A Nonlinear Regression Application via Machine Learning Techniques for Geomagnetic Data Reconstruction Processing

Huan Liu, *Member, IEEE*, Zheng Liu, *Senior Member, IEEE*, Shuo Liu, *Student Member, IEEE*, Yihao Liu, Junchi Bin, Fang Shi, and Haobin Dong

***Abstract*— The integrity of geomagnetic data is a critical factor in understanding the evolutionary process of Earth's magnetic field, as it provides useful information for near-surface exploration, unexploded explosive ordnance detection, and so on. Aimed to reconstruct undersampled geomagnetic data, this paper presents a geomagnetic data reconstruction approach based on machine learning techniques. The traditional linear interpolation approaches are prone to time inefficiency and high labor cost, while the proposed approach has a significant improvement. In this paper, three classic machine learning models, support vector machine, random forests, and gradient boosting were built. Besides, a deep learning algorithm, recurrent neural network, was explored to further improve the training performance. The proposed learning models were used to specify a continuous regression hyperplane from a training data. The specified regression hyperplane is a mapping of the relation between the mock-up missing data and the surrounding intact data. Afterward, the trained models, essentially the hyperplanes, were used to reconstruct the missing geomagnetic traces for validation, and they can be used for reconstructing further collected new field data. Finally, numerical experiments were derived. The results showed that the performance of our methods was more competitive in comparison with the traditional linear method, as the reconstruction accuracy was increased by approximately 10%∼20%.**

***Index Terms*— Deep neural network, geomagnetic, machine learning, modeling, reconstruction.**

Manuscript received July 11, 2017; revised September 12, 2017, November 23, 2017, and April 3, 2018; accepted May 22, 2018. Date of publication July 31, 2018; date of current version December 24, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 41474158 and Grant 41504137, in part by the National Key Scientific Instrument and Equipment Development Project of China under Grant 2014YQ100817, in part by the Foundation of Science and Technology on Near-Surface Detection Laboratory under Grant TCGZ2015A008, and in part by the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan). *(Corresponding author: Haobin Dong.)*

H. Liu is with the School of Automation, Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan 430074, China, also with the Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China, and also with the School of Engineering, The University of British Columbia Okanagan Campus, Kelowna, BC V1V 1V7, Canada (e-mail: huan.liu@cug.edu.cn).

Z. Liu, S. Liu, Y. Liu, J. Bin, and F. Shi are with the School of Engineering, Faculty of Applied Science, The University of British Columbia Okanagan Campus, Kelowna, BC V1V 1V7, Canada (e-mail: zheng.liu@ubc.ca).

H. Dong is with the School of Automation, China University of Geosciences, Wuhan 430074, China, and also with the Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China (e-mail: donghb@cug.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2852632

I. Introduction

CONTINUOUS observations of the geomagnetic field are carried out with the characteristic times spanning from seconds to decades at magnetic observations [1]. However, the integrity of geomagnetic data cannot be guaranteed consistently, especially when malfunctions of the observing system happen during the observations [2]. In this case, undersampled data or missing traces could compromise the interpretation accuracy of the geomagnetic data [3], which necessitate the research on the reconstruction of the geomagnetic data.

Up to now, several geomagnetic data reconstruction methods were developed. In terms of numerical simulations studies, [4] and [5] investigated how to predict the unknown geomagnetic field changes based on the data assimilation technique. The simulation results were encouraging, but they stayed at the stage of theoretical simulation that has not been adapted for a real application. The performance remains to be verified. Some other methods have been applied to practical scenarios. Several globe geomagnetic data models were proposed in [6]–[9], and the prediction of the series of models (e.g., CALS3K) can be adopted as a baseline for evaluation and observation of geomagnetic field changes [10]. There are some other well-accepted methods being reported, such as the spherical harmonic method [11] and multiple model fusion [12]. Nonetheless, getting good reconstruction results from these methods is dependent on certain assumptions. For example, the geomagnetic records should comprise a limited number of linear events, and the reconstructed data should be sparser than the observed data with missing traces, which are posing considerably restrictions for now.

Nowadays, because machine learning can automatically explore the hidden features or relationships within the data set, it has become increasingly used in many scientific fields, including nondestructive testing [13], objective detection [14], and classification and linear regression [15]. This is often an attractive alternative to reduce the manual work in varieties of fields. A vast array of studies has established the primary tools of machine learning, such as linear regression [16], decision trees [17], support vector machine (SVM) [18], artificial neural networks [19], and instance-based learning [20]. The primary models performed by machine learning include regression, classification, clustering, and so on. To sum up, due to its good

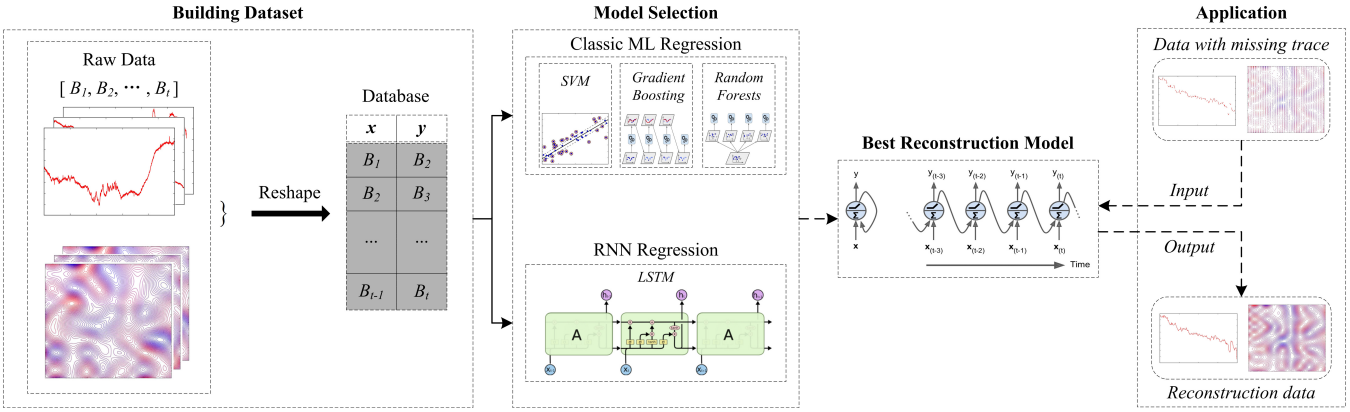


Fig. 1. Pipeline of the proposed geomagnetic data reconstruction framework, which includes four main modules, including data set building, classic machine learning regression, RNN regression, and the selected model for regression and reconstruction.

performance, machine learning spreads rapidly throughout multiple fields, suggesting that it is likely to lead the next wave of innovation in geophysics [21].

How can we process geomagnetic data using machine learning? Despite the above-mentioned applications, it is still to be investigated in the improvement of geomagnetic exploration. The related techniques have been preliminarily applied in the reservoir parameters characterization [22], [23]. Inspired by machine learning, we consider the geomagnetic data reconstruction or interpolation as a regression problem for continuous output. In other words, machine learning can be used to generate an approximating function (a continuous hyperplane) to derive the missing traces.

The proposed method mainly consists of three steps. The first step is the process of building training data set. The training data set does not contain missing traces, such that the model later trained by this data set is a mapping of the previous values, $x(t)$, to the following values, $x(t+1)$. The data set is randomly split into three parts: training, validation, and testing data. The second step is to train a regression model (classic machine learning and RNN), which fits a hyperplane from the training data. The third step is to reconstruct the missing geomagnetic traces of the mock-up defected data set. Afterward, the reconstructed data are validated and tested, and the best optimistic model is obtained according to the accuracy. The proposed machine learning approach only depends on the characteristics of the originally collected data, which can overcome some drawbacks, e.g., the previously mentioned restriction of linear events, the sparsity of reconstruction data, and so on. In consequence, it not only breaks out the previous limitations but also shows stronger adaptability in different kinds of the data set. Furthermore, it can help researchers to obtain the complete information of Earth's magnetic field for near-surface exploration and detection for magnetic materials.

II. Theory

As a recently vastly developed technology, machine learning methods have not been used to reconstruct the undersampled geomagnetic data. A framework, geomagnetic data reconstructor (GDR) based on machine learning, is proposed, while an overview of the algorithms for training and inference in the GDR framework is presented in Algorithms 1 and 2. As shown in Fig. 1, the framework is composed of four modules: 1) database building module (refer to Fig. 5), which aims to transform 2-D or 3-D data to a 1-D time-series spaced data; 2) classic machine learning regression module, which is used for extracting pattern representations from the time-series data and establishing deterministic models (refer to Algorithm 1); 3) recurrent neural network (RNN) regression module, which manipulates on time-series data, and it is utilized for generating candidate memory cells to build a feedforward neural network (refer to Algorithm 2); and 4) one of the models is selected based on customized requirement, which is accurate in most instances.

