Detecting Weak Physical Signal from Noise: A Machine-Learning Approach with Applications to Magnetic-Anomaly-Guided Navigation

Zheng-Meng Zhai, Mohammadamin Moradi, Ling-Wei Kong, and Ying-Cheng Lai

Phys. Rev. Applied **19**, 034030 — Published 9 March 2023

DOI: 10.1103/PhysRevApplied.19.034030

Detecting weak physical signal from noise: A machine-learning approach with applications to magnetic-anomaly-guided navigation

Zheng-Meng Zhai,[1] Mohammadamin Moradi,[1] Ling-Wei Kong,[1] and Ying-Cheng Lai[1, 2, *]
[1] *School of Electrical, Computer and Energy Engineering,*
Arizona State University, Tempe, AZ 85287, USA
[2] *Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*
(Dated: February 14, 2023)

Detecting a weak physical signal immersed in overwhelming noises entails separating the two, a task for which machine learning is naturally suited. In principle, such a signal is generated by a nonlinear dynamical system of intrinsically high dimension for which a mathematical model is not available, rendering unsuitable traditional linear or nonlinear state-estimation methods that require an accurate system model (e.g., extended Kalman filters). We exploit the architectures of reservoir computing and feed-forward neural networks (FNNs) with time-delayed inputs to solve the weak signal detection problem. As a prototypical example, we apply the machine-learning schemes to earth magnetic anomaly field-based navigation. In particular, the time series are collected from the interior of the cockpit of a flying aircraft during different maneuvering phases, where the overwhelmingly strong noise background is the result of other components of the earth magnetic field and the fields generated by the electronic devices in the cockpit. We demonstrate that, when combined with the traditional Tolles-Lawson model for the earth magnetic field, the articulated machine-learning schemes are effective for accurately detecting the weak anomaly field from the noisy time series. The schemes can be applied to detecting weak signals in other domains of science and engineering.

I. INTRODUCTION

Detecting a weak physical signal from strong noises is a challenging problem in many applications. Take, for example, modern navigation based on the Global Positioning System (GPS). The GPS signals, due to their weak intensity and the need to travel through vast distances, are vulnerable to external interference such as jamming and spoofing [1]. Because of the fragility of GPS, there has been some recent interest in developing navigational systems as alternatives to the GPS. A promising idea is to exploit earth's magnetic fields as a means to detect the instantaneous locations of a moving object, e.g., an aircraft [2, 3]. The underlying physical principle is that the intensity of the magnetic field exuding from the earth's surface depends on the location as characterized by the magnetic anomaly maps. Magnetic sensors or magnetometers attached to different parts of an aircraft could then be used to detect the strength of the magnetic field, thereby providing the locations of the airplane in reference to the magnetic anomaly maps. A great difficulty is that the earth anomaly field is weak and the magnetic signals collected by the magnetometers are noisy. Compounding this difficulty is the various types of electronic equipment and devices embedded in the cockpit, which generate all kinds of electromagnetic noises that can not only interfere with, but also overwhelm the desired magnetic signal. For the idea of magnetic navigation to be feasible, to develop effective schemes to mitigate/remove strong noise so as to extract the weak magnetic anomaly signal is essential. We note that, for magnetic signals collected from the interior of the cockpit of an airplane, jamming is usually not of concern because there is no need to send the signals through a large distance.

The need to develop effective methods to remove "noisy magnetic fields" to detect the real magnetic anomaly signal can be further justified, as follows. The Tolles-Lawson (TL) model [4–6] provides a means to isolate the aircraft magnetic field so that it can be removed from the total magnetic field, ideally yielding only the earth magnetic anomaly field that can be exploited for navigation. This process is based on examining the permanent, induced, and eddy current aircraft magnetic fields and using band-pass filtered measurements from an additional magnetometer to remove these fields. However, this model works only when the magnetic field from the aircraft is weak enough relative to that from the earth at the sensor. One way to weaken the disturbing magnetic fields is to place the magnetometer onto the tail stinger of the aircraft, but this may not be practical. Alternatively, the magnetometer can be placed inside the cockpit. The difficulty is then that the magnetometer is now close to multiple magnetic interference sources within the aircraft. A challenge is to detect the real magnetic anomaly signal from the noisy signals.

In this paper, we develop a general machine-learning framework to address the problem of detecting extraordinarily weak electromagnetic signals from an overwhelmingly strong noisy environment. The basic principle underlying our work is that noise mitigation to detect the magnetic anomaly signal can be viewed as a nonlinear signal filtering problem that can be solved by employing modern machine learning. The two distinct features of the earth's magnetic anomaly signal are its weak strength and vulnerability to strong interference such as in-band and overwhelming noises. A suitably trained neural ma-

---

[*] Ying-Cheng.Lai@asu.edu

chine with features of the desired signal would view the disturbances, e.g., cockpit noises, as undesired and remove them. The real data used in our work were from the Open Call for developing machine-learning approaches to signal enhancement for magnetic navigation (MagNav) Challenge organized in 2020 by the Air Force-MIT Artificial Intelligence Accelerator, where the objective was to use magnetometer readings recorded from within a cockpit and to remove the aircraft magnetic noise to yield a clean magnetic anomaly signal [7].

How to select a suitable machine-learning architecture to detect weak signals from strong noises? In this paper, we study two machine-learning schemes: reservoir computing and time-delayed feed-forward neural networks (FNNs), for the following reasons. Consider again the magnetic navigation problem. The magnetic field signal detected by a flying aircraft can effectively be regarded as a time series from a continuous-time nonlinear dynamical system. To detect such a signal from a noisy background based on machine learning, the neural networks should possess certain memory capacity. The traditional multilayer perceptrons (MLPs) [8] do not have memories, but recurrent neural networks do. Moreover, because of the nature of the nonlinear signal prediction and classification, reservoir computing [9–11], a class of recurrent neural networks, may be suited. Further justification comes from the fact that there have been significant recent efforts establishing reservoir computing as a powerful paradigm for chaotic time series and signal prediction [12–32]. For example, reservoir computing has been demonstrated to be effective at distinguishing and separating characteristically different chaotic signals [15, 31, 33]. As an alternative to reservoir computing, FNNs with time-delayed inputs, the so-called "next-generation reservoir computing" [29, 30, 34] is also a viable solution, which has a memory capacity and low computational cost. Combining each of the machine-learning methods with the standard Tolles-Lawson model, we show that the magnetic anomaly signal hidden in strong noises can be detected with high accuracy. Our success represents a necessary step forward in the development of magnetic-anomaly based navigation.

In Sec. II, we provide a brief overview of the physics background of magnetic-anomaly based navigation and existing signal-processing methods. The Tolles-Lawson (TL) model for magnetic calibration, reservoir computing and time-delayed FNN based machine-learning architectures, as well as a data description are presented in Sec. III. Testing results with the MagNav Challenge data are demonstrated in Sec. IV and the detection performance of reservoir computing and time-delayed FNNs are compared. A discussion and potential future studies are presented in Sec. V.

II. PHYSICS OF MAGNETIC-ANOMALY NAVIGATION AND CURRENT SIGNAL-PROCESSING METHODOLOGIES

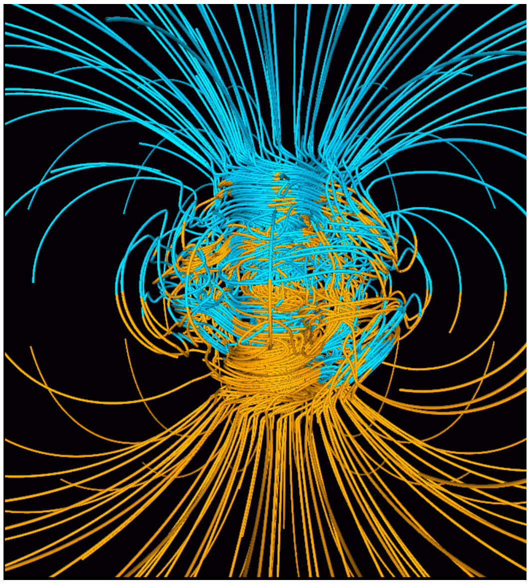

FIG. 1. Earth's magnetic field simulated in a period of normal polarity between reversals. Shown are the magnetic field lines, blue when the field points towards the center and yellow when pointing away [35] (from NASA image data sets in the public domain that are not subject to copyright restriction.)

The total magnetic field of the earth, as schematically illustrated in Fig. 1, is a linear superposition of several fields, each with a distinct physical origin. The dominant source is the core field. The north direction to which a compass points is almost entirely due to the core field whose magnitude ranges from 20 mT to 60 mT. The anomaly field with the magnitude of about 100 nT is the second source of the geomagnetic field, which is due to the permanent or induced magnetization of the rocks in the earth's crust. The strength of the anomaly field depends on the location, providing the possibility of exploiting this field for positioning and navigation. When collected from, e.g., an airplane, the anomaly field is effectively a time series signal. Whereas both the core and anomaly fields are generated inside the earth, the third source is the temporal variations generated externally due to the field contributions from the ionosphere, the magnetosphere, and the coupling currents between the two. Usually, the magnitude of this third field is ten times smaller than the anomaly field [36].

The magnetic anomaly field is a robust, accessible and reliable source of information that can be used for positioning. The fact that GPS signals are not accessible everywhere and are vulnerable to jamming [37] highlights the importance of developing magnetic field-based navigation. This is usually referred to as GNSS (Global Navigation Satellite System) denied environments, examples of which include indoor environments (e.g., under-ground bunkers) and underwater (e.g., submarines) [38–40]. Magnetic field aided navigation is passive and, because the strength of a magnetic field decreases with the inverse cubic power of the distance, it is practically impossible to jam a magnetic field-based navigation device from a distance by using another magnetic field [41]. In some applications, magnetic field measurements gathered from an array of magnetometers are incorporated into Inertial Navigation Systems (INS) to correct the drifting error, resulting in more reliable and robust navigation [38, 42–44]. Magnetic aided navigation can serve as a reliable backup plan in situations where GPS is denied [45]. The present positioning errors of magnetic navigation achieved based on actual flight test data are about 10 meters, which are larger than the GPS errors. The accuracy demonstrated in [45] was achieved by leveraging only passive sensors, providing support for the idea that the earth's magnetic field may be a viable approach for future GPS backup and alternative positioning systems for aircraft in flight. A significant challenge is that the magnetic anomaly field is weak [46]. As a result, navigation based on the anomaly signal is extremely vulnerable to external magnetic fields, e.g., those from the devices operating inside the aircraft. A well-defined signal-processing algorithm that detects the magnetic anomaly signals by removing the disturbance signals is needed for realizing magnetic navigation.

Our development of a machine-learning based method to detect a weak physical signal from overwhelming noises is based on the following considerations. The process of employing a stochastic process, e.g., the measurement process, to estimate another stochastic process, e.g., the signal process, has been a fundamental problem in signal processing. Typically, a filter is a device that removes unwanted components from a signal. A neural filter is a neural network trained with simulated data or experimental data to perform recursive processing. Since neural filtering is a data driven approach, no assumptions such as those about the specific dynamics, distribution or noise type are required. A properly trained neural network carries the most "informative" statistics in its dynamical state and approximates the optimal filtering performance to any accuracy [47]. Recursive neural-network based filtering is a measurement process to estimate a signal process and used in applications where the range of the measurement expands in time or is too large for recurrent neural networks to handle for the required filtering resolution [48]. While recursive neural networks are effectively hierarchical networks, recurrent neural filters or recurrent auto-encoder architectures that learn distinct representations for each Bayesian filtering step, captured by a series of encoders and decoders [49], are mainly used for sequential inputs where time is the main factor differentiating the components. In general, neural filters, which can be viewed as the extension of classical linear adaptive filters to deal with nonlinear modeling tasks [50], perform better than linear methods in dealing with stationary or non-stationary noise. In addition, neural filters are more computationally efficient and perform better [48, 51] compared to extended Kalman filters [52–56]. In fact, reservoir computing can also be used as a filter [57, 58].

