic tangent function is continuous, differentiable, and zero centered, which improves the modeling of strongly negative/positive and neutral values.

The Nguyen–Widrow initialization algorithm [23] is used to initialize the weights and biases of the network. The cost function used for training is given in (19), which consists of the sum-of-squares of the network errors with a regularization term that consists of the mean of the sum-of-squares of the network weights and biases. The regularization term improves generalization and makes the network less susceptible to overfitting

$$C = \gamma \underbrace{\sum_{i=1}^{N} (\mathbf{Y}_i - \mathbf{Y}_i)^2}_{\text{network errors}} + (1-\gamma)\underbrace{\left(\frac{1}{n}\sum_{j=1}^{n}\mathbf{w}_j^2 + \frac{1}{n}\sum_{j=1}^{n}\mathbf{b}_j^2\right)}_{\text{regularization term}} \tag{19}$$

where $\gamma$ is the performance ratio set to 0.99 in our case.

The scaled conjugate gradient backpropagation algorithm [24] is used to update the weights and biases to minimize the cost function. We use the full-batch algorithm to avoid the tuning of an additional parameter, i.e., the mini-batch size, as its performance in comparison to the mini-batch algorithm with different batch sizes is similar [25]. This tradeoff is achieved at the cost of more training time.

We apply an early stopping criterion to ensure that training is stopped when validation loss starts to increase, while the training loss may still be decreasing. We choose the check count to be 30 000 epochs, which means that the training stops if the validation loss does not go lower than the best validation performance for the succeeding 30 000 iterations.

# III. Results

For a solution that generalizes to broad geological settings, it is essential that the resistivity models used for the training of the neural network cover the entire model space. One way to achieve this is to generate random models. However, random models will hold many nonresolvable resistivity structures and variations for the tTEM method and therefore not directly suitable for the network training. Hence, it is favorable to restrict the model space to geophysically resolvable models. Therefore, we generate forward responses of 0.6 million models where the resistivity of each layer is chosen uniformly between 1 and 1000 Ω·m. The forward responses of these models are inverted using a standard least-squares inversion algorithm [26] to obtain geologically plausible and geophysically resolvable models.

Fig. 4(a) shows two examples of inverting random models into geophysically resolvable models. For the first scenario, the forward data of a random model with low resistive layers (represented by a thin blue line) are inverted and the shallow layers can be resolved (presented as a thick blue line). The variation in deeper layers is not resolved due to the limited DOI of tTEM system for such low resistive shallow layers [16], [27]. For the second scenario where the forward

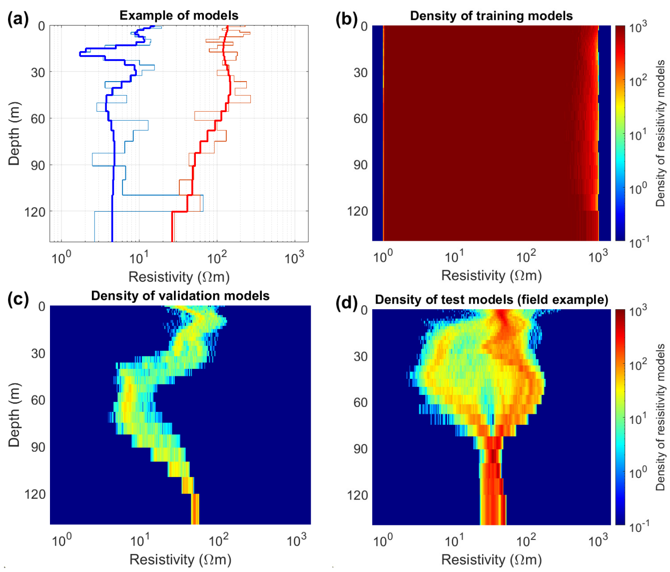



Fig. 4. Resistivity models for the training and testing of the network. (a) Example of two random and reinverted geophysically resolvable models (thick line represents reinverted models). (b) Density of the training set. (c) Density of the validation set. (d) Density of the field example models.

data of a model with relatively high resistive layers (shown in thin red line) are inverted, the overall larger variations in the model are recovered. Smaller variations are not captured in the inverted model due to limited sensitivity and equally limited resolving capability of TEM method toward thin resistive layers. Even if an actual subsurface model is represented by a high resistive model, it is expected that any TEM method would have difficulty in resolving such a model. Therefore, it is logical to use a set of training models that are geophysically resolvable. The comprehensive training model space of realistic subsurface patterns that are geophysically resolvable is shown in Fig. 4(b). It takes ~4 days to generate the training model space and ~6 h to produce the corresponding forward responses and partial derivatives. However, this process needs to be done only once for a given TEM system. We generate symmetric derivatives for 30-layered models, and the forward solver is called 61 times for each model to constitute the **G** matrix given in (5).