From the perspective of statistics, the reconstruction of geomagnetic data can be modeled as a regression problem. The logic of regression can be explained in the following descriptions. Suppose that the training data set with n data pairs as $\{(x_i, y_i), i = 1, 2, \ldots, n\}$, where x_i is the feature vector. The feature vector is an array of data that contains the values of the points that surrounding a target sample point. On the other hand, the value of the target sample point is to be calculated from the feature vector during the testing or reconstructing step. And y_i (the true value of the sample point) is the corresponding label of x_i, and it is used to compare with the calculated target sample point value, so an accuracy is derived. One should note that y_i is a continuous value. Solving a regression problem requires the construction of an approximate function $f(\boldsymbol{x})$ mapping from $\boldsymbol{x}$ to y ($y \approx f(\boldsymbol{x})$). It is expected that when an independent variable vector x_i is known, the dependent variable y_i can be predicted. In the procedure of training a model, the features with missing geomagnetic data traces are considered as the vector x_i, which is to be fed into a model. And the missing data are predicted to be $\hat{y}_i$, which is compared with the completed magnetic field strength as the ground truth y_i.

A. Classic Machine Learning Methods

In this reconstruction case, three classic machine learning models were built for this regression problem, i.e., *SVM*, *gradient boosting*, and *random forests*. The brief introduction of each model is described as follows.

1) Support Vector Machine: SVM was originally invented as a classification system and operates by maximizing the margin of decision boundary, but in terms of its regression algorithm, it also contains all the main features that characterize the maximum margin algorithm: a nonlinear function is learned by linear learning machine mapping into high-dimensional kernel-induced feature space [24]. The capacity of the system is controlled by the parameters that do not depend on the dimensionality of feature space. In a regression model of SVM, first, the input variables $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$ are mapped into an n-dimensional space using a fixed mapping method [25]. Afterward, a linear regression model is constructed in this n-dimensional space [18]. Hence, the linear regression model can be described using mathematical notation as

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \hat{y}_i g_i(\boldsymbol{x}) + b \tag{1}$$

where $g_i(\boldsymbol{x})$ stands for a set of nonlinear transformation, $\hat{y}_i$ stands for the predicted value, and b denotes the predicted "bias" value. SVM regression uses ε-insensitive loss as loss function for training. Further information can be found in [24].

2) Gradient Boosting: Gradient boosting can produce an ensemble regression model in the form of several weak models [26]. It not only optimizes the differential of loss functions, but also sets up additional models in a forward-looking way. It uses a training sample $\{(x_i, y_i)\}_1^n$ of known $(\boldsymbol{x}, y)$ values to obtain an approximation function $f(\boldsymbol{x})$ mapping $\boldsymbol{x}$ to y. In addition, it minimizes the value of some specified loss function $L(y, f(\boldsymbol{x}))$ over the joint distribution of all $(\boldsymbol{x}, y)$ values. The frequently employed loss function $L(y, f(\boldsymbol{x}))$ includes squared-error $(y - f(\boldsymbol{x}))^2$ and absolute error $|y - f(\boldsymbol{x})|$, depending on the type of problem being solved.

In this paper, the squared-error function is applied as the loss function. For a given loss function, a regression tree is fit on the negative gradient in each of the training stages. Thus, the approximation value can be predicted using the vector of $\boldsymbol{x}$ with known values and corresponding values of y. In the regression model of gradient boosting, it assumes that there exists an actual value, and a weighted sum of function $h_i(\boldsymbol{x})$ can be explored approximately [26], which is called base learners as

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \gamma_i h_i(\boldsymbol{x}) + b \tag{2}$$

where $\gamma_i = \arg\min_\gamma L(y_i, f(x_i) + \gamma)$ is a separate optimal multiplier, which can be selected using line search [27].

3) Random Forests: Random forest is a famous machine learning algorithm, which is constructed by numerous decision trees. When a regression model is trained, the output predicted value depends on the mean prediction of all the trees, rather than each individual tree. Using a multitude of decision trees can eliminate overfitting. Furthermore, the precision of the predicted value can be improved.

In the procedure of training a random forests model, the general bagging techniques are always applied to tree learners [28]. To be specific, given a training variable $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$ with ground truth $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$, bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples, for $b = 1, 2, \ldots, B$, as follows.

1) Replace n training variables from input data $\boldsymbol{x}$ and output ground truth $\boldsymbol{y}$ to B samples, which called $\boldsymbol{x}_b$ and $\boldsymbol{y}_b$, respectively.
2) Train a regression tree f_b on $\boldsymbol{x}_b$ and $\boldsymbol{y}_b$.

In this procedure, the model should be regarded as an ensemble of B trees $\{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_B(\boldsymbol{x})\}$, where $\boldsymbol{x}$ is an n-dimensional vector which is defined to be numerical properties and can be calculated from each tree. The ensemble produces B outputs $\{\hat{f}_1(\boldsymbol{x}), \hat{f}_2(\boldsymbol{x}), \ldots, \hat{f}_B(\boldsymbol{x})\}$, where $\hat{f}_b(\boldsymbol{x})$ is the prediction by Bth tree, which implies that the outputs of all trees are aggregated to produce one final prediction $\hat{f}$.

Consequently, the predictions from samples $\boldsymbol{x}$ that should not be seen can be obtained by calculating the mean of the predictions from all the individual decision trees on $\boldsymbol{x}$ as [29]

$$\hat{f} = 1/B \sum_{b=1}^{B} \hat{f}_b(\boldsymbol{x}). \tag{3}$$

In addition, taking the majority vote of all the decision trees can also get the predicted values.

The main steps in our reconstruction method using classic machine learning techniques are given in Algorithm 1. Once all the following steps have been successfully completed, the output associative function $f(x)$ of each model will be obtained. In practical application, the prediction procedures are as follows.

1) Input a new geomagnetic data vector x^* with missing traces and transpose it into a 1-D vector. It contains the magnetic field strength values varying with respect to time. The strength of magnetic field from time (t) to $(t+j)$ is unknown, $j \geq 1$, while that from time $(t-i)$ to

Algorithm 1 Reconstruction Based on Classic Machine Learning

Result: Geomagnetic data reconstruction.
Input : Experimental geomagnetic data x, missing geomagnetic data y.
Output: Predicted geomagnetic data.

Training:
A data set is built where x stand for the strength of magnetic field at time (t) and y stands for the strength of magnetic field at time $(t + 1)$;
Building different independent regression models:
    $clf1 = svr()$
    $clf2 = RandomForestRegression()$
    $clf3 = GradientBoostingRegression()$
Input all the pairs of (x_t, x_{t+1}) to each of the models to train.

Testing:
The trained models are applied to the geomagnetic data x with missing traces and obtain the predicted missing data y';
Evaluating the results to obtain the most optimist model.

$(t-1)$ is available, $i > 1$. Ideally, i is large enough, but in bad conditions, it would be relatively smaller compared with the total number of t.

2) Input x^*_{t-i} to the model function, and there is $x^*_{t-i+1} = f(x^*_{t-i})$.
3) Input x^*_{t-i+1} to obtain $x^*_{t-i+2} = f(x^*_{t-i+1})$.
4) Repeat steps 2) and 3) until x^*_t is predicted as $x^*_t = f(x^*_{t-1})$.
5) Input the predicted value x^*_t to obtain $x^*_{t+1} = f(x^*_t)$.
6) Repeat step 5) until x^*_{t+j} is predicted as $x^*_{t+j} = f(x^*_{t+j-1})$.

Generally, x^*_t could be directly obtained based on $x^*_t = f(x^*_{t-1})$ if x^*_{t-1} is known when using the classic machine learning methods. In this case, we assume that all inputs and outputs are independent of each other. However, in the reconstruction task, if we want to predict the next value, it would be better known which values came before it. RNNs (refer to Section II-B) are recurrent because they perform the same task for every element of a time-series vector, with the output being statistically dependent on the previous computations. Thus, $x^*_{t-i}, x^*_{t-i+1}, \ldots, x^*_{t-1}$ are known, and we should input them to the model function sequentially, to generate a long short-time memory (LSTM) when using RNNs. In this case, x^*_t not only depends on x^*_{t-1} but also depends on the previous inputs. Likewise, input the predicted value x^*_t into the network, and x^*_{t+1} can be predicted based on the LSTM formed by $x^*_{t-i}, x^*_{t-i+1}, \ldots, x^*_{t-1}$ and x^*_t. The rest can also be done iteratively in the same manner. The more the number of known values, the higher the accuracy of predicted values.

B. Deep Neural Network Method

To be specific, the reconstruction of the geomagnetic data is a time- and spatial-series prediction problem, which is also regarded as a type of regressive modeling problem, but it adds a degree of complexity of time. The sequence dependence from the input variables are considered in series in terms of both space and time, which is unlike that of a simple regression predictive model. Since the classic machine learning models have drawbacks in reflecting temporal dependence, a kind of deep neural network named RNN was adopted to make the reconstruction results more efficiently.

1) Recurrent Neural Network: RNN is a powerful neural network structure that is designed to handle sequence dependence [30]. It is also the extension of feedforward neural networks but adds a feedback connection [31]. The feedback connection means that the outputs of the model can be fed back into itself. To some extent, this kind of neural network has the ability to memorize.