III. DATA AND METHODS

A. Tolles-Lawson (TL) model for magnetic calibration

A challenge with magnetic navigation is that one can only measure the total magnetic field, which has several different components with magnitude much larger than the anomaly field needed for positioning. In particular, the total measured field $\vec{B}_{\mathrm{m}}$ can be written as [58]

$$\begin{aligned}\vec{B}_{\mathrm{m}} &= \vec{B}_{\mathrm{e}} + \vec{B}_{\mathrm{aircraft}} \\ &= \vec{B}_{\mathrm{core}} + \vec{B}_{\mathrm{anomaly}} + \vec{B}_{\mathrm{tv}} + \vec{B}_{\mathrm{aircraft}},\end{aligned} \tag{1}$$

where $\vec{B}_{\mathrm{e}}$ is the earth magnetic field that consists of three components: the core field $\vec{B}_{\mathrm{core}}$, the anomaly field $\vec{B}_{\mathrm{anomaly}}$, and temporally varying fields $\vec{B}_{\mathrm{tv}}$, while $\vec{B}_{\mathrm{aircraft}}$ represents the total field generated from the aircraft itself. The goal is to filter out all the other components and keep only the anomaly field. In this regard, it is relatively straightforward to remove the earth field components $\vec{B}_{\mathrm{core}}$ and $\vec{B}_{\mathrm{tv}}$. In particular, the core field $\vec{B}_{\mathrm{core}}$ can be calculated with the International Geomagnetic Reference Field (IGRF) coefficients, and the temporal variation field $\vec{B}_{\mathrm{tv}}$, which is mostly from the diurnal variations and space weather, can largely be removed using ground-based reference measurements [59]. It is thus the platform field $\vec{B}_{\mathrm{aircraft}}$ from the aircraft that becomes the focus of the problem. The goal of our work is to eliminate $\vec{B}_{\mathrm{aircraft}}$ from measured data using a combination of signal processing and machine learning.

The data sets used in our study (from the signal enhancement for magnetic navigation challenge problem [60]) consist of the time series of the measured magnetic field from five flights, each containing several segments (or lines) with 65 features measured in real time during each flight. There are in total five magnetometers placed on the aircraft, with one of them placed at the attached tail stinger, as shown in Fig. 2(a). After calibrated by the Tolles-Lawson model (described below), the reading of the magnetometer at the tail stinger is considered to be the real value of the earth field $\vec{B}_{\mathrm{e}}$, which serves as the training target in our supervised learning problem. The target signal is denoted as B_{SGL}. There are four other magnetometers placed at different positions in the cabin, as shown in Fig. 2(a), and 61 other various features recorded at the same time by some current and voltage sensors and readings from the INS system. (More details of these data can be found in Ref. [60].) Two examples of the magnectic-field map in the flight region are

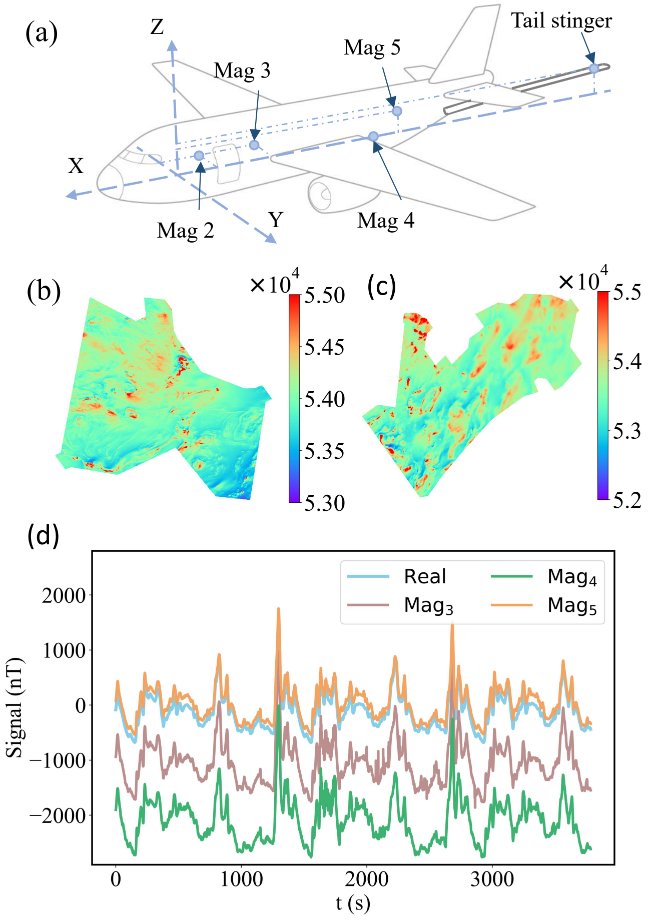

FIG. 2. Data collection scheme and representative time series of the measured magnetic field. (a) Positions of the magnetometers on the aircraft. One magnetometer (Mag 1) is placed at the tail stinger to reduce noises from the plane. After calibration using the TL model, the signal from Mag 1 is regarded as the real earth field $\vec{B}_{\mathrm{e}}$ and is used as the training target. Measurements from the other four magnetometers, together with various other signals, are used as inputs of the filtering problem. (b,c) Examples of the maps of the magnetic fields around the data-gathering flights. The units are nT. Subtracting this field by the core field calculated from IGRF yields the anomaly field. (d) Magnetic signals measured by different magnetometers in the plane.

shown in Figs. 2(b) and 2(c), and examples of the magnetic signals measured by different magnetometers in the plane are shown in Fig. 2(d). For the three additional field components in Eq. (1) other than $\vec{B}_{\mathrm{anomaly}}$, the two earth field components $\vec{B}_{\mathrm{core}}$ and $\vec{B}_{\mathrm{tv}}$ are already given in the data sets. The goal is thus to use the readings of these four magnetometers, together with other data features, to estimate the real earth field $\vec{B}_{\mathrm{e}}$ through minimizing some proper root mean square errors (E_{RMS}).

The TL model [58, 61, 62] is a classic method for magnetic calibration. It models the magnetic field generated by the body of the aircraft as three magnetic moments: the permanent, the induced, and the eddy current moments. The permanent magnetic moment represents the nearly constant magnetic moment of the entire aircraft. The induced magnetic moment is the magnetic response of the magnetically susceptible materials in the aircraft to the earth magnetic field. The eddy current moment is caused by the temporal variations of the earth magnetic field due to the motion of the aircraft. The TL correction terms can be expressed as

$$\begin{aligned} B_{\mathrm{TL}} &= B_{\mathrm{perm}} + B_{\mathrm{ind}} + B_{\mathrm{eddy}} \\ &= \vec{B}_{\mathrm{m}}^{T} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + |\vec{B}_{\mathrm{m}}| \vec{B}_{\mathrm{m}}^{T} \begin{bmatrix} x_4 & x_5 & x_6 \\ x_5 & x_7 & x_8 \\ x_6 & x_8 & x_9 \end{bmatrix} \vec{B}_{\mathrm{m}} \\ &+ |\vec{B}_{\mathrm{m}}| \vec{B}_{\mathrm{m}}^{T} \begin{bmatrix} x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} \\ x_{16} & x_{17} & x_{18} \end{bmatrix} \dot{\vec{B}}_{\mathrm{m}}, \end{aligned} \tag{2}$$

where x_1 to x_{18} are 18 constant coefficients that can be calculated [60] after a calibration flight [58, 61, 62]. The earth field $\vec{B}_{\mathrm{e}}$ calculated by the TL model can then be corrected by subtracting these terms from the total measured field $\vec{B}_{\mathrm{m}}$:

$$|\vec{B}_{\mathrm{e,\ TL}}| = |\vec{B}_{\mathrm{m}}| - B_{\mathrm{TL}}. \tag{3}$$

The TL model works well only when the flying aircraft is in a “magnetically quiet” mode and all the magnetometer measurements are performed on a tail stinger outside the cabin. During a normal flight mode without an additional tail stinger hanging outside of the plane, the TL model is not effective for magnetic calibration. In our work, we use the TL model only as a first-step processing to obtain the ground truth and exploit machine learning methods to perform further corrections.

B. Machine learning methods

1. *Reservoir computing*

The basic structure of a reservoir computer is shown in Fig. 3. The matrix W_{in} maps the input signal $u(t)$ into a high-dimensional state vector $r(t)$ of the neural network in the hidden layer. The state $r(t)$ is updated according to the input $u(t)$ and the state $r(t-\Delta t)$ at the previous time step, leading to the recurrent structure. The output matrix W_{out} maps $r(t)$ into the output layer to generate the output signal $v(t)$. In our problem of detecting the magnetic anomaly signal, $v(t)$ gives the (normalized) estimation of the anomaly field $\vec{B}_{\mathrm{anomaly}}$.

For reservoir computing, the intrinsic recurrent structure provides a basis to deal with the temporal dependency in the input time series. When the size of the hidden-layer network is adequately large and with optimal hyperparameter values determined by the input data through, e.g., Bayesian optimization, it has the ability to remove the noise from the multiple input features and yield a clean target anomaly field. The intrinsic memory

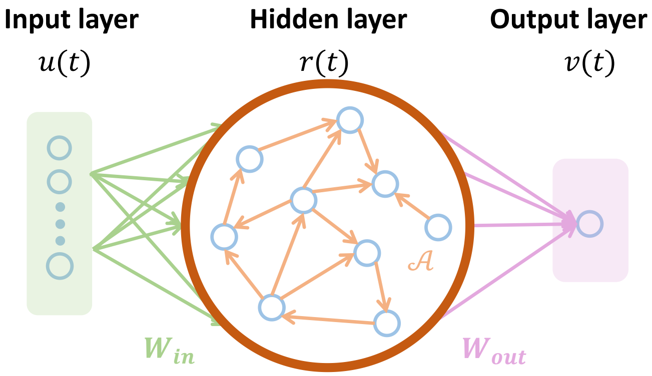

FIG. 3. Reservoir computing structure. A reservoir computer is composed of three layers: the input, the hidden layer, and the output. The input, state vector of the hidden-layer neural network, and the output are $u(t)$, $r(t)$ and $v(t)$, respectively. The matrices W_{in}, $\mathcal{A}$ and W_{out} represent the input weights, the network structure, and the output weights. The weights in W_{in} and $\mathcal{A}$ are fixed prior to training. The output matrix W_{out} is determined by training through a linear regression. The filtered signal is $v(t)$.

capacity embedded in the recurrent dynamics of the hidden layer guarantees that the machine output is a signal with the correct time history. Because a trained reservoir computer is a self-evolving nonlinear dynamical system, transient behaviors can arise, during which the prediction of the actual field can be quite inaccurate. It is thus necessary to remove the transients. More details about reservoir computing can be found in Appendix A 1, and the issue of transients is addressed in Appendix B.