The validation set that is used for the early stopping criterion is comprised of 697 resistivity models obtained from the data of a survey conducted at Søften, Denmark [see Fig. 4(c)]. Since the perturbation is required for each of the 30 layers, a total of 20 910 samples are used. It takes ~3 h for the early stopping criterion to take effect and conclude the network training on two NVIDIA GeForce RTX 2080Ti GPUs. Once trained, it can be put into practice for faster inversion of any 2-D or 3-D volume tTEM datasets.

The output of our network is postprocessed to transform back to raw values as

$$\frac{\partial \widehat{\mathbf{d}}}{\partial \widehat{m}_t} = \left(\widehat{\mathbf{Y}}_t \cdot \mathbf{d}\right)^5 \tag{20}$$

and

$$\widehat{g} = \frac{\left(\widehat{g}_N - a\right)(g_{\max} - g_{\min})}{b - a} + g_{\min}. \tag{21}$$

Then,

$$\frac{\partial \widehat{\mathbf{d}}}{\partial \widehat{m}_t} = \begin{cases} \partial \widehat{d}_g \big/ \partial \widehat{m}_t = 0, & \text{for } g \leq \text{ceil}(\widehat{g}) \\ \partial \widehat{d}_g \big/ \partial \widehat{m}_t & \text{otherwise.} \end{cases} \tag{22}$$

To test the performance of jNet, we use field data from another survey from Gedved, Denmark. The data are initially inverted using the full nonlinear solution to obtain 5978 resistivity models, as shown in Fig. 4(d). These models are used to generate the partial derivatives from the full solution to compare it with the output of jNet. Since the direction of the model update depends on the sign of the partial derivatives in the Jacobian matrix, we use the sign as the primary metric for performance evaluation. The sign accuracy of jNet on the test data is over 98%. It is to note here that the errors in jNet will affect the inversion as any other inaccuracy would. We also show a visual comparison of the normalized partial derivatives constructed by a full nonlinear solution, and jNet for a starting [see Fig. 5(a)–(d)], intermediate [see Fig. 5(e)–(j)], and the final model [see Fig. 5(i)–(l)] shows a similar pattern and magnitude. Therefore, we expect the proposed strategy to perform well.

To qualitatively evaluate the performance of jNet, we initially employ three iterative inversion schemes where the full nonlinear forward response (fFull) is used for the evaluation of the objective function, and the difference is only in the calculation of the Jacobian matrix. To have a full neural network solution, we also replace fFull with a neural network forward modeling surrogate (fNet) [13] retrained on the models used for jNet. By this, we invert full 3-D volume datasets collected from two surveys conducted in Gyldenholm and Gedved, two regions in Denmark, consisting of 11 601 and 5978 tTEM soundings, and evaluate the following solutions.

1) *Full Nonlinear Inversion:* fFull+jFull.
2) *Hybrid Inversion:* fFull+jNet until a certain misfit is reached, then changing to fFull+jFull.
3) *jNet Inversion:* fFull+jNet.
4) *Full Neural Network Inversion:* fNet+jNet.

Fig. 6 shows a cross section from each of the two surveys where the full inversion (fFull-jFull) serves as a reference (first row of Fig. 6). The hybrid inversion (fFull-jNet/jFull) gives identical results (second row of Fig. 6), whereas the jNet inversion (third row of Fig. 6) results in a minor deviation from the reference result. The full neural network inversion (fNet-jNet) gives sufficiently accurate results for medium-to-high conductivity regions [see Fig. 6(g)]; however, larger divergence is observed for high resistive areas [see Fig. 6(h)]. This is due to the forward responses from fNet having higher error for high resistive (low conductive) models.

For the medium-to-high conductivity regions (first column of Fig. 6 which shows a cross section from the Gyldenholm survey), the hybrid inversion shown in Fig. 6(c) deviates 0.4% in terms of comparison to the reference inversion, which is shown in Fig. 6(a). The jNet inversion presented in Fig. 6 has a mean resistivity difference of 4.3%, while the full neural network of geometric mean of model resistivity differences in inversion as in Fig. 6(g) results in a deviation of 4.9%.