As in Fig. 2, this is the general architecture of RNN. Fig. 2 (left) shows the original format of RNN. There is a loop around the hidden layer, which means that the output parameters of the hidden layer can be fed into not only output layer but also itself. This kind of loop structure allows the neural network to pass the information from the last step to current step. The loop format of the computational graph is tricky to understand, but this loop graph can be unfolded to a chaining of neurons,

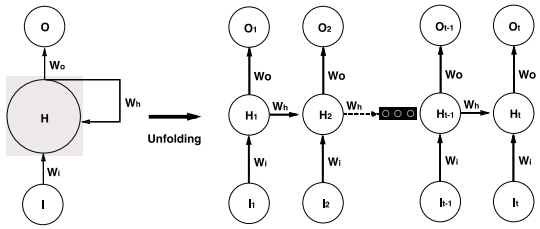


Fig. 2. Flow structure of RNN.

which has a repetitive structure as shown in Fig. 2 (right). In the unfolded graph, each repetitive unit represents a time step, and the length of total time steps is T. One should note that there is an output node at each time step, but the node is not necessary for some tasks. In our experiments, the final one output is our only concern, because the network should return a single regression value. Thus, there is only one output node in the last time step in this case. Each node in the middle denotes the status (a set of parameters) of the hidden layer at time t. The status of the hidden layer will be updated across all time steps, and the update function can be abstracted by the following equation [32]:

$$h(t) = f(W \cdot [h_{t-1}, x_t]) \tag{4}$$

the inputs of this update function are input data x_t and the state of last time step h_{t-1}, and it will output the state of the hidden layer of the current step. W stands for the set of weight parameters in different time steps of RNN.

There are several types of RNN. The main difference among them is the update function. In this paper, we adopted the most typical variants called the LSTM network.

2) Long Short-Term Memory: LSTM is a typical kind of RNN [33], which has the capacity of learning dependences within long term. The main reason for LSTM owing the ability to remember long-term information is that there is an extra state conveyor in addition to the memory block across the entire time series [34]. Therefore, it has the ability to transfer information from past time step to future time step. The update function of LSTM is obviously more complicated than that in basic RNN, as shown in Fig. 3.

The upstate function is composed of three functions, including input gate $\boldsymbol{f}_i$, forget gate $\boldsymbol{f}_f$, and output gate $\boldsymbol{f}_o$. In $\boldsymbol{f}_f$, first, it concatenates the current input vector $\boldsymbol{x} = (x_1; x_2; \ldots; x_t)$ and the output vector of hidden layer $\boldsymbol{h} = (h_1; h_2; \ldots; h_t)$, which is equal to the cell's output for the last time step, i.e., $h_t = x_{t-1}$. After this, each entry will go through a sigmoid ($\boldsymbol{\sigma}$) function [35]. The sigmoid function outputs numbers between zero and one, describing how much of each component should be let through. Therefore, if a pointwise multiplication operator (refer to $\otimes$) is done to them with the global state (the top horizontal conveyor from H_{t-1} to H_{t+1}, that is labeled as s_{t-1} and s_{t+1}) from the last time step, it will be a forgetlike gate which can decide what information in the global state should be forgot. Here is the formula of the forget gate [36]

$$\boldsymbol{f}_f = \boldsymbol{s}_{t-1} \otimes \boldsymbol{\sigma}(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

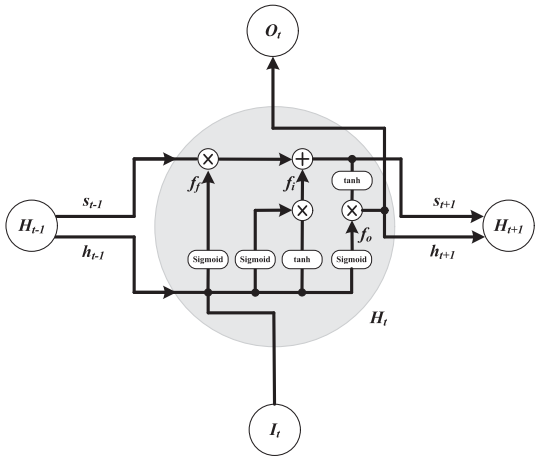


Fig. 3. Update function cell of LSTM.

where W_f are the weight matrices of $\boldsymbol{f}_f$ for its connection to $\boldsymbol{x}$ and $\boldsymbol{h}$. s_{t-1} stands for the cell state and b_f stands for the bias term. In $\boldsymbol{f}_i$, it is the same as $\boldsymbol{f}_f$, which also need to combine the input vector and the output vector of last time step, applying $\boldsymbol{\sigma}$ function to decide which value will be updated. The only difference from $\boldsymbol{f}_f$ is that it has another branch which applies ***tanh*** activation function to create a vector of new candidate values [37]. Then, a pointwise multiplication operator will be used to combine the two branches, and the result will be a vector of update information as

$$i_s = \boldsymbol{\sigma}(W_{\text{is}} \cdot [h_{t-1}, x_t] + b_{\text{is}}) \tag{6}$$

$$i_t = \boldsymbol{tanh}(W_{\text{it}} \cdot [h_{t-1}, x_t] + b_{\text{it}}) \tag{7}$$

$$\boldsymbol{f}_i = (i_s \otimes i_t) \oplus \boldsymbol{f}_f. \tag{8}$$

The update information will be added to the global state conveyor by pointwise addition operator, which at this point of time, is the result of $\boldsymbol{f}_f$. The updated global state will be transferred to $\boldsymbol{f}_o$ and the next time step. Finally, in $\boldsymbol{f}_o$, it needs to produce the logic gates by $\boldsymbol{\sigma}$ function. Then, it will apply ***tanh*** function to create the candidate vector for outputting at one branch of the global state conveyor, which at the point of time, is the result of $\boldsymbol{f}_i$. At last, a pointwise multiplication operator will be used to decide what information needs to be output. The resulting vector of $\boldsymbol{f}_o$ can be fed not only to the output neural layer but also to the hidden layers at the next time step [38].

The main steps using the deep neural network method are given in Algorithm 2, where *look_back* denotes the number of previous time series, which is regarded as the input variables to predict the values of next time period. Fig. 4 shows the LSTM RNN architecture used in this application. The network is built by stacking multiple LSTM blocks, which can be considered as a feedforward neural network unrolled in time where each block shares the same update cell as shown in Fig. 3. The input to the network at a given time step goes through multiple LSTMs in addition to propagation through time and LSTMs. In addition, the prediction procedures using this algorithm are the same as Algorithm 1.

Algorithm 2 Reconstruction Based on RNN

Result: Geomagnetic data reconstruction.
Input : Experimental geomagnetic data x, missing geomagnetic data y.
Output: Predicted geomagnetic data.

Training:
A data set is built where x stand for the strength of magnetic field at time (t) and y stands for the strength of magnetic field at time $(t + 1)$;
Building a LSTM neural network:
$model = Sequential()$
$model.add\ (LSTM(hidden\ layer,\ input\_shape =$
$(visible\ layer, look\_back)))$
$model.fit(x, y, epochs = 100)$
The *input visible* layer is set as 1 while the *hidden layer* contains 10 LSTM blocks. The *output layer* makes value predictions based on the *look_back*, which it defaulted to 1; *epochs* stands for the number of overall times for the training vectors;
Fit all pairs of (x_t, x_{t+1}) to the network, which is trained for a *epochs* of 100.

Testing:
Input the missing geomagnetic data x to the trained LSTM network to obtain the predicted missing data y';

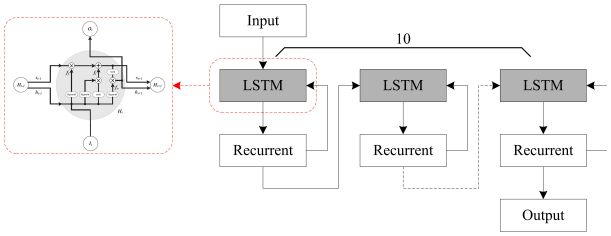


Fig. 4. LSTM RNN architecture.

To sum up, for the training geomagnetic data set in Algorithms 1 and 2, the point pairs (variable x and ground truth y) with completed data are known, and the GDR framework combined with our algorithms is used to create a best practice regression model $y = f(x)$ hidden in these training point pairs. Other missing data y^* in (x^*, y^*) can be predicted after feeding x^* into this trained model.