2. *Feed-forward neural networks with time-delayed inputs*

We use artificial neural networks to process the selected features associated with the magnetic measurements to estimate the anomaly field. As described in Sec. I, we exploit a class of feed-forward neural networks with time-delayed inputs [29, 30] as the machine-learning architecture for detecting the magnetic anomaly field, as shown in Fig. 4. Different from recurrent neural networks, the nodes in an FNN do not form cycles so it can not form memory using its own internal states. To embed the memory information, we use the present and historical data $u(t-\tau),\dots,u(t-m\tau)$ as the inputs to the network, where $u(t)$ includes the magnetic field signals, fluxgate, and noises recorded by different sensors.

The reasons that we choose FNNs with time-delayed inputs are as follows. Because the noises contain both high-frequency and low-frequency components, it is difficult to use conventional filtering methods to remove these noises. Time-delayed inputs are introduced to address this critical issue, where the estimation of the anomaly field for a time step t is determined not only by the measured signals at that time step but also by the data from the recent past. We choose FNNs with time-delayed inputs, where the the intrinsic temporal dependencies in the input magnetic-field data are accounted for by imposing time delays.

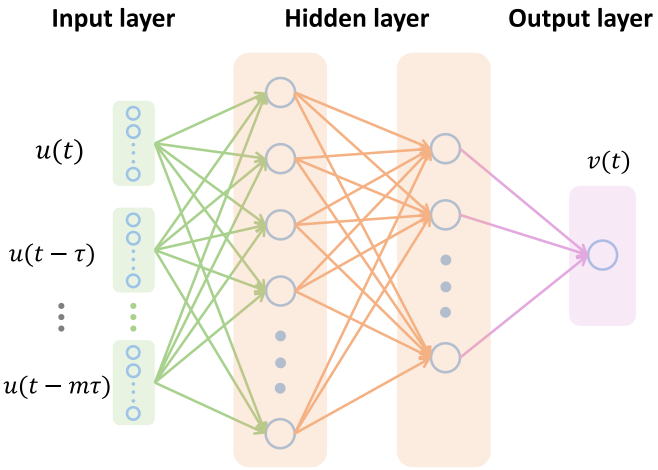

FIG. 4. The architecture of the proposed feed-forward neural network with time delayed inputs for detecting the anomaly earth magnetic field for navigation. The architecture includes three main components: the input layer, two hidden layers, and the output layer, where $u(t)$, $u(t-\tau)$, $\dots$, $u(t-m\tau)$ are the real-time and historical signal with τ being the time delay and m being the embedding dimension. The filtered signal is $v(t)$.

An FNN usually includes several hidden layers, each with dozens of nodes. An activation function is used to map the state from one hidden layer to the next in a nonlinear fashion.

We use the following hyperbolic tangent sigmoid (tansig(x_{tan})) as the activation function that compresses the output into the range $[-1,1]$:

$$\text{tansig}(x_{tan}) = \frac{2}{1+e^{-2x_{tan}}} - 1, \tag{4}$$

for all layers. The learning technique we choose for the FNN is scaled conjugate gradient backpropagation (SCG) [63]. (Details of FNN and this algorithm and FNN can be found in Appendix A 2.)

C. Data description

For all trials and features of the entire dataset, we use z-score normalization [64] to preprocess the time series so that they have zero mean and unit standard deviation. In particular, if the data x_z in each line has mean $\bar{x}_z$ and standard deviation S_z, the $z-$score normalization is given by $z=(x_z-\bar{x}_z)/S_z$. When training the machine learning models, we randomize the training length to accommodate realistic situations where the collected data sets can have varying lengths. Specifically, the training, validation and testing length are chosen to be 60%, 20% and 20% of the total length of the data line. For validation and testing, we use E_{RMS} to measure the performance:

$$E_{\mathrm{RMS}}(y, \hat{y}) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}[y(t) - \hat{y}(t)]^2}, \qquad (5)$$

where $y(t)$ and $\hat{y}(t)$ are the real and predicted signals, respectively. The E_{RMS}s are also averaged over different training length.

The sampling rate of the data, the average number of samples obtained in one second, is $f_s = 10\mathrm{Hz}$, so the time interval between two adjacent data points is $dt = 0.1\mathrm{s}$. The flight data lines do not have the same length, where the average length is 921s with the standard deviation $1,395$s. The average training, validation and testing lengths are 553s, 184s and 184s, respectively. More details about the data can be found in Appendix C.

IV. RESULTS

We train the machine-learning models on a computer with one RTX 4000 NVIDIA GPU, using MATLAB. For one flight line, the average training, validation and testing lengths are 5530, 1840, and 1840 data points, respectively. Since we have 45 lines of flight data, the total training, validation, and testing lengths are 248850, 82800, and 82800, respectively.

For this challenging task of nonlinear filtering of weak signals to be successful, it is essential that some historical information from the measurements is incorporated into the neural network architecture. For reservoir computing, once the optimal hyperparameter values have been determined through Bayesian optimization, it is computationally efficient provided that the size of the recurrent network in the hidden layer is reasonable (e.g., fewer than 1000 nodes). The time-delayed FNNs, by design, are computationally efficient. As mentioned, both machine-learning architectures intrinsically possess a memory capacity: through recurrent dynamics in reservoir computing and time-delayed inputs in FNN. Here we present results from reservoir computing, followed by those from time-delayed FNNs.

A. Filtering by reservoir computing

We reconstruct a reservoir computer by using a random network of size $n = 500$ in the hidden layer (a brief description of the reservoir computing architecture can be found in Appendix A 1). The hyperparameter values obtained through Bayesian optimization are: $\rho = 0.29$, $b = 0.75$, $\alpha = 0.47$, $\beta = 10^{-2.2}$, $p = 0.78$, and bias $= 0.32$. The dimension of the input signal $u(t)$ is $3 + 15 = 18$, where three components are the measurements from Mag_3, Mag_4 and Mag_5 [Fig. 2(a)], and the other 15 components are time series of various features with relative importance ranking calculated by a greedy algorithm, as displayed in Table I. The output dimension is one, corresponding to the anomaly field signal to be detected.

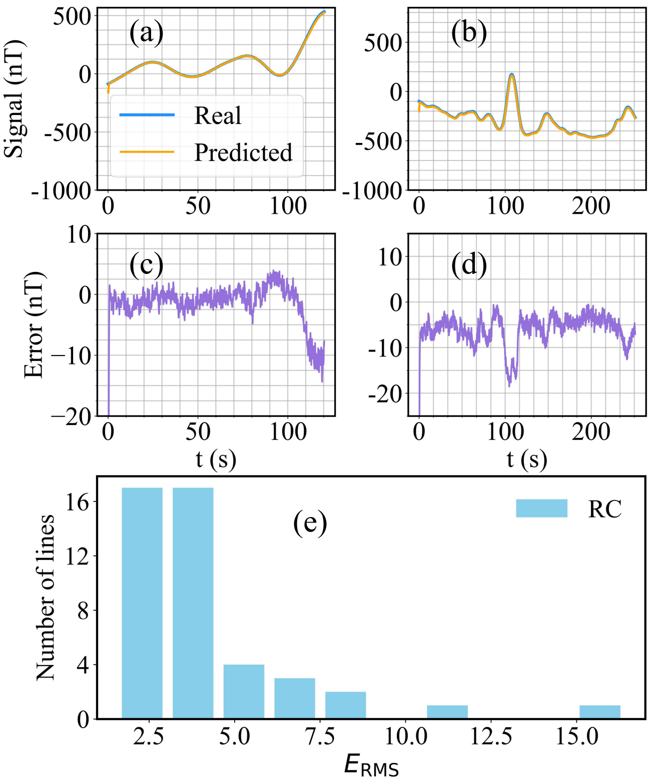

FIG. 5. Reservoir-computing based detection of anomaly magnetic field. (a,b) The real and predicted signals for lines 1004.4015 and 1006.07. (c,d) The errors between the real and predicted signals. Note that on this scale, the errors are not noticeable visually. (e) The E_{RMS} distribution.

We train and test 45 lines of flight data. Results

TABLE I. Importance ranking of the features selected by a greedy algorithm

Features	Units	Description
flux_c_t	nT	Flux C: fluxgate total
cur_ac_lo	A	Current sensor: air conditioner fan low
ins_alt	m	INS computed elevation
flux_c_z	nT	Flux C: fluxgate z axis
flux_a_t	nT	Flux A: fluxgate total
vol_back_p	V	Voltage sensor: resolver board(+)
vol_back_n	V	Voltage sensor: resolver board(-)
ins_lat	rad	INS computed latitude
cur_com_1	A	Current sensor: aircraft radio 1
flux_c_y	nT	Flux C: fluxgate y axis
vol_acpwr	V	Voltage sensor: aircraft power
ins_wander	rad	INS computed wander angle
cur_flap	A	Current sensor: flap motor
vol_bat_2	A	Current sensor: battery 2
ins_roll	deg	INS computed aircraft roll

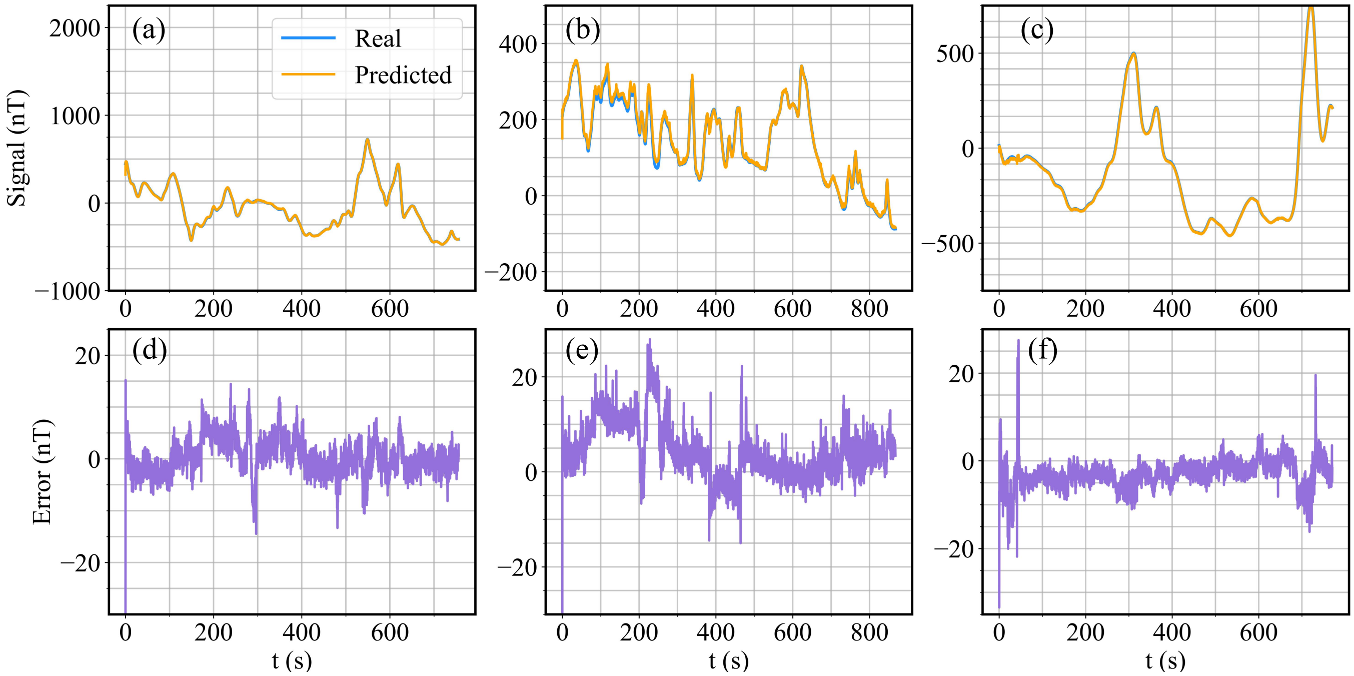

FIG. 6. Reservoir computing detected anomaly field signal for three additional lines of flight data. (a-c) The predicted signals together with the ground truth for the three lines (1003.02, 1003.08, and 1007.02). (d-f) The corresponding errors.