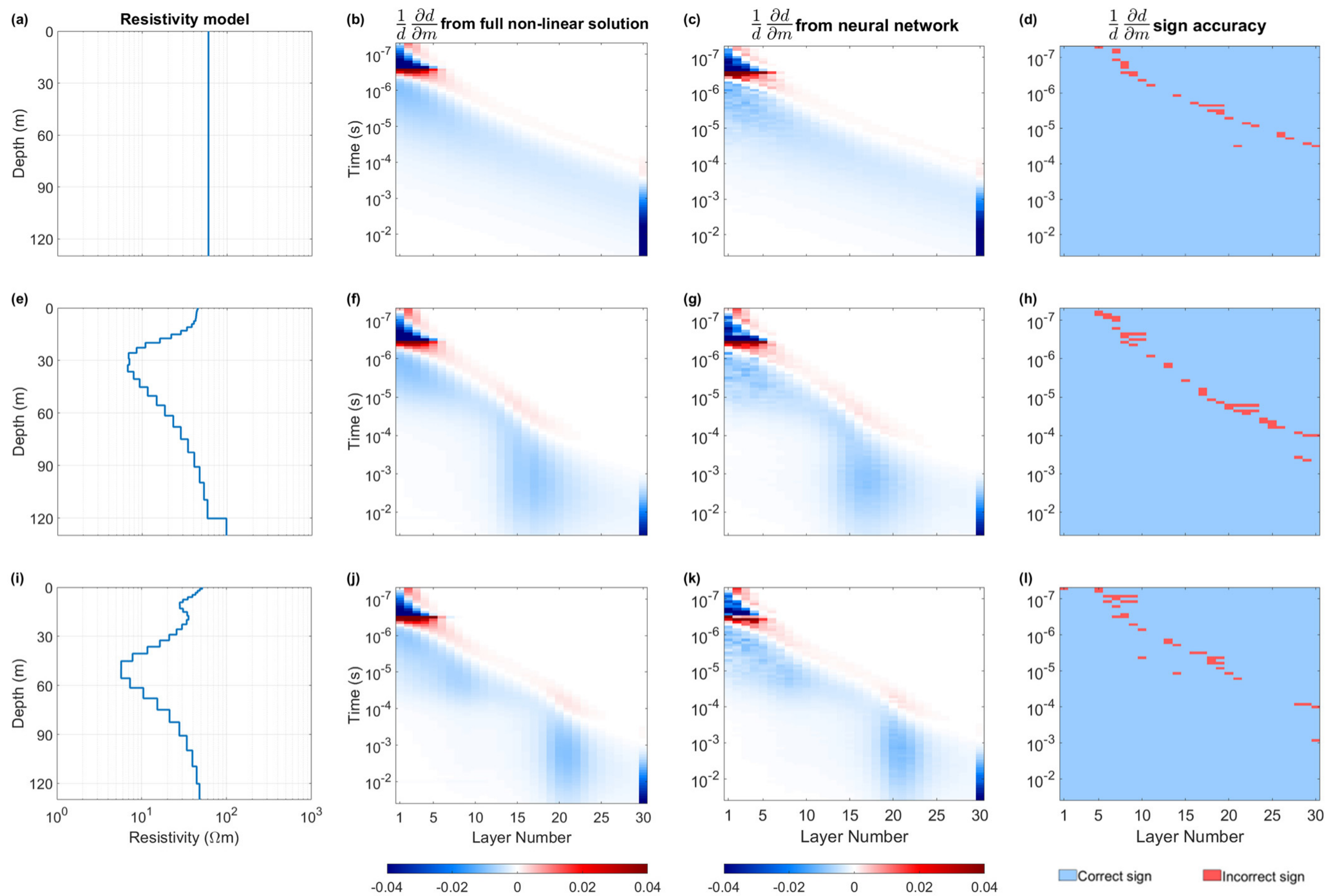



Fig. 5. Visualization of iterative model update and their corresponding partial derivatives. (a), (e), and (i) Resistivity models of first, intermediate, and final iteration for a TEM signal, respectively. (b), (f), and (j) Corresponding partial derivatives from full solution (jFull). (c), (g), and (k) Predicted partial derivatives from jNet. (d), (h), and (l) Sign accuracy between jFull and the jNet derivatives.

For low conductive areas (a section from Gedved survey in the second column of Fig. 6), the hybrid inversion shown in Fig. 6(d) results in a difference of mean of model resistivity of 0.3% compared with the reference inversion presented in Fig. 6(b). The jNet inversion as in Fig. 6(f) has the mean resistivity difference of 12.8% and the full neural network inversion shown in Fig. 6(h) gives a mean resistivity difference of 13.2%.

In Table I, we show a detailed analysis of four inversion schemes in terms of total misfit $\phi(m)$ defined in (2). We also show the time comparison between different inversion schemes. The comparison is made on a 3-D volume dataset of another survey conducted in Sondersoe, Denmark, which consists of 2822 soundings. Table I shows that the hybrid inversion uses an additional iteration while being at least four times faster than the reference inversion. The jNet inversion uses the same number of iterations as the reference inversion and achieves a speedup factor of over 13. Each iteration in the inversion is expedited ~14 times when jNet is used. The full neural network inversion is ~36 times faster than full nonlinear inversion and achieves a speedup of a factor of ~39 per iteration. It also takes the same number of iterations as the full nonlinear inversion. The jNet and full neural network inversions result in slightly higher misfits, which corresponds to errors introduced by fNet and jNet. However, the inaccuracies are compensated in the hybrid inversion framework by additional iterations that uses accurate modeling. On a single CPU core, jNet produces ~1750 derivatives per second in comparison to ~25 derivatives per second by the conventional approach [26]. Therefore, a speedup factor of over 70 is realized when jNet is used for the calculation of partial derivatives. We also note that the system response convolution takes 65% of the time during fNet and jNet calculations.