In the training stage, we selected several examples without missing traces, whose geomorphological structures are similar to those of the reconstructed geomagnetic data. Once the database is built, we feed all point pairs from the experimental training data into each of the regression models (SVM, gradient boosting, random forests, and LSTM) to train. Then, four continuous regression models can be generated, and they also could be saved for future use for reconstruction. In the prediction stage, the values of missing geomagnetic data are unknown, but they could be predicted using the values that are before or after them in a period of time. All variables x with missing traces are input consecutively to the regression models

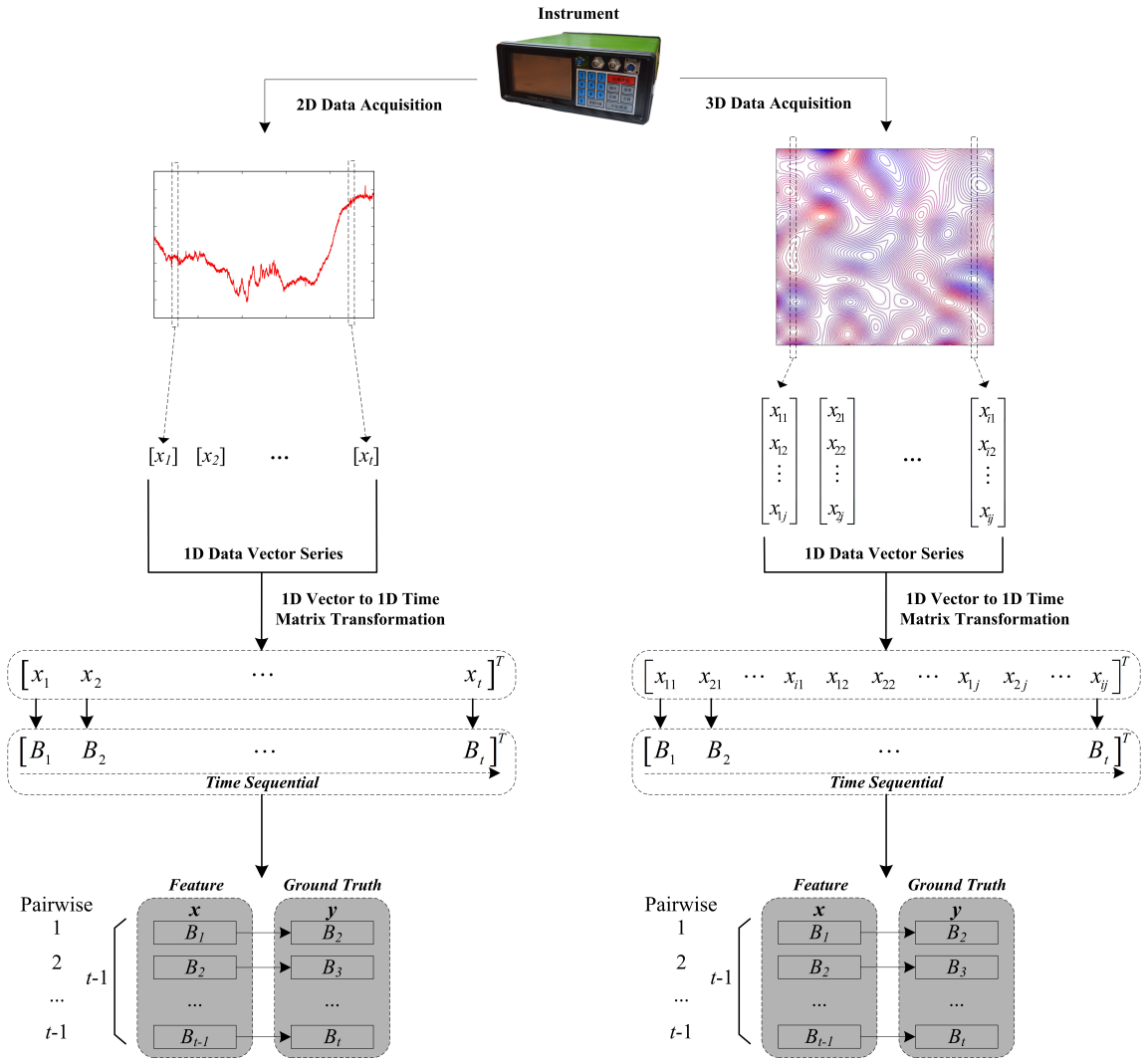


Fig. 5. Procedure of preprocess geomagnetic time-series data.

that have been trained in the previous stage, thus allowing us to obtain the missing geomagnetic data.

III. Experimental Results and Discussion

The proposed model framework for the reconstruction of geomagnetic data consists of four steps, including: 1) building data set; 2) training regression model; 3) reconstructing the missing geomagnetic traces; and 4) evaluating the results using new data sets. For all the experiments, the root mean square error (RMSE) and R-squared [39] were used to evaluate the prediction performance of different modeling methods. Their equations are as follows [13]:

$$\begin{cases} \text{RMSE} = \sqrt{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \\ R_{-\text{squared}} = 1 - \dfrac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y_i})^2} \end{cases} \tag{9}$$

where y_i denotes the practical value, $\hat{y}_i$ denotes the predicted value, and $\overline{y_i}$ denotes the mean value of all the predictions. R-squared is a relative measure of fitting, and RMSE is an absolute measure of fitting. The lower the RMSE, the better the model. For R-squared, it ranges from zero to one, and the higher value it is (close to one), the better the model. The experiments and the corresponding numerical analysis were implemented using Python scikit-learn and Keras with Windows 10, Intel Core i7, 3.2-GHz CPU, and 16-GB random access memory (RAM).

A. Geomagnetic Data Preprocessing

For using machine learning models in geomagnetic data reconstruction, it is necessary to include a rule to transform the geomagnetic data into the form of data pairs (feature, ground truth), which should be suitable for the input of each model. The overall data preprocessing flowchart of our proposed method is shown in Fig. 5. Typically, for a set of 2-D data, first, the input variables $\boldsymbol{x} = \{x_1, x_2, \ldots, x_t\}$ whose t subscript indicates the order of the data collected. The values in it are the magnetic field strength values varying with respect to time. Then, $\boldsymbol{x}$ can be transposed into a 1-D column vector $[x_1, x_2, \ldots, x_t]^T$ directly.

TABLE I
MACHINE LEARNING PARAMETERS USED FOR EACH OF THE DIFFERENT MODELS

Model	Main parameter definitions
SVM	cache_size=200, degree=3, gamma='auto', kernel='rbf', decision_function_shape='ovo'
Gradient Boosting	learning_rate=80, criterion='mse', n_estimators=100, max_depth=3, min_samples_leaf=10
Random Forests	random_state=80, criterion='entropy', n_estimators=100, max_depth=3, min_samples_leaf=20
LSTM	visible_layer=1, hidden_layer=10, look_back=1, epochs=100

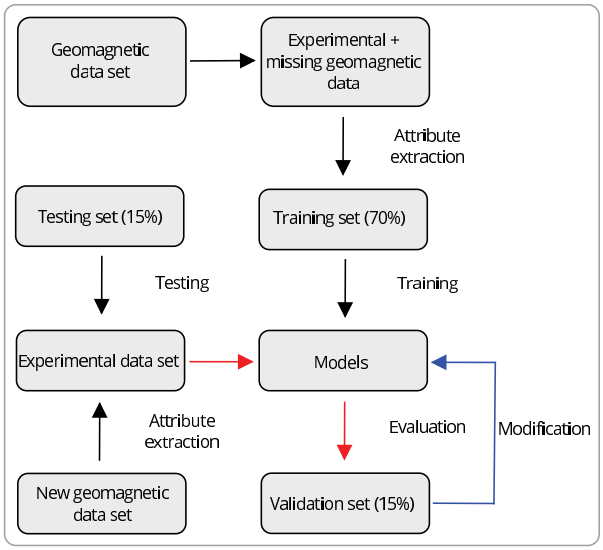


Fig. 6. Flowchart of training a regression model.

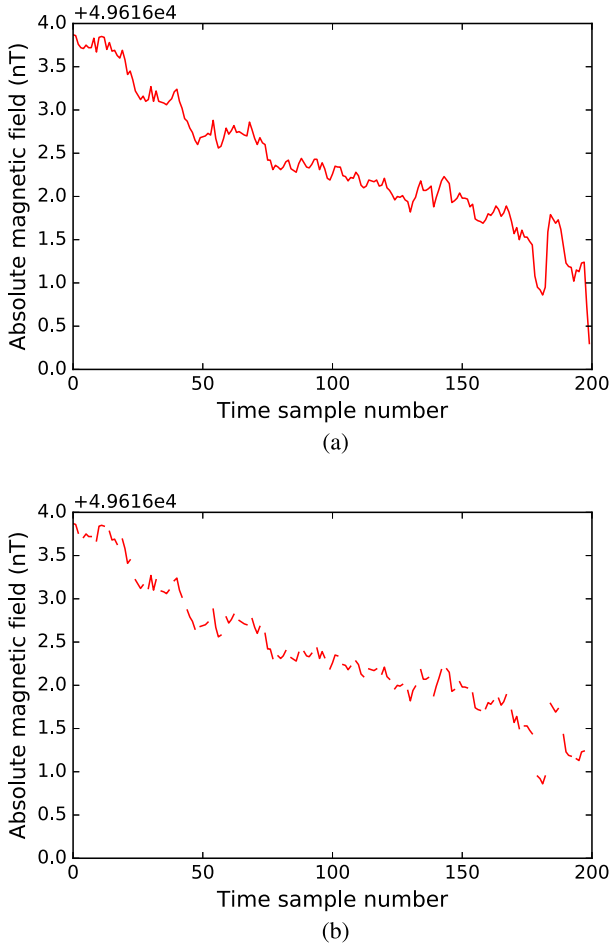


Fig. 7. 2-D field geomagnetic data for test. (a) Original 2-D field geomagnetic data. (b) Decimated data with 30% regular missing traces.