from two representative lines: line numbers 1004.4015 and 1006.07, are shown in Fig. 5. These two lines of data were from the Perth survey flown at 800 m within Eastern Ontario and the transit/descent from 3048 m to 400 m [7]. The reservoir computer predicted magnetic anomaly signals are shown in Figs. 5(a) and 5(b), together with the ground truth. The differences between the predicted and true signals are not apparent, and the overall $E_{\rm RMS}$ is about 6.0 nT. However, if we calculate the $E_{\rm RMS}$ without the transient, the overall $E_{\rm RMS}$ will be much lower and about 4.1 nT. An intrinsic difficulty with reservoir computing is that the initial nodal states of the trained machine are necessarily set to be zero and it takes a transient time for the output to approach the true signal. Figure 5(e) shows the distribution of the $E_{\rm RMS}$ from all 45 lines of fly data. If we neglect the large transient errors, the error values are small enough for the predicted field anomaly signals to be useful for navigation. Results from three additional flight data lines are shown in Fig. 6, where the line numbers are 1003.02, 1003.08, and 1007.02 that specify the magnetic-field measurements [7] of free-fly at 400m in Eastern Ontario, free-fly at 400 m on Renfrew, and free-fly at 800 m within the Perth minisurvey area, respectively. The three panels in the top row of Fig. 6 show the FNN predicted magnetic anomaly signals together with the ground truth, while the three panels in the bottom row show the corresponding errors. The ensemble averaged $E_{\rm RMS}$s for the three data sets are 3.6 nT, 7.2 nT, and 4.4 nT, respectively.

The accuracy of the existing magnetic anomaly maps is an important factor underlying the positioning accuracy. In some traditional anomaly maps, the accuracy is about 10 nT. With the aid of GPS, modern magnetic anomaly maps can have the accuracy of 1-3 nT [2]. The mean magnetic signal error below 6.5 nT typically corresponds to an error of less than 45 m in navigation positioning for each line [58]. While the relationship between the signal and the positioning error is generally nonlinear, we set the $E_{\rm RMS}$ threshold to 10 nT. $E_{\rm RMS}$ values greater than 10 nT are regarded as large errors.

The overall detection errors from reservoir computing are relatively small, even when transients are present at the beginning of the prediction. The advantages of reservoir computing include fast training and testing, intrinsic recurrence, and high prediction accuracy. The most time-consuming part of reservoir computing is Bayesian optimization, which is dependent upon the model complexity and the hidden-layer network size. The hyperparameter values from the Bayesian optimization ensures that the reservoir computer can “learn” the dynamics of this system and then output the target signal. The effects of the network size on the prediction performance are discussed in Sec. IV C. At the start of the prediction, the states of all nodes in the hidden-layer network are set to zero, leading to transients that can result in large detection errors. A brief analysis of the errors obtained from the two cases (with or without transient) is presented in Appendix B.

B. Results from time-delayed FNNs

Time delayed FNNs have the following advantages: the required small size of the underlying neural networks, incorporation of the historical information in the measurements directly into the training process, absence of transients, low computational cost, and high prediction accuracy. Typically, the machine learning architecture requires only a few hidden layers, each with a relatively small number of nodes. For the three magnetic signals and 15 feature signals (Table I, to be justified below) as the input data, we find that using two or three hidden layers suffices. To be concrete, we choose two settings: (1) two hidden layers, one with 30 and another with 10 nodes; (2) three hidden layers, with 50, 30, and 10 nodes in the first, second, and third layers, respectively. The three basic input signals from Mag_3, Mag_4, Mag_5 (Fig. 2) are preprocessed by the Tolles-Lawson model.

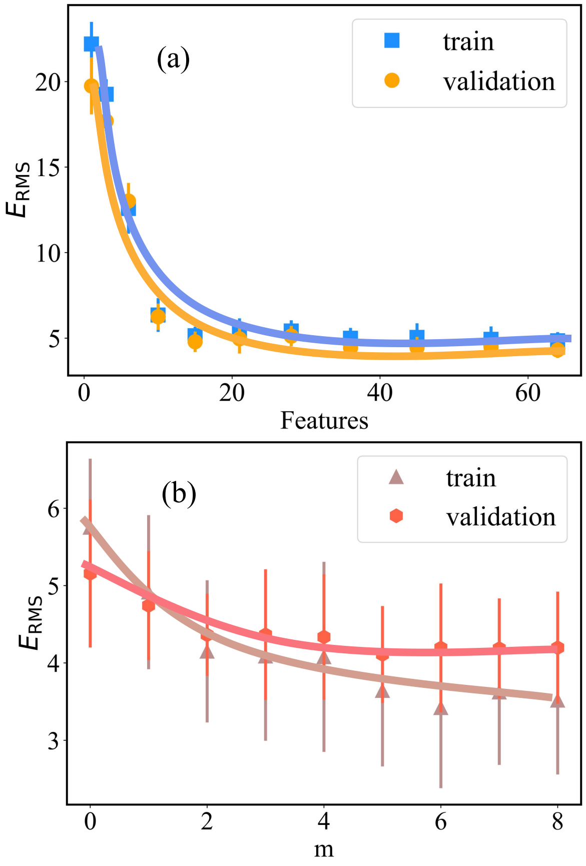

FIG. 7. Selecting the number of features and embedding dimension. (a) Ensemble averaged E_{RMS} (with 50 independent realizations) versus the number of features. The E_{RMS} decreases rapidly as the number increases from one but tends to increase only slightly as the number of more than about 20. (b) Ensemble averaged E_{RMS} versus the embedding time m. The E_{RMS} decreases with m but tends to a constant for $m > 5$. The E_{RMS}s are calculated for the training set and validation set, separately.

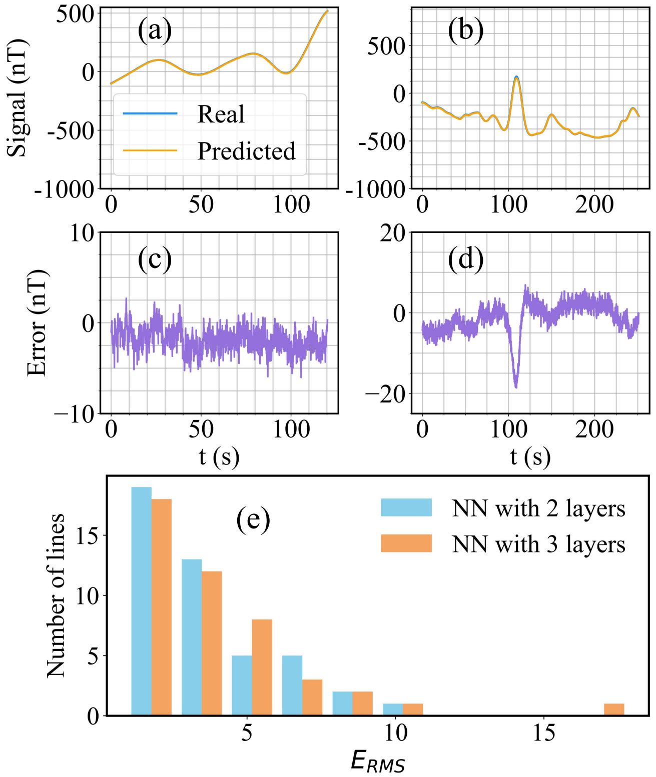

FIG. 8. FNN filtered anomaly field signal. (a,b) Comparing the real and predicted signals for lines 1004.4015, 1006.07. (c,d) The difference (error) between the predicted and true signals. (e) Distribution of the average E_{RMS} from 45 lines of flight data. On the whole, the prediction accuracies are adequate for magnetic anomaly guided navigation

As described in Sec. III, the three magnetic signals constitute the basic input signals. In addition, a large number of feature measurements are available. Utilizing all the additional features as inputs in general does not lead to the desired performances due to high computational cost associated with training. How to choose the proper features? We use the greedy algorithm for this task, which can provide a local best choice of the features [65] for complex optimization problems. (It should be noted that the algorithm has been widely used in optimizing the neural-network performance to reduce the E_{RMS} [66–68].) Specifically, for each iteration of the algorithm, we perform a testing loop by adding each feature to the input layer to measure the corresponding decrease in the resulting E_{RMS}, then remove this feature and test another candidate feature. After looping over all the remaining candidate features, we select several features that have the top performance on reducing the average E_{RMS} calculated from 50 independent runs. Once a feature has been selected, it will be an input signal in the following iterations. Figure 7(a) shows the continuous reduction in the ensemble averaged E_{RMS} as more features are included as the inputs. It can be seen that as the number of features increases from one to 15, the av-

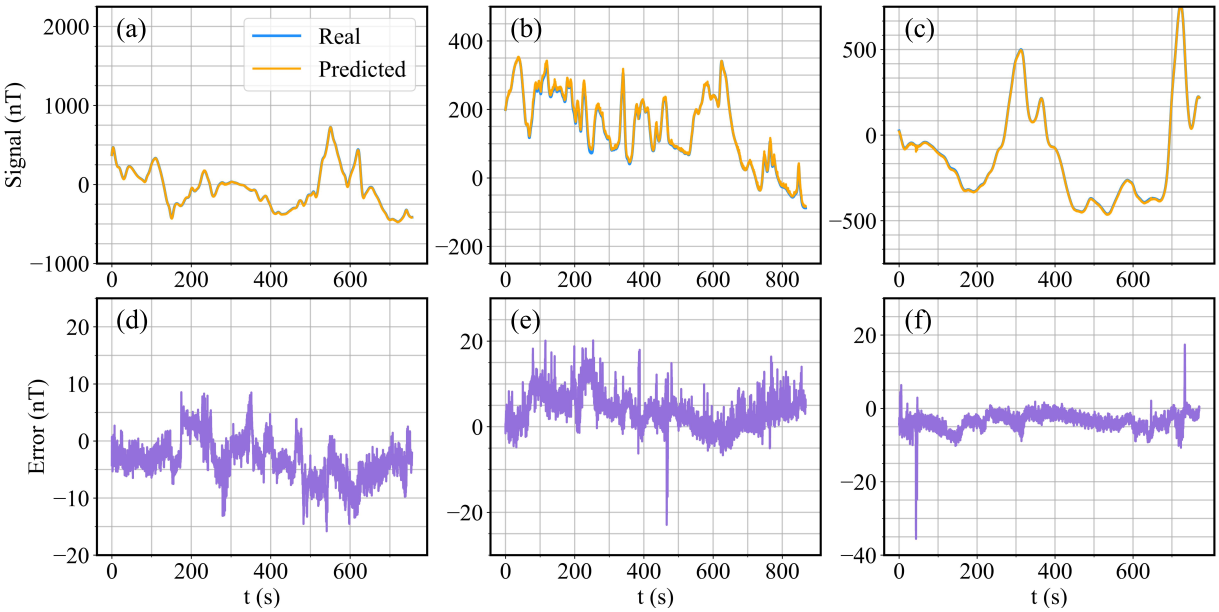

FIG. 9. Time delayed FNN filtered anomaly field signal for three additional lines of flight data. (a-c) The FNN predicted signals together with the ground truth the three lines (1003.02, 1003.08, and 1007.02). (d-f) The corresponding errors.

erage E_{RMS} decreases rapidly but tends to increase only slightly as more features are included. We thus choose 15 best features generated by a greedy algorithm as the additional input signals, as listed in Table I, which include the measurements from fluxgate, INS and current or voltage sensors. In all cases, there is no indication of any overfitting.