TABLE I

INVERSION TIME COMPARISON (ONE CPU CORE)

| Inversion Method | # of iterations | Total inversion time (minutes) | Total misfit $\phi(m)$ |
|---|---|---|---|
| fFull-jFull | 6 | 357 | 0.2284 |
| fFull-jNet/jFull | 7 | 84 | 0.2289 |
| fFull-jNet | 6 | 27 | 0.2304 |
| fNet-jNet | 6 | 9 | 0.2322 |

Our fNet and jNet have been trained on models produced by a smooth inversion result in order to limit the training model space to geophysically resolvable models. To investigate whether these networks can also be employed with an alternative regularization scheme, we invert the field data from Gyldenholm and Gedved surveys by using a minimum gradient

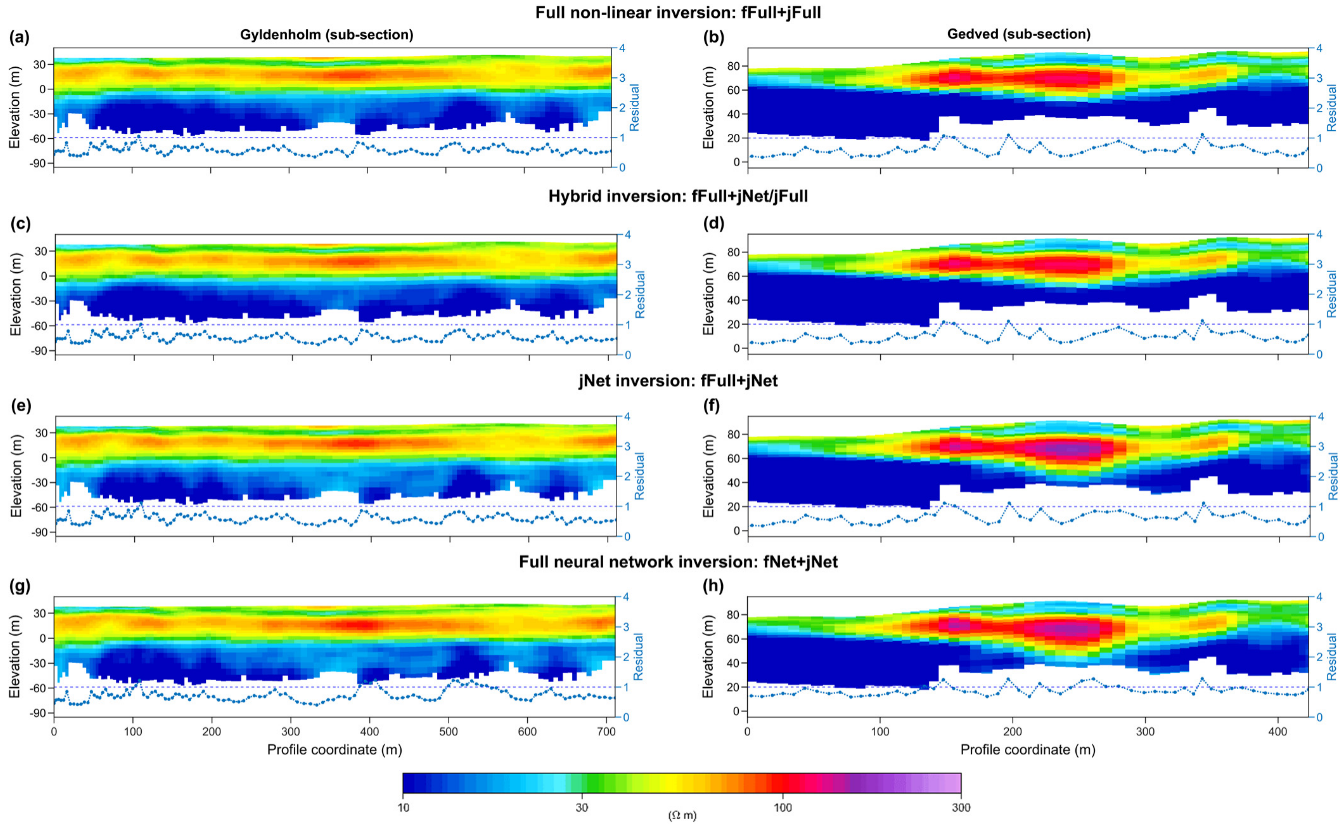



Fig. 6. Inversion results (masked below the depth of investigation) and the corresponding data residual (blue line and right axis). (a) and (b) Inversion by the formulation of the Jacobian by full nonlinear forward modeling (full nonlinear inversion). (c) and (d) Inversion by the Jacobian calculated by neural network for most iterations and finalized by full nonlinear derivatives (hybrid inversion). (e) and (f) Inversion where the Jacobian is computed by neural network in every iteration until convergence (jNet inversion). (g) and (h) Inversion when Jacobian is computed by neural network in every iteration and the forward response is computed from a neural network (full neural network inversions).

support (MGS) regularization scheme [28] (also called a sharp regularization). Fig. 7 shows an equivalent set of inversions as in Fig. 6, which shows that full nonlinear inversion, hybrid inversion, and the jNet inversion find the boundaries of various geological units effectively and result in similar data residual. However, the full neural network inversion results in higher deviation, which is a similar trend observed for smooth inversion as in Fig. 6(g) and (h). This suggests that the proposed schemes can be readily employed in practice.