The method of establishing the database for 3-D is different with that of 2-D because it has two spatial dimensions (refer to i and j) describing all locations of the data sources. The original data can be expressed as a sequence of images where each magnetic field strength in each position is represented by the corresponding sequential pixels along the time axis, while i and j are the number of pixels along the row and column axes, respectively. In this case, the input variables $\boldsymbol{x} = \{\{x_{11}, x_{21}, \ldots, x_{i1}\}; \ldots; \{x_{1j}, x_{2j}, \ldots, x_{ij}\}\}$. First, the 3-D sequential data are reshaped into numerous 1-D vectors. After this, transform $\boldsymbol{x}$ to a 1-D space sorted from x_{11} to x_{ij}, and a 1-D column vector $[x_{11}, x_{12}, \ldots, x_{ij}]^T$ is obtained. Hence, each sample of reshaped data is an ensemble of pixels from original data. And the number of samples is the width i multiplied by the height j of the image.

Consequently, the original 2-D or 3-D geomagnetic data are converted into an array, where the feature $\boldsymbol{x}$ stands for the value (refer to B_t) of magnetic field at a given location and the ground truth $\boldsymbol{y}$ is the value (refer to B_{t+1}) of magnetic field at the next location. Each array is a set of adjacent data value. In addition, one thing needs to be emphasized is that the 2-D or 3-D geomagnetic data have their own database $\boldsymbol{B}$ (training separately), which is built based on their own time sequential t during measurement.

B. Training Model

For reconstruction of missing geomagnetic traces, we utilized three classic machine learning models and one typical deep learning model to learn from the data instances with multiple labels. The instances with a particular event were labeled as input variables based on machine learning models. Four different regression models mentioned before were used and evaluated in order to find a relative optimist model with better performance. Moreover, numerous comparison experiments using different geomagnetic data sets were conducted to these trained models. All parameter definitions for each machine learning model using Python are summarized in Table I.

1) Training: In general, the training procedure is the critical stage in machine learning models. In this case, 70% of the geomagnetic data set was used for training that as input variables to the regression model for intrinsic parameters selection. Meanwhile, 15% of the geomagnetic data set was reserved for cross validation and the last 15% for testing the performance of

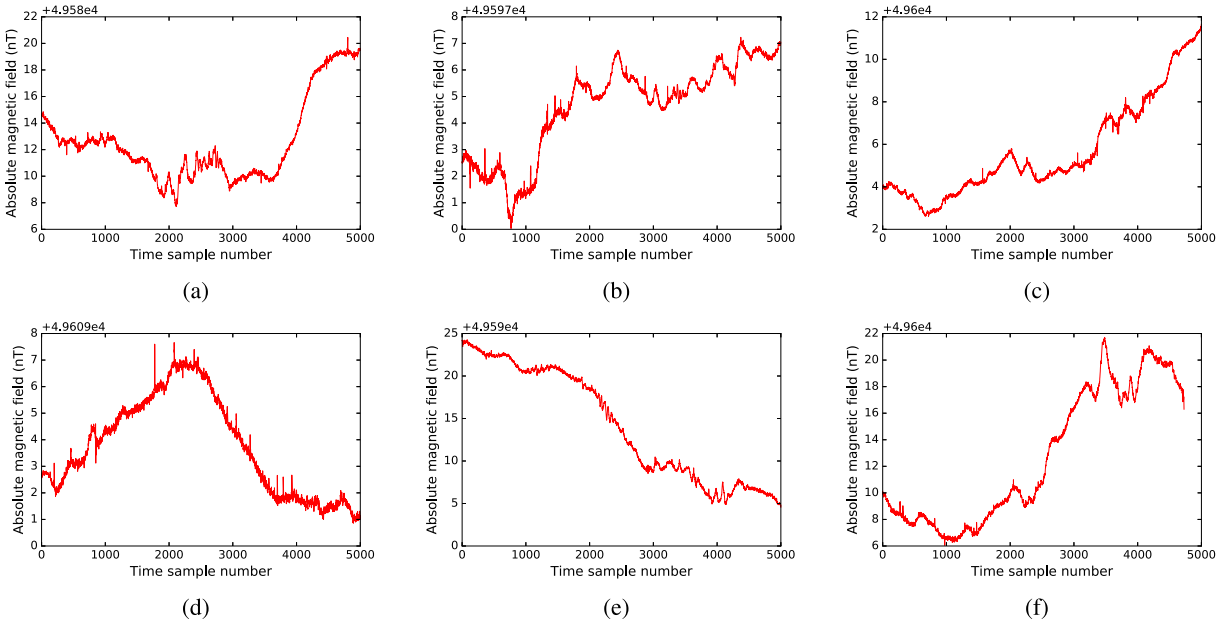


Fig. 8. Six 2-D field geomagnetic data used to feed into the training model. (a) 1st dataset. (b) 2nd dataset. (c) 3rd dataset. (d) 4th dataset. (e) 5th dataset. (f) 6th dataset.

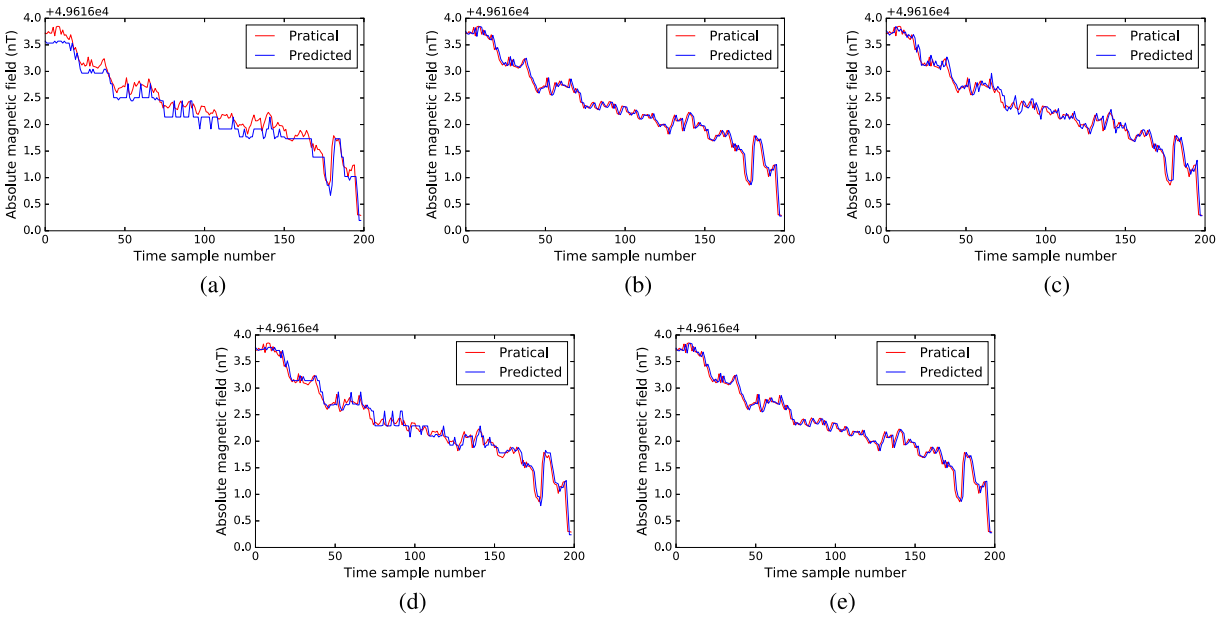


Fig. 9. Reconstruction results of different models. (a) Linear. (b) SVM. (c) Random forests. (d) Gradient boosting. (e) LSTM.

each model. The relation between the missing and completed traces in this training data set could be learned by utilizing machine learning algorithms. Then, a trained model could be used to predict the values of missing traces.

2) Validation: Basically, after the model training has been finished based on the training data set, a cross validation was implemented to eliminate the overfitting problem. As mentioned before, 15% of the geomagnetic data set was used as the validation data set. The efficacy of the model could be seen by using this data set to cross validate each trained model. If the performance of the trained models was not adequate on the validation data set, a set of parameters should be further modified in order to improve the performance of these models. Then, each of the trained models was retrained, and the revalidations were implemented on the same data sets of the retrained models, respectively. This validation process would be stopped as long as the machine learning models reached an adequate performance.

3) Testing: When the process of training and validation was finished, all the regression models mentioned before could be obtained. Thus, the testing data set was used to further evaluate the performance of each model. 15% of the geomagnetic data set was reserved for testing. Since the testing data were never used, this is an honest and efficient measure to ensure all the trained models' predictive capacity.