It is worth mentioning that the greedy algorithm usually does not produce optimal solutions. In fact, it might miss important features - each alone would not reduce the E_{RMS} but their combination would. We have used the greedy algorithm because the feature selection in our task is an NP-hard problem and the algorithm provides a viable solution. Indeed, if the global optimum is not reachable, a reasonable alternative is to use the greedy algorithm to find a local optimum to best select the most relevant features.

Two important parameters underlying our time-delayed FNNs are the amount of time delay τ associated with the input signals and the embedding dimension m. In traditional nonlinear or chaotic time series analysis based on the classic Takens' delay-coordinate embedding theory for reconstructing the phase space of the underlying nonlinear system [69–74], the delay time can be chosen empirically as the average oscillation period of the underlying time series (corresponding to unit value in a discrete-time map) [73, 75–77]. We apply the same criterion here to choose the proper value of τ. To determine the embedding dimension m is more complicated, where the standard Grassberger-Procaccia correlation integral algorithm [75] is often used for the task of reconstructing a chaotic attractor. Our task here to determine a proper configuration of the input signals is different from attractor reconstruction, so the Grassberger-Procaccia algorithm is not directly applicable. We resort to empirical testing. In particular, we select an adequate number of feature signals and calculate the ensemble averaged E_{RMS} versus the embedding dimension m. As m increases, we expect the average E_{RMS} to decrease and possibly approach a constant (relatively small) value when m exceeds a critical value. Figure 7(b) shows such a behavior when ten additional features are used (the total number of input signals being 13). It can be seen that for $m > 5$, the average training E_{RMS} levels off. We thus set $m = 5$. It is worth noting that the choice of the m value depends on the sampling rate of the data set. If the sampling rate is high in comparison with the intrinsic frequency of the data, choosing a small m value for reducing the computations may be beneficial. In our study, the sampling rate taken is not much higher than the intrinsic frequency of the magnetic-field data, so reducing the sampling rate may lead to large errors. For example, if we reduce the sampling rate two times, the testing error will increase approximately by over 20%. If the sampling rate is decreased three times, at some time steps the testing error would diverge.

We train and test 45 lines of available flight data following the same division scheme as reservoir computing. Because of the choice of 15 features and $m = 5$, the dimension of the input to the FNN is $d = (15{+}3){\times}6 = 108$. For consistency, we show the filtered results on the same lines as Sec. IV A shows. Figures 8(a) and 8(b) show, for the two hidden-layer ([30,10]) configuration, the FNN predicted signals for the data lines 1004.4015 and 1006.07. The corresponding errors are shown Figs. 8(c) and 8(d), which are comparable with those from reservoir computing [Fig. 5]. We have also tested an FNN configuration with three hidden layers ([50,30,10]) and obtained similar results. Figure 8(e) shows the distributions of the E_{RMS} from all 45 lines of the flight data, obtained with the two and three hidden-layer FNN configurations. The E_{RMS}s are in general quite small. Particularly, for the two hidden-layer neural network configuration, the ensemble averaged E_{RMS} and the standard deviation are 4.5 nT and 2.7 nT, respectively. For the three hidden-layer configuration, the corresponding values are 4.5 nT and 2.4 nT. Results from three additional flight data lines are shown in Fig. 9, where the line numbers are 1003.02, 1003.08, and 1007.02 that specify the magnetic-field measurements [7] of free-fly at 400m in Eastern Ontario, free-fly at 400 m on Renfrew, and free-fly at 800 m within the Perth mini-survey area, respectively. The three panels in the top row of Fig. 9 show the FNN predicted magnetic anomaly signals together with the ground truth, while the three panels in the bottom row show the corresponding errors. The ensemble averaged E_{RMS}s for the three data sets are 4.8 nT, 5.7 nT, and 4.3 nT, respectively. The small error values from time-delayed FNNs for all cases suggest the feasibility of achieving magnetic anomaly navigation aided by INS.

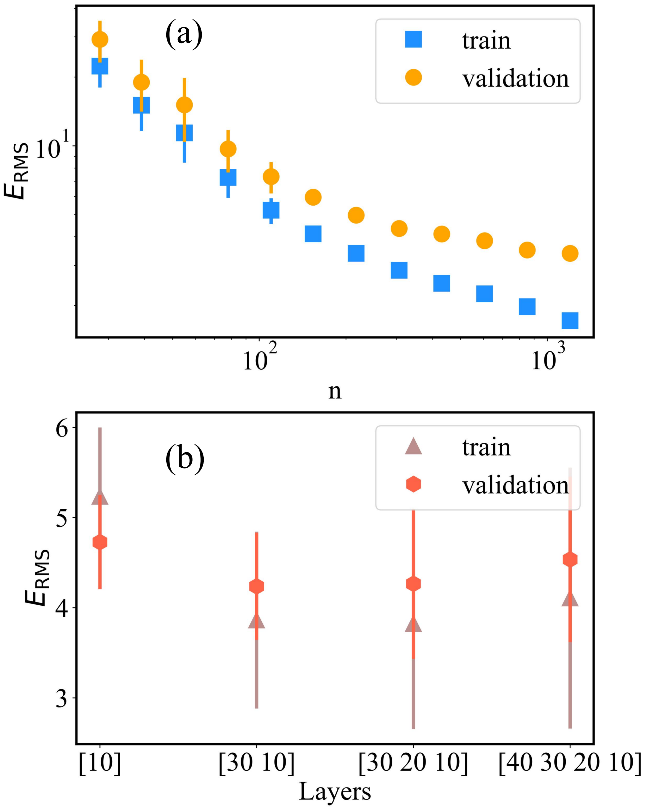

FIG. 10. Effects of altering the neural-network structure on the detection performance. (a) E_{RMS} for filtering anomaly field signal versus the network size n for reservoir computing. (b) E_{RMS} for four different configurations of time-delayed FNN. Each data point is the result of ensemble averaging 50 independent realizations.

C. Performance comparison between reservoir computing and time-delayed FNN

The structures of the two machine-learning methods are characteristically different, in the following aspects. First, reservoir computing has a recurrent structure that is intrinsically capable of preserving the memory in the data. The conventional FNN architecture does not have this "memory-preserving" property, but our time-delayed FNNs have. Second, a reservoir computer has a single hidden layer, rendering necessary using a relatively large network size (500 nodes in our work). An FNN typically contains multiple hidden layers, each with a relatively small size. Figure 10 shows the effects of the network structure for the two machine-learning architectures. In particular, Fig. 10(a) shows, for reservoir computing, both the training and validation errors decrease continuously as the hidden-layer network becomes larger. Figure 10(b) demonstrates that four configurations of the time-delayed FNN have comparable errors with the two-layer structure $(30, 10)$ performing slightly better than the others. Since the training time tends to increase significantly with the number of hidden layers, we choose the two-layer structure $(30, 10)$. In all cases studied, there is no apparent sign of overfitting [78].

The parameter optimization algorithms for the two machine-learning methods are also different. As explained in Appendix A 1, after updating the network state, linear regression is performed. As a result, the training time depends only on the training data length. For FNNs, as described in Appendix A 2, the parameters are optimized through a gradient-descent based algorithm, so the training time depends on the training data length and epoch. We set a small epoch number for feature and embedding time selection (See Fig. 7) for reducing computation time, and a longer epoch number for testing results. Furthermore, a unique feature of reservoir computing, because of its being a closed-loop, self-evolving dynamical system in the test phase, is the occurrence of transients that can lead to inaccuracies, especially at the beginning of the filtering process. Our results indicate that, despite the different properties and structures of the two machine-learning schemes, both have the ability to output the correct anomaly filed signal from several measurement signals immersed in overwhelming noises.

Finally, we discuss the issue of computational cost for the two machine-learning architectures. For the time-delayed FNN, the average computer time required for training with the optimal hyperparameters is 650s, and the one-step inference computation time is 0.003s. For reservoir computing, a substantial amount of the computation time is required for finding the optimal hyperparameter values: about four hours. However, one can use pre-recorded data for this task. Once the optimal hyperparameter values have been found, they are fixed. The mean training time is 23s and the one-step inference computation time is 4×10^{-5}s. For navigation to be practically feasible, the time interval of prediction should be, e.g., less than 1s for an airplane and 5ms for a hypersonic vehicle. Our estimates of the computational time indicate that both neural-network architectures can meet this requirement.

V. DISCUSSION

The focus of our study is the detection of physical signals, e.g., those generated in natural systems, whose measurements come with strong noises. The signals of interest are thus weak in comparison with the various noises that occur inevitably during the measurements. The physical signals themselves are generated by some nonlinear dynamical systems whose effective intrinsic dimension can be extremely high. Detecting such weak signals from an overwhelmingly strong noise background constitutes one of the most challenging problems in nonlinear signal processing. The achievement of the present work is the articulation and demonstration of a machine-learning framework to address this challenging problem.

The application investigated in this paper is precise positioning and navigation of flying objects guided by the earth's magnetic-anomaly field. The hypothetical scenario is that GPS is not available so navigation must rely on some alternative signals, and the anomaly field provides the most natural physical signal. Measurements of the anomaly field typically occur in the cockpit of an aircraft. There are two types of noise sources. The first is associated with the physical environment of the earth: the measured earth magnetic field contains other components such as the fields generated by the core of the earth, its atmospheric environment, and possibly magnetic sources in the solar system. The second is the magnetic interference from the sophisticated electronic equipment within the airplane cockpit, which can be orders of magnitude stronger than the earth anomaly field. To obtain the anomaly field for navigation, these noises must be filtered out from the measurements.

Because of the physical and dynamical nature of the weak signals to be detected, a machine-learning framework must necessarily possess a memory capacity. Recurrent neural networks such as reservoir computing have intrinsic memories (Appendix A 1) with the advantage of fast training and prediction. With Bayesian optimization for hyperparameters, a reservoir computer can generate accurate predictions, especially when transients are removed. Time-delayed FNNs can also be effective for weak signal detection, with the advantage that neural networks of small sizes can be used.

The real-world application studied in this paper is based on noisy flight data collected within the aircraft cockpit and the ground truth, i.e., the actual magnetic anomaly field was recorded by the tail stinger located outside the aircraft. The data sets thus provide an ideal platform to test our machine learning framework. From the noisy cockpit data, we first use the Tolles-Lawson model to obtain a number of preprocessed magnetic signals: these from Mag_3, Mag_4 and Mag_5, as shown in Fig. 2(a). We then use our reservoir computing and time-delayed FNNs to calibrate the signals and predict the magnetic anomaly field. More specifically, to achieve accuracies as high as possible, we input the three preprocessed signals together with 15 additional feature signals selected by a greedy algorithm, which include the magnetic field signals, fluxgate, and noise recorded by different sensors inside the aircraft. We justify the rationals behind the selection of the additional feature signals through assessing the change in the ensemble averaged E_{RMS} as the available feature signals are included in the input signal to the neural network one after another. Our results indicate that the machine learning methods with the combination of the input signals so selected can lead to accurate and stable predictions of the magnetic anomaly field. With the aid of INS, the anomaly field can be used for precise positioning so as to navigate the aircraft. For example, extended Kalman filters [79] can be exploited to develop the navigation algorithm, where the aircraft can estimate its location according to the data recorded by the sensors inside the aircraft, the predetermined magnetic maps, and INS. The mean magnetic signal errors in our work are around 4 nT, where an error below 6.5 nT corresponds to the positioning error of less than 45 m. Empirically, the position error is approximately about $10 - 40$ m when the magnetic signal error is around 4 nT. The anomaly field detected by our methods can thus be used for actual aircraft navigation positioning.