## IV. Discussion

A visual analysis of the presented inversion results shows that the hybrid inversion framework is the recommended approach for future inversion schemes where high precision is required. The jNet inversion also captures all the features of the underlying geological structure and results in minor deviations in comparison to the conventional inversion scheme. Therefore, it can be used for applications that have some amount of tolerance in modeling accuracy. The full neural network inversion (fNet-jNet) results in near-identical results for medium-to-high conductive areas but has higher deviation in low conductive regions. However, it is significantly faster and provides a reasonable indication about the overall subsurface geology. Therefore, it can be used for preliminary inversion to estimate the potential target areas in real time. It may be effective for applications where relatively higher inaccuracies can be tolerated, as the overall structures and approximate resistivity values are still captured quite well. The resistivity models obtained from the full neural network inversion can also be used as starting models for the hybrid inversion framework.

Given that certain applications have larger tolerances for modeling inaccuracy and that there can be varying degrees of tolerance throughout different phases of interpretation [3], the presented inversion schemes give the flexibility in tuning an adaptable balance between computational time and inversion accuracy when modeling tTEM datasets.

It is important to note that we predict the double-sided, i.e., the symmetric derivatives instead of the typically used single-sided derivatives, since there is no additional overhead for the neural network. However, the speedup factor for jNet is given against the full nonlinear inversion when single-sided derivatives are used. If jNet is compared head-to-head with inversions based on double-sided derivatives, the speedup factor is doubled. It should also be noted that the computational time is calculated on a single processing core, and the utilization of GPUs for neural networks would drastically reduce the inversion computation times. In addition, the optimization of the system response convolution would further expedite the inversion process significantly.

The errors introduced in inversion by jNet are compensated by the hybrid framework where the derivatives from jNet are switched to the full nonlinear solution in the final iterations. For the case study presented in Table I, it takes only one