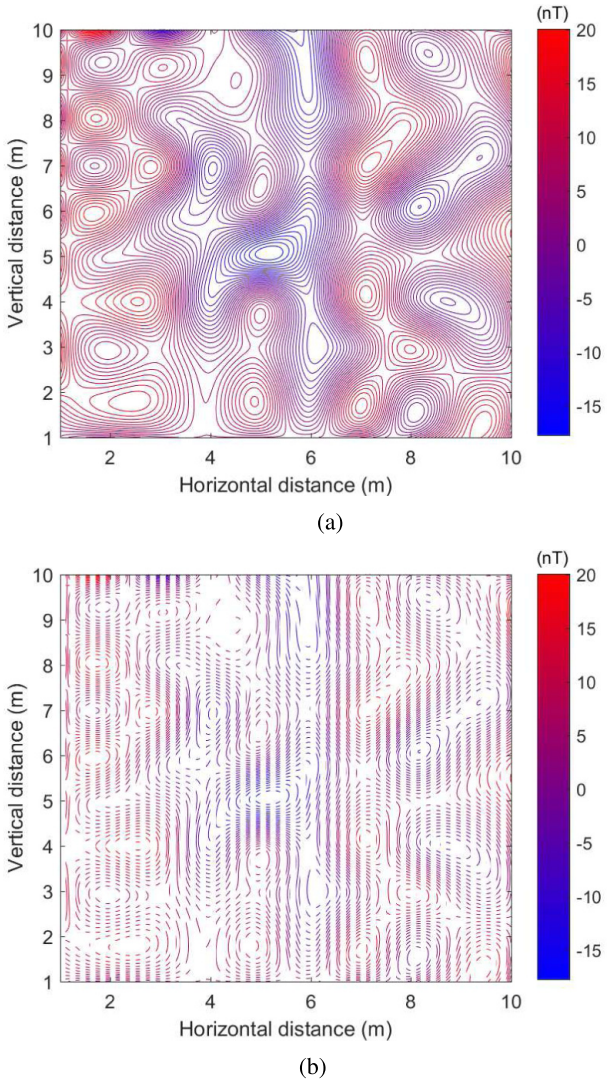


Fig. 10. 3-D field geomagnetic data for test. (a) Original 3-D field geomagnetic data. (b) Decimated data with 50% regular missing traces.

4) Performance Evaluation: Aimed to further evaluate the adaptability of the trained models, several new field geomagnetic data sets that were not trained before were selected to further test, and were fed into each of the obtained regression models, respectively. Unlike the testing sets mentioned before, these data sets have not been trained before. Therefore, this is a strong way to compare and evaluate the performance of different regression models. All the general procedure of training a regression model are summarized in Fig. 6. We observe that the different steps, including attribute extraction, training, testing, and evaluation, were conducted in developing the reconstruction machine learning models. Once the models were identified and validated, the experimental data should be fed into each of the trained machine learning models for reconstruction and subsequent interpolation.

C. 2-D Example

A 2-D completed field geomagnetic data are shown in Fig. 7(a), which is a subset of the data set that was recorded by a commercial magnetometer in the field for 24 h. In this case, we implemented a comparison of our results with the commonly used linear reconstruction method in a case of 30% regular missing traces (blank spaces), as shown in Fig. 7(b). Fig. 8 shows another six geomagnetic data sets without missing traces, from which 29 728 training point pairs of feature vectors and labels can be extracted. These are six patches from a large-field data set.

TABLE II
COMPARISON RESULTS USING DIFFERENT METHODS FOR 2-D DATA

Model	dataset	R-squared	RMSE
Linear	Testing	**0.788**	**0.279**
SVM	Training	0.899	0.188
	Validation	0.889	0.198
	Testing	0.878	0.208
Random Forests	Training	0.896	0.179
	Validation	0.885	0.200
	Testing	0.873	0.220
Gradient Boosting	Training	0.893	0.185
	Validation	0.883	0.204
	Testing	0.872	0.223
LSTM	Training	**0.988**	**0.094**
	Validation	**0.983**	**0.108**
	Testing	**0.978**	**0.122**

The original geomagnetic data are a time-series data that can be perceived as a sequence. After the preprocessing mentioned in Fig. 5, each sample of pairwise data is an ensemble of variables along the time axis from the original data. As a consequence, we got 29 728 pairwise samples totally. In this case, each sample is a 1-D vector, which is a suitable data form for the input of each model.

Fig. 9 shows the reconstruction results of the geomagnetic data from Fig. 7(b). By observing the comparison results using traditional linear, SVM, random forests, gradient boosting, and LSTM methods, we can conclude that the proposed machine learning and deep learning-based methods were slightly better than the commonly used linear regression method, and the predicted data followed approximately the same trend as the experimental data in all cases, especially using SVM and LSTM. The predicted values were almost the same as the experimental values. However, the predictions were not accurate in every corresponding measurement point, such as using random forest and gradient boosting methods.

To make the results in reconstruction quality more convincing, the RMSE and R-squared by using different methods were recorded in Table II, where the best and worst results were both highlighted in bold font. Among these five methods, LSTM got the best quality with training, validation, and testing data sets. The R-square and RMSE gave better results. The calculated values of R-square and RMSE in training phase by LSTM were 0.988 and 0.094, respectively. The cross validation was used to eliminate the overfitting problem. The calculated values of R-square and RMSE in testing phase by LSTM were 0.978 and 0.122, respectively. The results are worse than the training data set; however, when compared

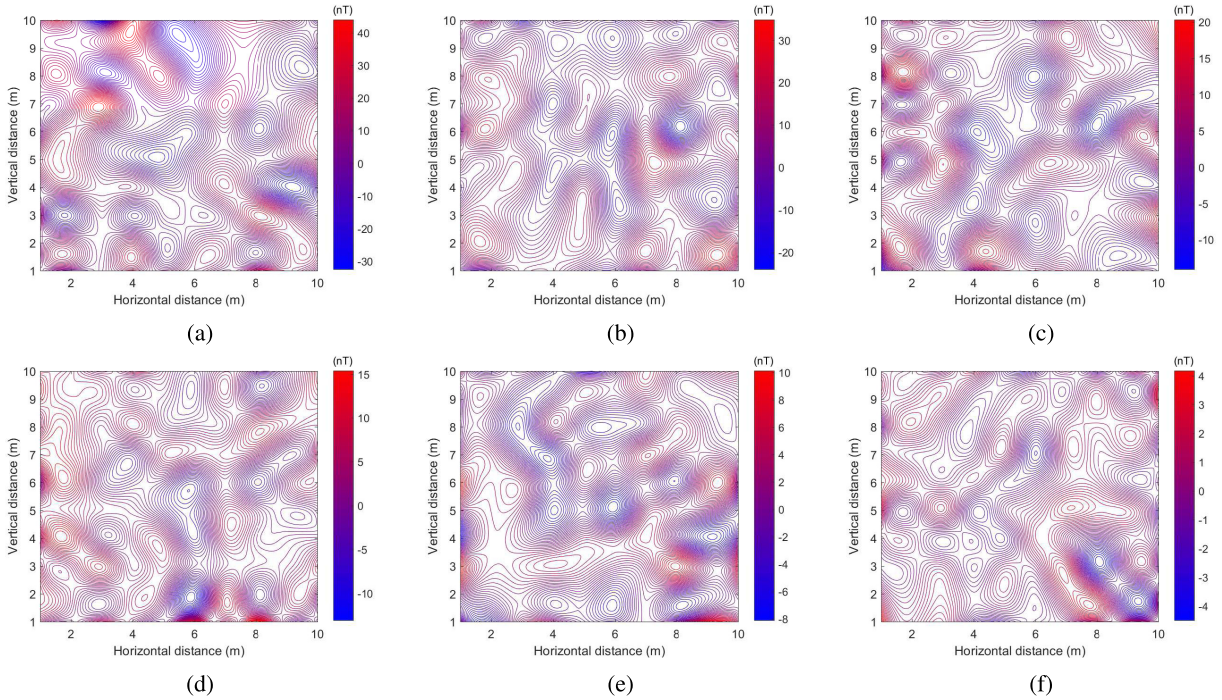


Fig. 11. Six 3-D field geomagnetic data used to feed into the training model. (a) 1st dataset. (b) 2nd dataset. (c) 3rd dataset. (d) 4th dataset. (e) 5th dataset. (f) 6th dataset.

with linear, SVM, and so on, it demonstrated a significant improvement.

D. 3-D Example

To better demonstrate the reconstructive capacity of our new methods, a series of experiments was conducted on 3-D field geomagnetic data sets. We chose different grounds of the measurement region to make sure the diversity and stochastic of the data. The length and width of the experiment region were all 10 m, and the measure step distance was 0.1 m, which means that 100 points should be measured in each region. Fig. 10(a) shows a 3-D completed field geomagnetic data with size 10 m $\times$ 10 m. The geomagnetic data have been downsampled with a regular 50% missing traces, which is shown in Fig. 10(b). To construct the regression model, we selected another six field geomagnetic data sets that have different morphological structures, as shown in Fig. 11. As a consequence, we got $100 \times 100 \times 6$ pairwise samples as the training data set. Each sample is a 1-D vector, which is a suitable data form for the input of each model.