We remark on the necessity of combining the Tolles-Lawson model with machine learning. While the Tolles-Lawson model still represents a state-of-the-art approach to calibrating the earth anomaly magnetic field, its sole use will result in a large distance root-mean-square error. A basic reason is that the Tolles-Lawson model is a linear model, while the underlying dynamical process generating the time-varying data of the anomaly field is nonlinear. To reduce the error, it is then necessary to combine the Tolles-Lawson model with some additional nonlinear signal processing method. Exploiting machine learning represents a viable approach. Nonetheless, the currently available machine-learning architectures for complex signal filtering is not powerful enough in the sense that, if it is used alone without the Tolles-Lawson correction, the resulting errors will be large. In our work, the Tolles-Lawson model is used for data preprocessing, aiding the machine learning methods to better filter the magnetic signal. This strategy aligns well with the recent trend of developing "physics-informed neural networks," where specific physical principles are incorporated into machine learning to improve performance, robustness and efficiency. (For example, in mechanics, the Hamiltonian structure can be built into the neural-network architecture for it to better learn the dynamics [80–84]. Physics-based neural networks have also been developed in areas such as fluid dynamics [85], optical meta surface design [86] and quantum tomography [87].)

A number of issues remain. First, while the achieved average $E_{\rm RMS}$s are small and may be deemed as qualified for actual navigation, there are random fluctuations associated with the predicted magnetic anomaly field. It is desired that these fluctuations be significantly reduced to achieve higher prediction accuracy, which requires their physical and dynamical origin to be known. Second, both machine-learning schemes have shortages. Reservoir computing has transient and, for time-delayed FNNs, it is necessary to determine the key hyperparameter - the embedding dimension of the time-delayed signals. We have used some empirical observation to determine the embedding dimension, but a more justified criterion based on mathematical and physical considerations is desired. Third, the sizes of the FNN neural networks in the hidden layers used in our study are relatively small, which are determined also empirically through numerical tests. A mathematical understanding of how the network sizes affect the prediction performance is essential to extending the machine learning methods to detecting weak signals arising from other domains of research.

ACKNOWLEDGMENT

This work was supported by AFOSR under Grant No. FA9550-21-1-0438.

Appendix A: Machine learning methods

1. Reservoir computing

A reservoir computer consists of three layers: an input layer, a hidden recurrent layer, and an output layer. The major advantage of reservoir computing is that the input weights and hidden layer weights are preselected and only the weights of the output layer $W_{\rm out}$ need to be determined by training through a linear regression associated with the updating of the dynamical state of the hidden-layer neural network.

The iterative equation of the dynamical process in the hidden layer is described by

$$\begin{aligned} r(t) &= (1-\alpha) r(t-\Delta t) \\ &\quad + \alpha \tanh(\mathcal{A} r(t-\Delta t) + W_{\rm in} u(t)), \qquad \text{(A1)} \\ v(t) &= W_{\rm out} r(t), \qquad \text{(A2)} \end{aligned}$$

where α is the leakage parameter. During the training, the input data are injected as $u(t)$ to activate the dynamical state $r(t)$ of the hidden layer neural network. All states $r(t)$ stimulated this way are concatenated in terms of the temporal dimension to form a matrix R of dimension $n \times T_{\rm train}$, where $T_{\rm train}$ is the training length. Similarly, the time series of the training target $B_{\rm SLG}$ is also concatenated to form a matrix V. A ridge regression is finally performed between $R' = f(R)$ and the V to yield the output matrix:

$$W_{\rm out} = V \cdot R'^T (R' \cdot R'^T + \beta I)^{-1}, \qquad \text{(A3)}$$

where β is the coefficient of the l-2 regularization.

For reservoir computing, hyperparameter optimization is essential to achieving the desired performance. We use a Bayesian Optimization algorithm from Matlab (*surrogateopt*) to find the optimal values of the following hyperparameters: the leakage α, the regularization coefficient β, the scaling factor b of the input matrix $W_{\rm in}$, the spectral radius ρ of the recurrent network $\mathcal{A}$, and the nodal connection probability p of the network $\mathcal{A}$.

2. Feed-forward neural networks (FNNs)

FNNs are artificial neural networks that do not possess any loops in the directed connections among the neurons [88]. Different from the recurrent neural networks (RNNs) that have a feedback structure, in an FNN there is only one direction of information flow, i.e., the neurons can only process the signal forward through the connections from the inputs to the outputs. We use FNNs with a classic layered architecture that has an input layer, a number of hidden layers, and an output layer. Each layer contains a number of neurons, and there are only inter-layer connections from the neurons in a former layer closer to the inputs to a latter layer closer to the outputs, which are directed and weighted. A sufficiently large FNN is able to perform as a universal approximator, which can capture the hidden complex relationship between the input and output data, accomplished through a training process that adjusts the weights to map the inputs to the outputs.

We use an improved version of FNNs: time-delayed FNNs, as in Fig. 4, which is essentially the architecture of "next-generation reservoir computing" [29, 30, 34], where the present and historical information of the time series data is inputted into the neural network through time-delayed embedding. In particular, we define the following weight vector of all the connections and neurons:

$$w = (\cdots, w_{i,j}^l, w_{i+1,j}^l, \cdots, w_{N_l,j}^l, \theta_j^{l+1}, w_{i,j+1}^l, \cdots)^T, \qquad \text{(A4)}$$

where w_{ij}^l is the weight between neuron i in layer l to neuron j in the next layer $l+1$, N_l is the number of neurons in layer l, and θ_j^{l+1} is the bias of neuron j in layer l. In the training phase, a global error function $E(w)$ is minimized, which depends on the weight vector in Eq. (A4), where $E(w)$ can be a least-squares function or other reasonable error functions. While a variety of error function optimization strategies with different features are available, they share the same optimization principle: searching for the neighbors around the current point in the weight space and minimizing the error function to gradually reach the global minimum. More specifically, to minimize the error function $E(w_k)$ at the kth step, we seek the search direction p_k and step size σ_k such that $E(w_k + \sigma_k p_k) < E(w_k)$ and then update w_{k+1} to $w_k + \sigma_k p_k$. In general, the search direction and step size are important because they represent the direction of search in weight space from the present point and the distance to proceed in that direction, respectively. The iterating process stops when the gradient of the error function $E'(w_k)$ becomes approximately zero.

When the search direction p_k and the step size σ_k are set as the negative gradient and as a constant, respectively, the algorithm is essentially a classical back-propagation (BP), gradient-descent based algorithm. However, a constant step size tends to make the algorithm non-adaptive and inefficient and the most negative of the gradient can lead to poor convergence. Let y be the change added to the weight vector, then the linear approximation $E(w+y) \approx E(w) + E'(w)y$ constitutes the base of minimization in BP. In fact, the BP algorithm relies heavily on parameters such as the learning rate and the momentum constant [63, 89], which can lead to difficulties large-scale problems. In this work, we choose the search direction and the step size by using the information from the second-order approximation

$$E(w+y) \approx E(w) + E'(w)^T y + y^T E''(w) y/2.$$

Further, a conjugate gradient (CG) algorithm [63] can yield faster convergence while keeping the error minimized so it can handle complex problems in a more effective way. In CG, the search direction goes along the conjugate direction and the step size is adjusted at each iteration, which is a variant of the gradient descent with an additional term from the last search step, defined as

$$p_{k+1} = -g_{k+1} + \beta_k p_k, \tag{A5}$$

where g_{k+1} is again the gradient vector $g_{k+1} = -E'(w_{k+1})$, p_k is the conjugate direction from the last search step, and

$$\beta_k = (|g_{k+1}|^2 - g_{k+1}^T g_k)/p_k^T g_k$$

is the weight of the previous direction p_k. The step size is scaled by a second-order term: $s_k = E''(w_k)p_k$. More details of the algorithm can be found in [63].

In our work, we use scaled conjugate gradient (SCG) as the optimization algorithm. Similar to CG, SCG also uses the conjugate direction calculated by Eq. (A5). Compared with CG, the main advantage of SCG is that it uses another more efficient method to estimate the step size at each step. In particular, the second-order information term $s_k = E''(w_k)p_k$ can be replaced by:

$$s_k \approx \frac{E'(w_k + \sigma_k p_k) - E'(w_k)}{\sigma_k} + \lambda_k p_k \tag{A6}$$

where λ_k is a scalar to regulate the indefiniteness of $E''(w_k)$ [90]. Overall, through adjusting the search direction and step size in a reasonable and efficient way, SCG is efficient and adaptive, and it does not depend on any individual dependent parameters.

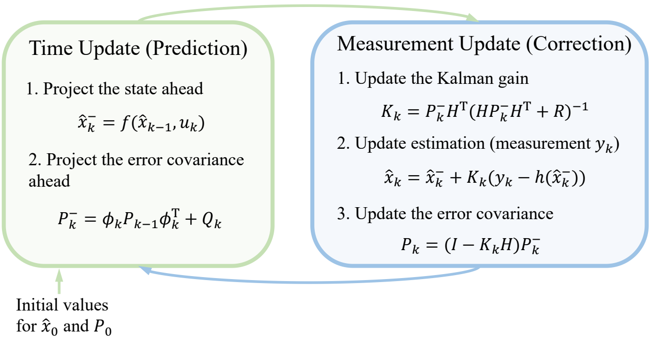

FIG. 11. An extended Kalman filter. It contains two steps: prediction and correction. See text for legends.

3. Extended Kalman filters

The two machine-learning architectures studied in this paper, i.e., reservoir computing and time-delayed FNNs, are generally suited for time series prediction and classification applications. In contrast, Kalman filtering is for state estimation of the system from noisy and uncertain measurements. In particular, the classical Kalman filtering algorithm is suitable for time invariant linear dynamical systems while extended Kalman filtering is applicable to nonlinear dynamical systems. In both cases, the Kalman filter predicts the state of the system at the next time step from a mathematical model of the system that is known *a priori*, and then refines the forecast using the current measurement of the state variables. If the known system model is accurate, a high accuracy in the estimated state by the Kalman filter can be achieved.

What does it entail if one intends to apply extended Kalman filtering to our problem of anomaly magnetic field detection? In classical Kalman filtering, the initial state estimation and the initial covariance matrix are required as initial conditions. Also needed are the predicted noise covariance and the measurement noise covariance matrix representing the inaccuracies during the state-estimation process. The dynamical system underlying the anomaly magnetic field is nonlinear, requiring extended Kalman filtering whose general structure is shown in Fig. 11, where f is the nonlinear state transition model, h represents the nonlinear measurement function, and their respective Jacobian matrices are Φ and H. Because the magnetic field is passive, there is no system input u. The Tolles-Lawson model provides the measurement y_k at each time step. For the extended Kalman filtering to be applicable to our problem, the nonlinear models f and h to map the state forward in time are needed, but these are unknown. For Kalman filtering, precise knowledge of the mathematical model governing the dynamical evolution of the system of interest is an essential requirement.