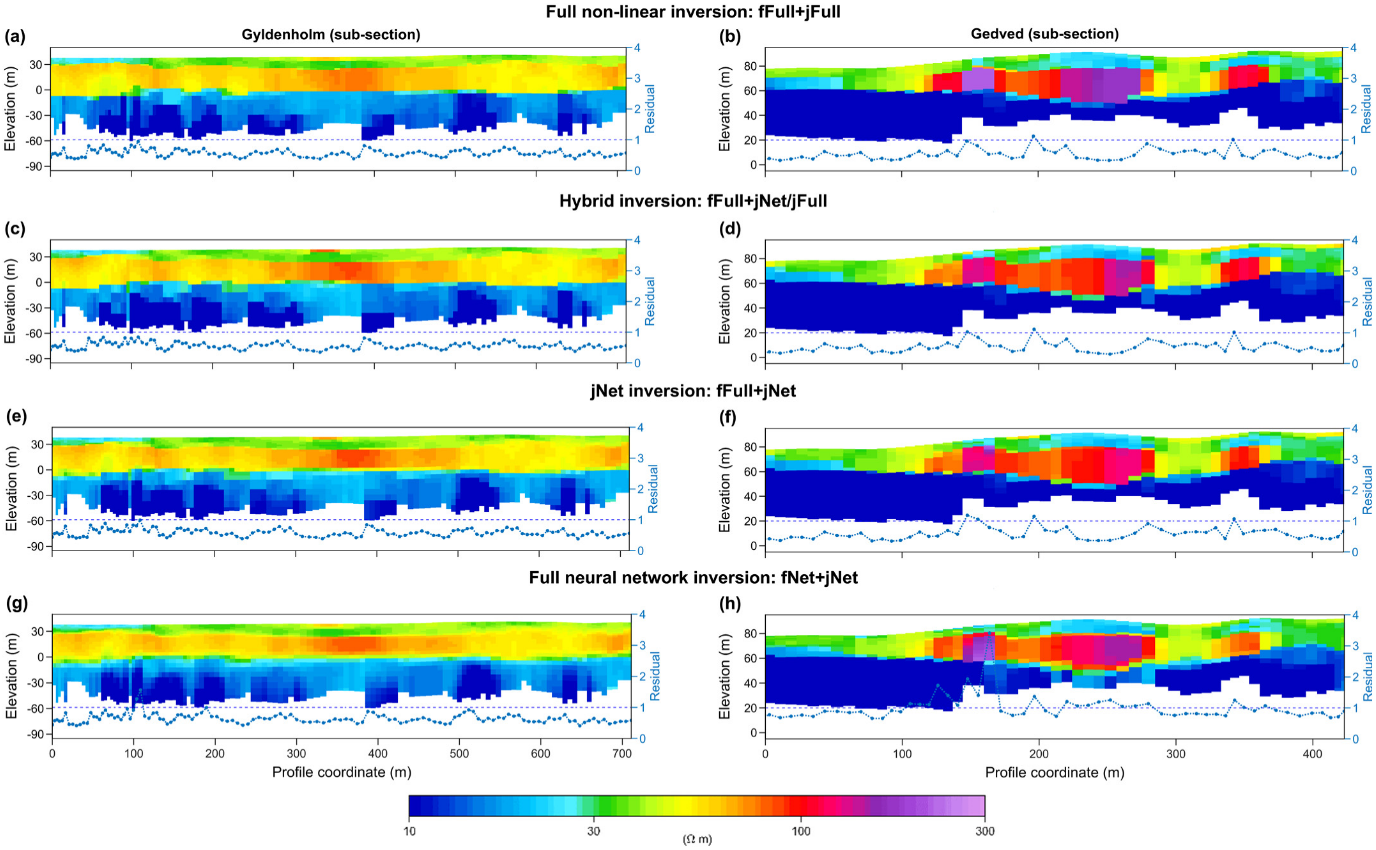



Fig. 7. Inversion results (masked below the depth of investigation) and the corresponding data residual (blue line and right axis) using the sharp inversion setup. (a) and (b) Full nonlinear inversion. (c) and (d) Hybrid inversion. (e) and (f) jNet inversion. (g) and (h) Full neural network inversions.

additional iteration. However, this may not hold true for every tTEM dataset as the inversion process is entirely model-dependent. However, in our experience, the hybrid framework generally requires 1–2 additional iterations on average to converge.

We have preferred to use a deep fully connected network due to its robustness, easy transferability, and effectiveness in approximating any arbitrary function to good accuracy [29]. However, it might be possible to further improve the accuracy of fNet and jNet by including sharp boundary models in the training model space and by making use of, e.g., state-of-the-art convolutional neural networks for refined inversion results, which is to be considered in future work.

Although it takes ~5 days for the preparation of the dataset and training the network, it has to be done only once for a particular TEM system, i.e., the tTEM system in our case. Once this procedure is completed, the network can be used indefinitely for faster inversions of any 2-D or 3-D volume tTEM datasets with satisfactory performance accuracy.

One of the limitations of the proposed study is its restriction to 1-D models. The extension to 3-D is possible but is not a trivial task and may be considered in future work. In addition, the field datasets used in this study are obtained from a ground-based towed TEM system. For the applicability of the proposed scheme for airborne systems, the network has to be trained taking flight height into consideration.

## V. Conclusion

We have presented a use case of neural networks within the inversion framework for a ground-based TEM system. The proposed framework provides a customized balance between inversion accuracy and computational time when modeling tTEM datasets. Neural network-based partial derivatives open up for faster inversions with little to no loss in inversion precision. A significant speedup factor of over 70 is realized in the calculation of the Jacobian matrix when the derivatives from jNet are used. We conclude that our hybrid inversion framework provides an efficient way of speeding up the inversion process. Inversion results show that the proposed scheme can safely be used since it gives near-identical results with the full nonlinear inversion for the tTEM system. The proposed methodology can also be extended for other ground-based and airborne systems.

## Acknowledgment

The tTEM data from Søften was collected during the project TopSoil, an Interreg project supported by the North Sea Programme of the European Regional Development Fund of the European Union, while the tTEM data from Gedved was collected during the project TREX from the Grønt Udviklings- og Demonstrations Program (GUDP) funded project with grant no. 34009-18-1453.

## REFERENCES


[1] E. Auken, T. Boesen, and A. V. Christiansen, "A review of airborne electromagnetic methods with focus on geotechnical and hydrological applications from 2007 to 2017," *Adv. Geophys.*, vol. 58, pp. 47–93, 2017.

[2] E. Auken and A. V. Christiansen, "Layered and laterally constrained 2D inversion of resistivity data," *Geophysics*, vol. 69, no. 3, pp. 752–761, May 2004.

[3] A. V. Christiansen *et al.*, "An efficient hybrid scheme for fast and accurate inversion of airborne transient electromagnetic data," *Explor. Geophys.*, vol. 47, no. 4, pp. 323–330, 2016.

[4] V. Das *et al.*, "Effect of rock physics modeling in impedance inversion from seismic data using convolutional neural network," in *Proc. 13th SEGJ Int. Symp.*, Tokyo, Japan, 2019, pp. 522–525.

[5] F. B. Jiang, Q. W. Dai, and L. Dong, "Nonlinear inversion of electrical resistivity imaging using pruning Bayesian neural networks," *Appl. Geophys.*, vol. 13, no. 2, pp. 267–278, 2016.

[6] R. Li *et al.*, "A fast approximation for 1-D inversion of transient electromagnetic data by using a back propagation neural network and improved particle swarm optimization," *Nonlinear Processes Geophys.*, vol. 26, no. 4, pp. 445–456, 2019.