Fig. 12 shows the reconstruction results and their trace comparison obtained using linear, SVM, gradient boosting, random forests, and LSTM methods. Our proposed machine learning and deep learning-based methods were also better than the linear regression method as a whole, especially in the area around coordinate (7, 6). However, for these proposed approaches, the reconstructions were not accurate in each corresponding missing data, such as the area around coordinate (4, 7) in Fig. 12(b), (9, 2) and (9, 9) in Fig. 12(c), (4, 8) in Fig. 12(d), and so on. These predicted errors could be due to the different features of each training model. Overall, the predicted deviations using linear reconstruction were larger than others, while the predictions using LSTM were almost the same as the experimental values, as shown in Fig. 10(a).

TABLE III
COMPARISON RESULTS USING DIFFERENT METHODS FOR 3-D DATA

Model	dataset	R-squared	RMSE
Linear	Testing	**0.778**	**0.283**
SVM	Training	0.903	0.179
	Validation	0.895	0.190
	Testing	0.887	0.201
Random Forests	Training	0.883	0.192
	Validation	0.876	0.212
	Testing	0.869	0.232
Gradient Boosting	Training	0.879	0.213
	Validation	0.869	0.219
	Testing	0.858	0.225
LSTM	Training	**0.972**	**0.124**
	Validation	**0.965**	**0.143**
	Testing	**0.958**	**0.162**

Table III shows the quantitative comparison in this case. Both the best and worst results were highlighted in bold font. For all these five methods, LSTM got the best quality with training, validation, as well as testing data sets, which were consistent with the results of 2-D example. The calculated values of R-square and RMSE in training phase by LSTM were 0.972 and 0.124, respectively, while in testing phase were 0.958 and 0.162, respectively. The results were worse than that of 2-D example overall because the proportional of missing traces in 3-D data was larger. However, it demonstrated the better performance significantly by using the machine learning or deep learning method compared with the linear

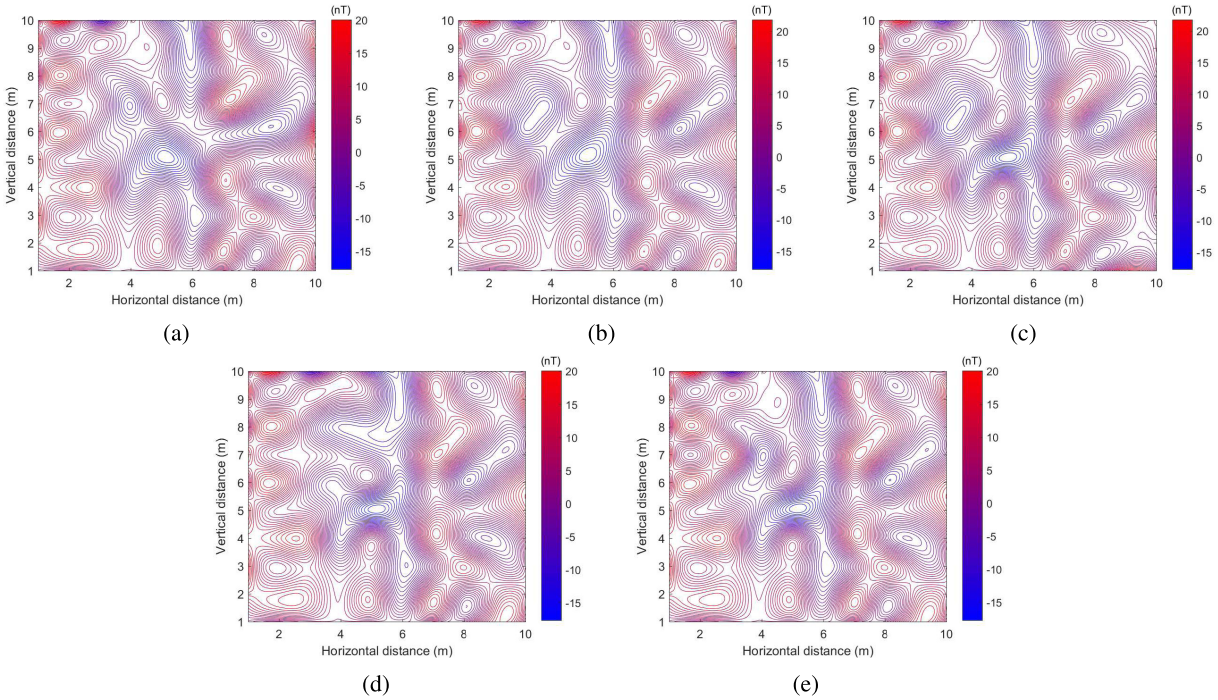


Fig. 12. Reconstruction results of different models. (a) Linear. (b) SVM. (c) Random forests. (d) Gradient boosting. (e) LSTM.

TABLE IV
COMPARISON OF COMPUTATIONAL TIME AND RAM CONSUMPTION EXAMINED USING DIFFERENT METHODS FOR TRAINING AND TESTING

Model	Procedure	Time (s)	RAM (MB)
SVM	Training	46.02 ± 1.15	51.23
	Testing	**0.76 ± 0.02**	
Random Forests	Training	16.79 ± 0.62	36.76
	Testing	0.12 ± 0.01	
Gradient Boosting	Training	**3.33 ± 0.03**	**2.11**
	Testing	**0.014 ± 0.002**	
LSTM	Training	**913.94 ± 74.32**	**126.79**
	Testing	0.51 ± 0.09	

regression method, especially LSTM. In addition, our method can deal with reconstruction problems of real field data more intelligently without setting complex parameters.

The most significant advantage of machine learning is that it can intelligently accomplish tasks assigned by a human, thereby reducing the time cost and manual workloads dramatically. However, in a practical situation, the machine learning or deep learning training step is extremely time- and memory-consuming. To be specific, in 3-D example, if all of the training pairs (60 000) are used in the training stage, the reconstruction experiment takes around 900 s and 130-MB RAM when using the LSTM method. Table IV shows the comparison of computational time and RAM consumption examined using different reconstruction models. The most and least running time for training, testing, as well as the RAM were highlighted in bold font, respectively. The LSTM took the most running time (913.94 ± 74.32 s) and RAM (126.79 MB), while gradient boosting took the least (3.33 ± 0.03 s and 2.11 MB, respectively) in the training procedure. In terms of testing, the SVM took the most running time (0.76 ± 0.02 s), while gradient boosting took the least (0.014 ± 0.002 s). Because the consumed RAM in the testing stage was trivial, we do not show the results here. To sum up, the running time was all no more than one second in the testing procedure. In general, when the regression model has been trained, it can be saved for future use in analyzing geomagnetic data. Therefore, although training models, especially LSTM, cost extensive computing resources, this factor can be ignored if a set of regression models is completed and saved.

IV. CONCLUSION

In this paper, we proposed both the machine learning and deep learning methods for geomagnetic data reconstruction. A hidden relationship (continuous hyperplane) can be determined using sufficient exemplified training sets, from which the missing data can also be derived. We present the RNN method to improve the performance of the classic machine learning method (SVM, gradient boosting, and random forests). Besides, the LSTM-based approach allowed us to avoid previous drawbacks in the existing reconstruction methods, and it is generally applicable to different data sets. Furthermore, the trained regression model can be saved for future use to reconstruct the geomagnetic data with similar geomorphological structure. Overall, the experimental results showed that the proposed method can achieve a reconstruction accuracy higher than 90%, which showed an increased by about 20% than the traditional method. The deep learning methods showed the high accuracy, while it still can be

improved as they are being further developed and many interesting ideas are emerging.

In the future work, we will also investigate the application of ensemble methods for geomagnetic data processing, which can integrate multiple machine learning models and assign each model with a weighting factor, such that different methods are assigned to apply on their better performed data, so the accuracy can be improved.

REFERENCES

[1] S. Khomutov, V. Sapunov, A. Denisov, D. Savelyev, and I. Babakhanov, "Overhauser vector magnetometer POS-4: Results of continuous measurements during 2015–2016 at geophysical observatory 'Paratunka' of IKIR FEB RAS, Kamchatka, Russia," in *Proc. E3S Web Conf.*, vol. 11. Les Ulis, France: EDP Sciences, 2016, Art. no. 00007.

[2] H. Dong, H. Liu, J. Ge, Z. Yuan, and Z. Zhao, "A high-precision frequency measurement algorithm for FID signal of proton magnetometer," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 4, pp. 898–904, Apr. 2016.

[3] R. M. Winslow *et al.*, "Observations of Mercury's northern cusp region with MESSENGER's Magnetometer," *Geophys. Res. Lett.*, vol. 39, no. 8, p. L08112, 2012.

[4] F. Lhuillier, J. Aubert, and G. Hulot, "Earth's dynamo limit of predictability controlled by magnetic dissipation," *Geophys. J. Int.*, vol. 186, no. 2, pp. 492–508, 2011.