The reliance of Kalman filtering on the accurate mathematical model of the underlying dynamical system presents a fundamental difficulty in applications where such a model is not available. Our problem of estimating the earth anomaly magnetic field from noisy measurements belongs to this category because the underlying dynamical model governing the magnetic field is not available. Another difficulty with applying Kalman filtering to our problem lies in the nature of the noises. In Kalman filtering, classic or extended, the noises in the system are assumed to be Gaussian, whose distributions are symmetric with a constant variance. However, in our anomaly magnetic field detection problem, there are a variety of noise sources whose distributions deviate from Gaussian, which include the other components of the earth magnetic field and the field generated by the extensive electronics in the cockpit of the airplane. The deficiencies of Kalman filtering for nonlinear state estimation have long been recognized [91].

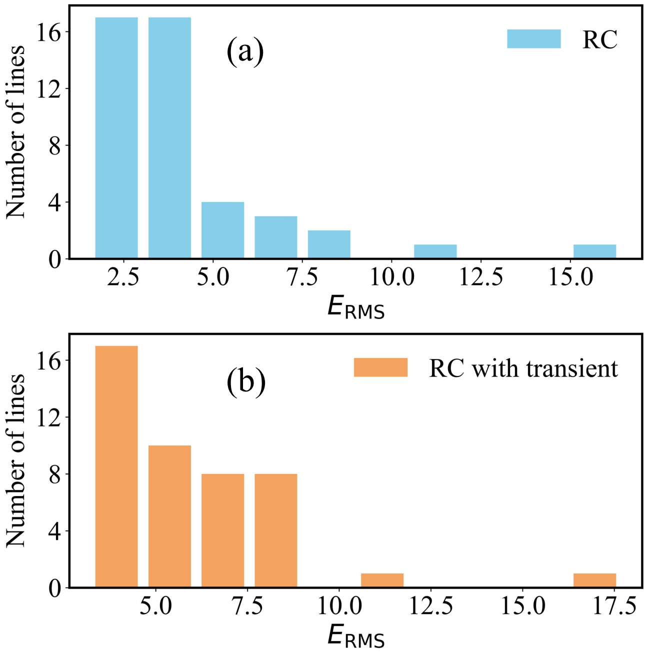

FIG. 12. Effects of transients on reservoir-computing performance. (a) Small E_{RMS} with the transients removed. (b) Relatively large E_{RMS} without removing the transient.

Appendix B: Effects of transients and low-pass filtering

In Sec. IV, the issue of transients in reservoir computing is mentioned. How do transients affect the detection performance? Figures 12(a) and 12(b) show the global E_{RMS} distributions for the two cases where the transient is removed and retained, respectively. The results suggest the benefits of removing the transients in reducing the error.

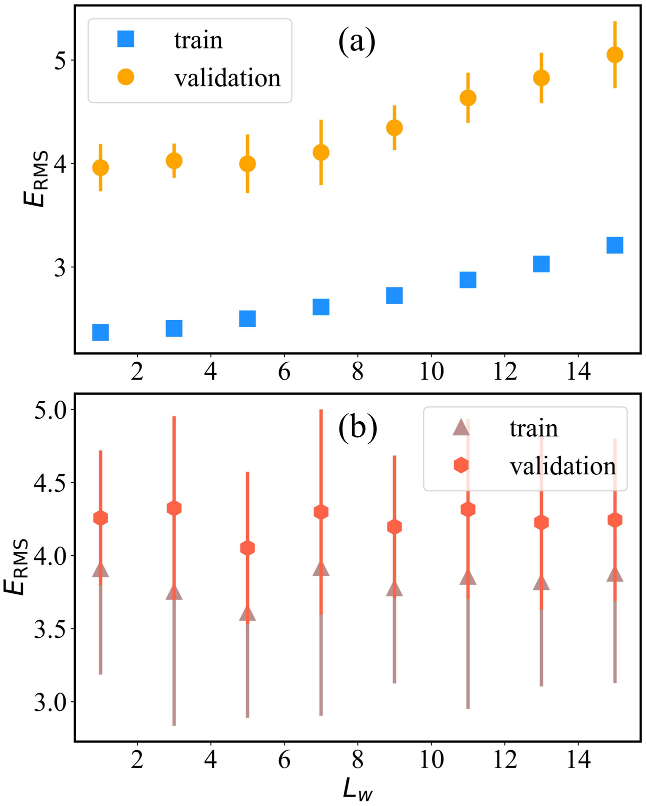

FIG. 13. Effect of low-pass filtering for data preprocessing on detection performance. Shown is E_{RMS} versus the Gaussian filter window size. (a) Training and validation E_{RMS} for reservoir computing, which tends to increase with the window size. (b) Training and validation E_{RMS} for time-delayed FNNs. Varying the window size has little effect on the errors. In both panels, each point is the result of ensemble average of 50 independent realizations.

In addition, we study if low-pass filtering the raw data can help improve the machine-learning performance in detecting the anomaly magnetic field. In particular, we employ a half Gaussian-window filter of window size L_w to filter out the high-frequency fluctuations in the original data, which operates on the historical half of the data within the window. The results are shown in Fig. 13. For both reservoir computing and time-delayed FNN, applying the low-pass filtering does not lead to any performance improvement. In fact, as the window width L_w increases, the errors tend to increase. A plausible explanation is that the machine-learning architecture intrinsically has some filtering capability. More importantly, the high-frequency fluctuations in the data that low-pass filtering aims to remove may constitute historical information (memory) and are often beneficial to training a

TABLE II. Line number summary for flight 1003

Line number	Description	Training length (s)	Validation/Test length (s)
1003.02	Eastern Ontario Free-Fly 400m	2246.5	748.8
1003.03	Climb to 800m	61.3	20.4
1003.04	Eastern Ontario Free-Fly 800m	2877.7	959.2
1003.05	Transit at 800m	246.7	82.2
1003.06	Descend to 400m	83.5	27.8
1003.07	Transit to Renfrew Free-Fly	58.9	19.6
1003.08	Renfrew Free-Fly 400m	2581.9	860.6
1003.09	Climb to 800m	82.3	27.4

TABLE III. Line number summary for flight 1004

Line number	Description	Training length (s)	Validation/Test length (s)
4019.00	Perth Survey Line	318.7	106.2
4018.00	Perth Survey Line	328.7	109.6
4017.00	Perth Survey Line	327.1	109.0
4016.00	Perth Survey Line	331.5	110.5
421.00	Perth Survey Line	77.6	25.9
419.00	Perth Survey Line	90.2	30.1
417.00	Perth Survey Line	82.7	27.6
415.00	Perth Survey Line	97.4	32.5
413.00	Perth Survey Line	77.2	25.7
411.00	Perth Survey Line	89.4	29.8
409.00	Perth Survey Line	74.2	24.7
408.00	Perth Survey Line	91.0	30.3
407.00	Perth Survey Line	74.4	24.8
405.00	Perth Survey Line	102.7	34.2
403.00	Perth Survey Line	76.9	25.6
401.00	Perth Survey Line	92.6	30.9
4015.00	Perth Survey Line	339.7	113.2
4012.00	Perth Survey Line	334.3	111.4
4001.00	Perth Survey Line	339.3	113.1

neural network, so removing them from the original data may deteriorate the machine-learning performance.

Appendix C: Data description

We describe in more detail the 45 lines of flight data used in this study. Five flights were flown to collect data, named 1003, 1004, 1005, 1006, 1007. Each individual flight contains several flight segments, e.g., 1003.02, 1003.03, etc. The details of the flights and their related description are presented in Tabs. II, III, IV, V, and VI.

[1] R. T. Ioannides, T. Pany, and G. Gibbons, Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques, Proc. IEEE **104**, 1174 (2016).

[2] A. Canciani and J. Raquet, Absolute positioning using the earth's magnetic anomaly field, Navigation **63**, 111 (2016).

[3] A. Canciani and J. Raquet, Airborne magnetic anomaly navigation, IEEE Trans. Aero. Elec. Sys. **53**, 67 (2017).

[4] W. E. Tolles, *Compensation of Induced Magnetic Fields in MAD Equipped Aircraft* (Airborne Instruments Lab, New York, 1943).

[5] W. E. Tolles and J. D. Lawson, *Magnetic Compensation of MAD Equipped Aircraft* (Airborne Instruments Lab, New York, 1950).

[6] Q. Han, Z. Dou, X. Tong, X. Peng, and H. Guo, A modified TollesLawson model robust to the errors of the three-axis strapdown magnetometer, IEEE Geosci. Remote Sen. Lett. **14**, 334 (2017).

[7] A. R. Gnadt, J. Belarge, A. Canciani, L. conger, J. Curro, A. Edelman, P. Morales, M. F. O'Keeffe, J. Taylor, and C. Rackauckas, Signal enhancement for magnetic navigation challenge problem, arXiv preprint arXiv:2007.12158 (2020).

TABLE IV. Line number summary for flight 1005

Line number	Description	Training length (s)	Validation/Test length (s)
4011.00	Perth Survey Line	325.0	108.3
4010.00	Perth Survey Line	366.8	122.3
4009.00	Perth Survey Line	325.2	108.4
4008.00	Perth Survey Line	334.9	111.6
4007.00	Perth Survey Line	328.5	109.5
4004.00	Perth Survey Line	331.3	110.4
4003.00	Perth Survey Line	323.5	107.8
4002.00	Perth Survey Line	335.7	111.9

TABLE V. Line number summary for flight 1006

Line number	Description	Training length (s)	Validation/Test length (s)
1006.03	Climb to 17,000ft	448.3	149.4
1006.04	Compensation maneuvers at 17,000ft	2547.7	849.2
1006.05	Descent to 10,000ft	317.5	105.8
1006.06	Compensation maneuvers at 10,000ft	369.1	123.0
1006.07	Transit/Descent to Eastern Ontario	732.1	244.0
1006.08	Compensation maneuvers in Eastern Ontario at 400m	479.5	159.8

[8] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley Publishing Company, Redwood City, California, 1991).

[9] H. Jaeger, The echo state approach to analysing and training recurrent neural networks-with an erratum note, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report **148**, 13 (2001).

[10] W. Maass, T. Natschläger, and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neu. Comput. **14**, 2531 (2002).

[11] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).

[12] N. D. Haynes, M. C. Soriano, D. P. Rosin, I. Fischer, and D. J. Gauthier, Reservoir computing with a single time-delay autonomous Boolean node, Phys. Rev. E **91**, 020801 (2015).

[13] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification, Phys. Rev. X **7**, 011015 (2017).

[14] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Phys. Rev. Lett. **120**, 024102 (2018).

[15] T. L. Carroll, Using reservoir computers to distinguish chaotic signals, Phys. Rev. E **98**, 052209 (2018).

[16] K. Nakai and Y. Saiki, Machine-learning inference of fluid variables from data using reservoir computing, Phys. Rev. E **98**, 023111 (2018).

[17] Z. S. Roland and U. Parlitz, Observing spatio-temporal dynamics of excitable media using reservoir computing, Chaos **28**, 043118 (2018).

[18] A. Griffith, A. Pomerance, and D. J. Gauthier, Forecasting chaotic systems with very low connectivity reservoir computers, Chaos **29**, 123108 (2019).

[19] J. Jiang and Y.-C. Lai, Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius, Phys. Rev. Research **1**, 033056 (2019).

[20] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Recent advances in physical reservoir computing: A review, Neu. Net. **115**, 100 (2019).

[21] H. Fan, J. Jiang, C. Zhang, X. Wang, and Y.-C. Lai, Long-term prediction of chaotic systems with machine learning, Phys. Rev. Research **2**, 012080 (2020).