[7] V. Puzyrev, S. Meka, and A. Swidinsky, "Deep convolutional neural networks for 1D inversion of electromagnetic data," in *Proc. 81st EAGE Conf. Exhib.*, 2019, pp. 1–5.

[8] D. Moghadas, "One-dimensional deep learning inversion of electromagnetic induction data using convolutional neural network," *Geophys. J. Int.*, vol. 222, no. 1, pp. 247–259, 2020.

[9] D. Conway, B. Alexander, M. King, G. Heinson, and Y. Kee, "Inverting magnetotelluric responses in a three-dimensional Earth using fast forward approximations based on artificial neural networks," *Comput. Geosci.*, vol. 127, pp. 44–52, Jun. 2019.

[10] P. Bai *et al.*, "(Quasi-) real-time inversion of airborne time-domain electromagnetic data via artificial neural network," *Remote Sens.*, vol. 12, no. 20, p. 3440, 2020.

[11] R. Guo, M. Li, G. Fang, F. Yang, S. Xu, and A. Abubakar, "Application of supervised descent method to transient electromagnetic data inversion," *Geophysics*, vol. 84, no. 4, pp. E225–E237, Jul. 2019.

[12] M. R. Asif *et al.*, "Effect of data pre-processing on the performance of neural networks for 1-D transient electromagnetic forward modeling," *IEEE Access*, vol. 9, pp. 34635–34646, 2021.

[13] T. S. Bording *et al.*, "Machine learning based fast forward modelling of ground-based time-domain electromagnetic data," *J. Appl. Geophys.*, vol. 187, Apr. 2021, Art. no. 104290.

[14] E. Auken, N. Foged, K. R. Andersen, N. S. Nyboe, and A. V. Christiansen, "On-time modelling using system response convolution for improved shallow resolution of the subsurface in airborne TEM," *Explor. Geophys.*, vol. 51, no. 1, pp. 4–13, Jan. 2020.

[15] C. Kirkegaard and E. Auken, "A parallel, scalable and memory efficient inversion code for very large-scale airborne electromagnetics surveys," *Geophys. Prospecting*, vol. 63, no. 2, pp. 495–507, 2015.

[16] E. Auken *et al.*, "tTEM—A towed transient electromagnetic system for detailed 3D imaging of the top 70 m of the subsurface," *Geophysics*, vol. 84, no. 1, pp. E13–E22, 2018.

[17] Y.-Y. Chen *et al.*, "Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes," *Sensors*, vol. 19, no. 9, p. 2047, 2019.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Nature*, vol. 323, pp. 533–536, 1986.

[19] S.-C. Wang, "Artificial neural network," in *Interdisciplinary Computing in Java Programming*. New York, NY, USA: Springer, 2003, pp. 81–100.

[20] A. V. Joshi, "Perceptron and neural networks," in *Machine Learning and Artificial Intelligence*. Cham, Switzerland: Springer, 2020, pp. 43–51.

[21] B. C. Csáji, "Approximation with artificial neural networks," M.S. thesis, Dept. Sci., Eötvös Loránd Univ., Budapest, Hungary, 2001.

[22] Z. Zhang, "Artificial neural network," in *Multivariate Time Series Analysis in Climate and Environmental Research*. Cham, Switzerland: Springer, 2018, pp. 1–35.

[23] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 1990, pp. 21–26.

[24] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, no. 4, pp. 525–533, 1993.

[25] J. Zheng, Q. Ma, and W. Zhou, "Performance comparison of full-batch BP and mini-batch BP algorithm on spark framework," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2016, pp. 1–5.

[26] E. Auken *et al.*, "An overview of a highly versatile forward and stable inverse algorithm for airborne, ground-based and borehole electromagnetic and electric data," *Explor. Geophys.*, vol. 46, no. 3, pp. 223–235, Sep. 2015.

[27] A. Vest Christiansen and E. Auken, "A global measure for depth of investigation," *Geophysics*, vol. 77, no. 4, pp. WB171–WB177, Jul. 2012.

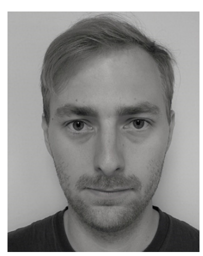
[28] G. Vignoli, G. Fiandaca, A. V. Christiansen, C. Kirkegaard, and E. Auken, "Sharp spatially constrained inversion with applications to transient electromagnetic data," *Geophys. Prospecting*, vol. 63, no. 1, pp. 243–255, Jan. 2015.

[29] J. Heaton, *Introduction to Neural Networks With Java*. St. Louis, MO, USA: Heaton Research, 2008.



**Muhammad Rizwan Asif** received the B.S. degree in electrical (telecommunication) engineering from COMSATS University Islamabad (CUI), Islamabad, Pakistan, in 2013, and the M.S. and Ph.D. degrees in information and communication engineering from Xi'an Jiaotong University, Xi'an, China, in 2016 and 2019, respectively.