[5] G. Hulot, F. Lhuillier, and J. Aubert, "Earth's dynamo limit of predictability," *Geophys. Res. Lett.*, vol. 37, no. 6, p. L06305, 2010.

[6] M. Dumberry and C. C. Finlay, "Eastward and westward drift of the Earth's magnetic field for the last three millennia," *Earth Planet. Sci. Lett.*, vol. 254, nos. 1–2, pp. 146–157, 2007.

[7] M. Korte, F. Donadini, and C. Constable, "Geomagnetic field for 0–3 ka: 2. A new series of time-varying global models," *Geochem., Geophys., Geosyst.*, vol. 10, no. 6, p. Q06008, 2009.

[8] I. Wardinski and M. Korte, "The evolution of the core-surface flow over the last seven thousands years," *J. Geophys. Res., Solid Earth*, vol. 113, no. B5, p. B05101, 2008.

[9] M. Korte and R. Holme, "On the persistence of geomagnetic flux lobes in global Holocene field models," *Phys. Earth Planet. Interiors*, vol. 182, nos. 3–4, pp. 179–186, 2010.

[10] M. Korte, C. Constable, F. Donadini, and R. Holme, "Reconstructing the Holocene geomagnetic field," *Earth Planet. Sci. Lett.*, vol. 312, no. 3, pp. 497–505, 2011.

[11] A. Nilsson, R. Holme, M. Korte, N. Suttie, and M. Hill, "Reconstructing Holocene geomagnetic field variation: New methods, models and implications," *Geophys. J. Int.*, vol. 198, no. 1, pp. 229–248, 2014.

[12] L. Kapper, F. Donadini, V. Serneels, E. Tema, A. Goguitchaichvili, and J. J. Morales, "Reconstructing the geomagnetic field in West Africa: First absolute intensity results from Burkina Faso," *Sci. Rep.*, vol. 7, Mar. 2017, Art. no. 45225.

[13] H. Liu, S. Liu, Z. Liu, N. Mrad, and H. Dong, "Prognostics of damage growth in composite materials using machine learning techniques," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2017, pp. 1042–1047.

[14] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: Machine learning and application to spontaneous behavior," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 568–573.

[15] A. M. Prasad, L. R. Iverson, and A. Liaw, "Newer classification and regression tree techniques: Bagging and random forests for ecological prediction," *Ecosystems*, vol. 9, no. 2, pp. 181–199, 2006.

[16] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied Linear Statistical Models*, vol. 4. Chicago, IL, USA: Irwin Chicago, 1996.

[17] S. K. Murthy, "Automatic construction of decision trees from data: A multi-disciplinary survey," *Data Mining Knowl. Discovery*, vol. 2, no. 4, pp. 345–389, Dec. 1998.

[18] O. Chapelle and V. Vapnik, "Model selection for support vector machines," in *Proc. NIPS*, 1999, pp. 230–236.

[19] S. Haykin and N. Network, "A comprehensive foundation," *Neural Netw.*, vol. 2, p. 41, Feb. 2004.

[20] D. M. Dutton and G. V. Conroy, "A review of machine learning," *Knowl. Eng. Rev.*, vol. 12, no. 4, pp. 341–367, 1997.

[21] Y. Jia and J. Ma, "What can machine learning do for seismic data processing? An interpolation application," *Geophysics*, vol. 82, no. 3, pp. V163–V177, 2017.

[22] J.-S. Lim, "Reservoir properties determination using fuzzy logic and neural networks from well data in offshore Korea," *J. Petroleum Sci. Eng.*, vol. 49, nos. 3–4, pp. 182–192, 2005.

[23] T. Helmy, A. Fatai, and K. Faisal, "Hybrid computational models for the characterization of oil and gas reservoirs," *Expert Syst. Appl.*, vol. 37, no. 7, pp. 5353–5363, 2010.

[24] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.

[25] S.-X. Yang, Y. Cao, D. Liu, and C.-F. Huang, "RS-SVM forecasting model and power supply-demand forecast," *J. Central South Univ. Technol.*, vol. 18, no. 6, pp. 2074–2079, 2011.

[26] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[27] X. Ma, C. Ding, S. Luan, Y. Wang, and Y. Wang, "Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2303–2310, Sep. 2017.

[28] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[29] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[30] C. Goller and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 1, Jun. 1996, pp. 347–352.

[31] K. Cho *et al.* (Sep. 2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." [Online]. Available: https://arxiv.org/abs/1406.1078

[32] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, vol. 2, 2010, p. 3.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] O. Tilk and T. Alumäe, "LSTM for punctuation restoration in speech transcripts," in *Proc. INTERSPEECH*, 2015, pp. 683–687.

[35] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, vol. 3, no. 1, pp. 115–143, 2003.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[37] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.

[38] H. Sak, A. Senior, and F. Beaufays. (Feb. 2014). "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition." [Online]. Available: https://arxiv.org/abs/1402.1128

[39] K. Wagstaff. (Jun. 2012). "Machine learning that matters." [Online]. Available: https://arxiv.org/abs/1206.4656

Huan Liu (S'15–M'18) received the Ph.D. degree in geodetection and information technology from the Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan, China, in 2018.

From 2016 to 2017, he was a joint training Ph.D. Student in electrical engineering and computer science at the School of Engineering, The University of British Columbia Okanagan Campus, Kelowna, BC, Canada. He has been involved in developing intelligent geophysical instruments, especially, the proton magnetometer and the Overhauser magnetometer. He is currently an Associate Professor with the School of Automation, China University of Geosciences. His research interests include weak magnetic detection, signal processing, data mining, and machine learning.

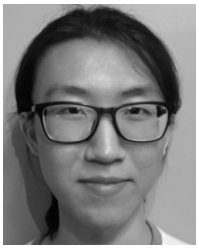
Zheng Liu (S'99–M'02–SM'06) received the Ph.D. degree in engineering from Kyoto University, Kyoto, Japan, in 2000, and the Ph.D. degree from the University of Ottawa, Ottawa, ON, Canada, in 2007.

From 2000 to 2001, he was a Research Fellow with Nanyang Technological University, Singapore. He then joined the Institute for Aerospace Research (IAR), National Research Council Canada, Ottawa, ON, Canada, as a Governmental Laboratory Visiting Fellow nominated by Natural Sciences and Engineering Research Council. After being with IAR for five years, he transferred to the National Research Council Institute for Research in Construction, where he was a Research Officer. From 2012 to 2015, he was a Full Professor with the Toyota Technological Institute, Nagoya, Japan. He is currently with the School of Engineering, The University of British Columbia Okanagan Campus, Kelowna, BC, Canada. His research interests include image/data fusion, computer vision, pattern recognition, senor/sensor network, condition-based maintenance, and nondestructive inspection and evaluation.

Dr. Liu is a member of International Society for Optics and Photonics. He is the Chair of the IEEE Instrumentation and Measurement Society Technical Committee on Industrial Inspection (TC-36). He holds a Professional Engineer License in British Columbia and Ontario. He serves on the Editorial Board for the journals, such as the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, the *IEEE Instrumentation and Measurement Magazine*, the IEEE JOURNAL OF RADIO FREQUENCY IDENTIFICATION, *Information Fusion*, *Machine Vision and Applications*, and *Intelligent Industrial Systems*.

Shuo Liu (S'16) received the B.S. degree in computer science from the Beijing University of Technology, Beijing, China, in 2015. He is currently pursuing the M.A.Sc. degree in electrical engineering and computer science with the School of Engineering, The University of British Columbia Okanagan Campus, Kelowna, BC, Canada.

His research interests include computer vision, machine learning, image fusion, and image translation.



Yihao Liu is currently pursuing the B.S. degree major in electrical engineering and minor in computer science with the School of Engineering, The University of British Columbia Okanagan Campus, Kelowna, BC, Canada.

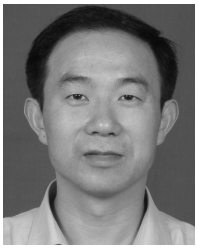
His research interests include machine learning, data science, and nondestructive testing.

Junchi Bin received the B.S. degree from the School of Electronic Control, Chang'an University, Xi'an, China, in 2016. He is currently pursuing the M.A.Sc. degree in electrical engineering and computer science with the School of Engineering, The University of British Columbia Okanagan Campus, Kelowna, BC, Canada.

His research interests include urban computing, machine learning, and data fusion.

Fang Shi received the B.S. degree in electrical engineering from Fuzhou University, Fujian, China, in 2016. She is currently pursuing the M.A.Sc. degree in electrical engineering and computer science with the School of Engineering, The University of British Columbia Okanagan Campus, Kelowna, BC, Canada.

Her research interests include data analytics, machine learning, and nondestructive evaluation.



Haobin Dong received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2002.

He was a Visiting Associate Professor with the Well Logging Laboratory and the Subsurface Sensing Laboratory, Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA, from 2005 to 2006. He is currently a Professor with the School of Automation, China University of Geosciences, Wuhan. His research interests include weak signal detection and intelligent geophysical instrument.