[22] C. Zhang, J. Jiang, S.-X. Qu, and Y.-C. Lai, Predicting phase and sensing phase coherence in chaotic systems with machine learning, Chaos **30**, 083114 (2020).

[23] C. Klos, Y. F. K. Kossio, S. Goedeke, A. Gilra, and R.-M. Memmesheimer, Dynamical learning of dynamics, Phys. Rev. Lett. **125**, 088103 (2020).

[24] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Machine learning prediction of critical transition and system collapse, Phys. Rev. Research **3**, 013090 (2021).

[25] D. Patel, D. Canaday, M. Girvan, A. Pomerance, and E. Ott, Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity, Chaos **31**, 033149 (2021).

[26] J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, Teaching recurrent neural networks to infer global temporal structure from local examples, Nat. Machine Intell. **3**, 316 (2021).

[27] H. Fan, L.-W. Kong, Y.-C. Lai, and X. Wang, Anticipating synchronization with machine learning, Phys. Rev. Resesearch **3**, 023237 (2021).

[28] L.-W. Kong, H. Fan, C. Grebogi, and Y.-C. Lai, Emergence of transient chaos and intermittency in machine learning, J. Phys. Complexity **2**, 035014 (2021).

[29] E. M. Bollt, On explaining the surprising success of reservoir computing forecaster of chaos? The universal ma-

TABLE VI. Line number summary for flight 1007

Line number	Description	Training length (s)	Validation/Test length (s)
1007.02	Free-Fly at 800m within Perth Mini-Survey Area	2289.7	763.2
1007.03	Descent/Transit to 400m Eastern Ontario Free-Fly	78.1	26.0
1007.05	Transit to Renfrew Free-Fly	240.1	80.0
1007.06	Free-Fly at 400m within Renfrew Area	3120.1	1040.0

chine learning dynamical system with contrast to VAR and DMD, Chaos **31**, 013108 (2021).

[30] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, Next generation reservoir computing, Nat. Commun. **12**, 1 (2021).

[31] T. L. Carroll, Optimizing reservoir computers for signal classification, Front. Physiol. **12**, 685121 (2021).

[32] T. L. Carroll, Optimizing memory in reservoir computers, Chaos **32**, 023123 (2022).

[33] S. Krishnagopal, M. Girvan, E. Ott, and B. Hunt, Separation of chaotic signals by reservoir computing, Chaos **30**, 023123 (2019).

[34] W. A. Barbosa and D. J. Gauthier, Learning spatiotemporal chaos using next-generation reservoir computing, Chaos: An Interdisciplinary Journal of Nonlinear Science **32**, 093137 (2022).

[35] G. A. Glatzmaiers and P. H. Roberts, A three-dimensional self-consistent computer simulation of a geomagnetic field reversal, Nature **377**, 203 (1995).

[36] A. R. Gnadt, Advanced aeromagnetic compensation models for airborne magnetic anomaly navigation (2022).

[37] H. Hu and N. Wei, A study of GPS jamming and anti-jamming, in *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, Vol. 1 (2009) pp. 388–391.

[38] C. Huang, G. Hendeby, and I. Skog, A tightly-integrated magnetic-field aided inertial navigation system, preprint (2022).

[39] T. N. Lee and A. J. Canciani, Magslam: Aerial simultaneous localization and mapping using earth's magnetic anomaly field, Navigation **67**, 95 (2020), https://onlinelibrary.wiley.com/doi/pdf/10.1002/navi.352.

[40] H. F. Rice, V. Benischek, and L. Sczaniecki, Application of atom interferometric technology for GPS independent navigation and time solutions, in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)* (2018) pp. 1097–1106.

[41] M. Mandea and M. Korte, *Geomagnetic Observations and Models* (Springer, 2011).

[42] I. Skog, G. Hendeby, and F. Gustafsson, Magnetic odometry-a model-based approach using a sensor array, in *2018 21st International Conference on Information Fusion (FUSION)* (IEEE, 2018) pp. 794–798.

[43] I. Skog, G. Hendeby, and F. Trulsson, Magnetic-field based odometry–an optical flow inspired approach, in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (IEEE, 2021) pp. 1–8.

[44] E. Dorveaux, T. Boudot, M. Hillion, and N. Petit, Combining inertial measurements and distributed magnetometry for motion estimation, in *Proceedings of the 2011 American Control Conference* (IEEE, 2011) pp. 4249–4256.

[45] A. Canciani and J. Raquet, Airborne magnetic anomaly navigation, IEEE Trans. Aero. Electro. Sys. **53**, 67 (2017).

[46] W. Erdmann, H. Kmita, J. Kosicki, and L. Kaczmarek, How the geomagnetic field influences life on earth an integrated approach to geomagnetobiology, Orig. Life Evol. Biosph. **51**, 1 (2021).

[47] J. T.-H. Lo, Neural Filtering, Scholarpedia **4**, 7868 (2009), revision #91563.

[48] J. Lo and L. Yu, Recursive neural filters and dynamical range transformers, Proc. IEEE **92**, 514 (2004).

[49] B. Lim, S. Zohren, and S. Roberts, Recurrent neural filters: Learning independent Bayesian filtering steps for time series prediction, in *2020 International Joint Conference on Neural Networks (IJCNN)* (2020) pp. 1–8.

[50] J. Larsen, *Design of Neural Network Filters* (Citeseer, 1996).

[51] F. L. A. and P. D. V., Recurrent neural networks for state estimation, in *Proceedings of the 12th Yale Workshop on Adaptive and Learning Systems* (2003).

[52] M. Zmitri, H. Fourati, and C. Prieur, Bilstm network-based extended Kalman filter for magnetic field gradient aided indoor navigation, IEEE Sen. J. **22**, 4781 (2021).

[53] F. Viset, R. Helmons, and M. Kok, An extended Kalman filter for magnetic field SLAM using Gaussian process regression, Sensors **22**, 2833 (2022).

[54] G. Shorshi and I. Y. Bar-Itzhack, Satellite autonomous navigation based on magnetic field measurements, J. Guid. Cont. Dyn. **18**, 843 (1995).

[55] M. Zmitri, H. Fourati, and C. Prieur, Improving inertial velocity estimation through magnetic field gradient-based extended kalman filter, in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (IEEE, 2019) pp. 1–7.

[56] C.-I. Chesneau, M. Hillion, and C. Prieur, Motion estimation of a rigid body with an ekf using magneto-inertial measurements, in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (2016) pp. 1–6.

[57] S. Ganguly, Y. Gu, Y. Xie, M. R. Stan, A. W. Ghosh, and N. K. Dhar, Reservoir computing based neural image filters, in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society* (2018) pp. 3206–3210.

[58] A. Gnadt, Machine learning-enhanced magnetic calibration for airborne magnetic anomaly navigation, in *AIAA SCITECH 2022 Forum* (2022) p. 1760.

[59] Y. Li and M. Nabighian, Tools and techniques: Magnetic methods of exploration–principles and algorithms, Treat. Geophys. **11**, 335 (2015).

[60] The github repository of this Signal Enhancement for Magnetic Navigation Challenge Problem is at `github.com/MIT-AI-Accelerator/MagNav.jl`.

[61] W. E. Tolles, Magnetic field compensation system, U.S. Patent (1955).

[62] S. H. Bickel, Small signal compensation of magnetic fields resulting from aircraft maneuvers, IEEE Trans. Aero. Elect. Sys. **AES-15**, 518 (1979).

[63] M. F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neu. Net. **6**, 525 (1993).

[64] A. Jain, K. Nandakumar, and A. Ross, Score normalization in multimodal biometric systems, Pat. Recog. **38**, 2270 (2005).

[65] J. Edmonds, Matroids and the greedy algorithm, Math. Program. **1**, 127 (1971).

[66] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains* (John Wiley & Sons, 2013).

[67] M. Burger and A. Hofinger, Regularized greedy algorithms for network training with data noise, Computing **74**, 1 (2005).

[68] E. Lybrand and R. Saab, A greedy algorithm for quantizing neural networks., J. Mach. Learn. Res. **22**, 156 (2021).

[69] F. Takens, Detecting strange attractors in fluid turbulence, in *Dynamical Systems and Turbulence*, Lecture Notes in Mathematics, Vol. 898, edited by D. Rand and L. S. Young (Springer-Verlag, Berlin, 1981) pp. 366–381.

[70] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, Geometry from a time series, Phys. Rev. Lett. **45**, 712 (1980).

[71] G. Sugihara and R. M. May, Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series, Nature **344**, 734 (1990).

[72] T. Sauer, J. A. Yorke, and M. Casdagli, Embedology, J. Stat. Phys. **65**, 579 (1991).

[73] Y.-C. Lai and N. Ye, Recent developments in chaotic time series analysis, Int. J. Bif. Chaos **13**, 1383 (2003).

[74] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 1st ed. (Cambridge University Press, Cambridge, UK, 1997).

[75] P. Grassberger and I. Procaccia, Characterization of strange attractors, Phys. Rev. Lett. **50**, 346 (1983).

[76] Y.-C. Lai, D. Lerner, and R. Hayden, An upper bound for the proper delay time in chaotic time series analysis, Phys. Lett. A **218**, 30 (1996).

[77] Y.-C. Lai and D. Lerner, Effective scaling regime for computing the correlation dimension in chaotic time series analysis, Physica D **115**, 1 (1998).

[78] M. Belkin, Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation, Acta Numerica **30**, 203 (2021).

[79] M. I. Ribeiro, Kalman and extended Kalman filters: Concept, derivation and properties, Inst. Sys. Robot. **43**, 46 (2004).

[80] P. De Wilde, Class of Hamiltonian neural networks, Phys. Rev. E **47**, 1392 (1993).

[81] T. Bertalan, F. Dietrich, I. Mezić, and I. G. Kevrekidis, On learning Hamiltonian systems from data, Chaos **29**, 121107 (2019).

[82] A. Choudhary, J. F. Lindner, E. G. Holliday, S. T. Miller, S. Sinha, and W. L. Ditto, Physics-enhanced neural networks learn order and chaos, Phys. Rev. E **101**, 062207 (2020).

[83] A. Choudhary, J. F. Lindner, E. G. Holliday, S. T. Miller, S. Sinha, and W. L. Ditto, Forecasting Hamiltonian dynamics without canonical coordinates, Nonlin. Dyn. **103**, 1553 (2021).

[84] C.-D. Han, B. Glaz, M. Haile, and Y.-C. Lai, Adaptable Hamiltonian neural networks, Phys. Rev. Research **3**, 023156 (2021).

[85] H. Gao, L. Sun, and J.-X. Wang, Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels, Phys. Fluids **33**, 073603 (2021).

[86] J. Jiang and J. A. Fan, Global optimization of dielectric metasurfaces using a physics-driven neural network, Nano letters **19**, 5366 (2019).

[87] C.-D. Han, B. Glaz, M. Haile, and Y.-C. Lai, Tomography of time-dependent quantum Hamiltonians with machine learning, Phys.l Rev. A **104**, 062404 (2021).

[88] D. Svozil, V. Kvasnicka, and J. Pospichal, Introduction to multi-layer feed-forward neural networks, Chemome. Intell. Lab. Sys. **39**, 43 (1997).

[89] O. Baghirli, Comparison of lavenberg-marquardt, scaled conjugate gradient and bayesian regularization backpropagation algorithms for multistep ahead wind speed forecasting using multilayer perceptron feedforward neural network (Semantic Scholar, 2015).

[90] G. E. Hinton, Connectionist learning procedures, in *Machine Learning* (Elsevier, 1990) pp. 555–610.

[91] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches* (John Wiley & Sons, 2006).