Since 2019, he has been holding a post-doctoral position at the Department of Engineering, Aarhus University, Aarhus, Denmark. He is working in close collaboration with the Hydrogeophysics Group (HGG), Department of Geoscience, Aarhus University, to support research projects employing electrical and electromagnetic methods. His research interest includes machine learning applications for geophysical data. He is also interested in image processing and computer vision for intelligent automated systems.


**Thue S. Bording** received the B.Sc. degree in geology from Aarhus University, Aarhus, Denmark, in 2010, and the Ph.D. degree in geophysics from Aarhus University in 2016 with a focus on geothermal modeling and measurement of rock thermal properties using Monte Carlo methods.

Since 2016, he has been holding a post-doctoral position at the Hydrogeophysics Group (HGG), Department of Geoscience, Aarhus University. His research revolves around cross-borehole electrical tomography, induced polarization, and transient electromagnetics.



**Pradip K. Maurya** received the Ph.D. degree from Aarhus University, Aarhus, Denmark, in 2017, with a focus on the direct current resistivity and induced polarization (DCIP) method.

He is holding a post-doctoral position at the Hydro Geophysics Group, Aarhus University, Aarhus, Denmark. Apart from the DCIP method, his research also focuses in the field of airborne and ground-based electromagnetic methods for near-surface applications. He has a vast experience in data processing and inversion of electrical and electromagnetic data. He also has a strong interest in the development of geophysical instruments and is involved in the development of tTEM instrument.

**Bo Zhang** received the Ph.D. degree from Jilin University, Changchun, China, in 2017.

He then held a post-doctoral position for two years at Aarhus University, Aarhus, Denmark. He is an Associate Professor with the College of Geo-Exploration Sciences and Technology, Jilin University, where he is mainly engaged in the time-domain electromagnetic (EM) modeling and inversion.

**Gianluca Fiandaca** received the Ph.D. degree in applied geophysics from the University of Palermo, Palermo, Italy, in 2007.

He worked at the Hydrogeophysics Group, Aarhus University, Aarhus, Denmark, for ten years. He is an Associate Professor of applied geophysics at the University of Milan, Milan, Italy. His main research interests are data acquisition, processing, and inversion in electric and electromagnetic methods, with emphasis on induced polarization.

**Denys J. Grombacher** received the B.Sc. degree in astrophysics from the University of Alberta, Edmonton, AB, Canada, in 2010, and the Ph.D. degree in geophysics from Stanford University, Stanford, CA, USA, in 2015.

He is interested in electromagnetic and nuclear magnetic resonance technologies for groundwater explorations.

**Anders V. Christiansen** received the Ph.D. degree from Aarhus University, Aarhus, Denmark, in 2003, with a focus on airborne transient electromagnetic (TEM) methods and inversion methodologies.

He is an Associate Professor with the Hydrogeophysics Group, Aarhus University. In the later years, he has focused on the integration of various data types (e.g., geophysics with borehole information and hydrological observations) with the purpose of making objective and transparent processes that generate maps and other outputs that are in better agreement with the actual needs of the end users. Overall, his research focus still remains within these subjects.

**Esben Auken** received the M.Sc. and Ph.D. degrees in geophysics from Aarhus University, Aarhus, Denmark, in 1991 and 1996, respectively.

He was a Professor of geophysics with the Department of Geoscience, Aarhus University. During his affiliation with Aarhus University, he led a big research team [Hydrogeophysics Group (HGG)] consisted of 25–30 scientists all working with electrical and electromagnetic methods for high-resolution mapping of the shallow subsurface with a focus on surface and groundwater processes. He is the Vice Director of the Geological Survey of Denmark and Greenland (GEUS), Copenhagen, Denmark. GEUS carries out activities to explore and protect geological resources in Denmark and Greenland. He is also a Co-Founder of SkyTEM Surveys Aps, Denmark, Aarhus Geo-Software, Denmark, and Aarhus Geo-Instruments, Denmark, all spin-out companies from the research group. The focus areas are surface nuclear magnetic resonance, ground-based and airborne transient electromagnetic (TEM), and surface and cross-borehole direct current and induced polarization. His research interests include mapping, compilation, and storage of data, research, monitoring and consultancy within water, energy, minerals, and climate and environment.

**Jakob Juul Larsen** (Senior Member, IEEE) was born in Denmark, in 1973. He received the M.Sc. and Ph.D. degrees in experimental physics from Aarhus University, Aarhus, Denmark, in 1998 and 2000, respectively.

From 2000 to 2013, he held research and teaching positions in private and public companies. Since 2013, he has been with the Department of Engineering, Aarhus University, where he holds a position as an Associate Professor. He works with geophysical instrumentation, signal processing, and machine learning. His research interests are the domains of magnetics, transient electromagnetics, direct current-induced polarization, and surface nuclear magnetic resonance.