NASA Public Access

Author manuscript

IEEE Trans Plasma Sci IEEE Nucl Plasma Sci Soc. Author manuscript; available in PMC 2019 August 01.

Published in final edited form as:

IEEE Trans Plasma Sci IEEE Nucl Plasma Sci Soc. 2018 August ; 46(8): 3090–3099. doi:10.1109/TPS.2018.2849940.

Identification of Plasma Waves at Saturn Using Convolutional Neural Networks

Suranga Ruhunusiri [Member, IEEE]

Department of Physics and Astronomy, The University of Iowa, Iowa City, IA 52242 USA
(suranga-ruhunusiri@uiowa.edu).

Abstract

Cassini has recently completed its 13-year mission at Saturn leaving a vast data set. A large interest among the scientific community is to investigate plasma waves and instabilities at Saturn. It is no longer feasible to manually search through Cassini's vast data set to identify all such waves of interest. Thus, the feasibility of using artificial neural networks (ANNs) to identify plasma waves at Saturn is demonstrated using Cassini data. A convolutional neural network (CNN) was trained to identify low-frequency plasma waves that occur in the upstream region of Saturn using images constructed from the Cassini magnetometer time series data. By systematically varying the network architecture during training and validation, a CNN was obtained that can identify upstream waves with an accuracy of $94\% \pm 2\%$. The CNN's high accuracy for wave identification demonstrates that it is, in fact, feasible to use ANNs to identify plasma waves at Saturn and by extension in other planetary and lunar plasma environments using spacecraft data.

Keywords

Artificial neural networks (ANNs); Cassini, convolutional neural networks (CNNs); plasma waves, Saturn

I. Introduction

A. Plasma Waves at Saturn

Saturn upstream region and the magnetosphere are home to a variety of plasma waves. Plasma and magnetic field measurements by Voyagers 1 and 2, and recently by the Cassini spacecraft enabled the identification of these waves. Low-frequency waves with frequencies of few millihertz were commonly observed upstream of the bow shock [1]–[3] [see Fig. 1(a) for an example] and are inferred to be excited by upstream ion populations via mechanisms such as resonance [2]. Short duration waves with frequencies of few tens of millihertz were also observed in the upstream region of Saturn [see Fig. 1(b) for an example]. These waves are referred to as whistler precursor waves and they are inferred to be excited by a nonlinear process during the steepening of the low-frequency waves similar to the one shown in Fig.

LICENSING INFORMATION Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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

1(a) [2]. Water group ion cyclotron waves that have frequencies near few hundred millihertz were observed in Saturn's magnetosphere [4]–[11] [see Fig. 1(c) for an example]. These water group ion cyclotron waves are interpreted to be excited by the pickup of ionized E-ring material [6]. These low-frequency waves are typically quasi-monochromatic and can last from few minutes to several tens of minutes. Plasma waves with the frequencies of several hundred hertz to several kilohertz were observed in Saturn's magnetosphere. These include lightning induced whistler waves [12], electron cyclotron harmonic emissions [13], and whistler mode chorus emissions [14]. Plasma waves with frequencies of several kilohertz were also observed in the upstream region of Saturn [15], [16]. All these waves are of great interest to the scientific community as their investigations not only reveal the plasma environment dynamics at Saturn but also enable them to unravel how plasma waves are generated and damped.

Identification of these waves is typically done by manually surveying spacecraft instrument data. With a mission like Cassini, which has gathered 13 years of data at Saturn, it is no longer feasible to manually search through the entire data set to identify all such waves of interest. Thus, good automated methods need to be developed. The objective of this paper is to demonstrate the feasibility of using artificial neural networks (ANNs) to identify plasma waves at Saturn. The methods developed in this paper should be of interest to not only the Saturn scientific community but also to a larger scientific community who are involved in investigating waves at other planetary and lunar plasma environments.

B. Artificial Neural Networks

ANNs are computational units which can be trained to respond to observational data and are inspired by biological neural networks which can be trained to respond to external stimuli from the environment. Thus, ANNs are ideal tools for tasks such as pattern recognition. ANNs have been successfully used in a variety of pattern recognition tasks such as gravitational wave detection [17]–[19], exoplanet discovery by surveying light curves obtained by space telescopes [20], brain wave feature extraction and classification [21], seismic wave detection and identification [22], character recognition [23], and voice recognition [24]. ANNs are also used in more familiar tools that we use every day, for example, in the Google search engine and in the iPhone personal assistant Siri [25].

ANNs have also been used in several space physics applications. For example, Wu and Lundstedt [26] used ANNs to predict the geomagnetic storm conditions or the disturbance storm-time index (Dst) from the solar wind data. In this example, both magnetic field and plasma parameters were fed into an ANN as inputs and the ANN predicted the Dst index. Cander [27] used ANNs for modeling and predicting the temporal and spatial variation of the ionospheric conditions. Barkhatov and Revunov [28] used ANNs for the classification of discontinuities in space plasma using WIND spacecraft observations. Muret and Omidi [29] used ANNs to study the formation and evolution of low-frequency plasma waves generated in a hybrid simulation that modeled the upstream plasma environments of planets and comets. To date, ANNs have not been used in space physics applications involving identification of plasma waves using spacecraft data. Thus, this is the first time that the ANNs would be employed for such an application.

In this paper, I will describe the use of a class of ANNs called the convolutional neural networks (CNNs) [30] to identify the plasma waves at Saturn. These CNNs are currently gaining in popularity due to their powerfulness at performing pattern recognition tasks. A CNN has multiple internal layers and can contain several hundred neurons in each layer. One advantage of these networks is that they can be trained to identify waveforms of interest directly using raw time series data as inputs without the need to perform a Fourier transform on the time series data first [31]. Thus, the need to perform the computationally expensive Fourier transform for the network input data generation is removed. This is very attractive for applications involving analyzing large data sets such as Cassini's.

II. Method

I trained, validated, and tested a CNN in MATLAB to distinguish the whistler precursor waves shown in Fig. 2(a) from background turbulence (or times with no such waves) shown in Fig. 2(b) and magnetometer artifacts shown in Fig. 2(c) and (d). The whistler precursor waves are frequently observed in the upstream region of Saturn and they typically have time durations of 2 min or less. These waves were selected to train, validate, and test the CNN, as these waves can be easily identified via a manual survey of magnetometer data. First, I will describe the strategy for selecting data for training, validation, and testing. Then, I will describe the CNN architecture and the procedures for training, validation, and testing.

A. Data Selection for Training, Validation, and Testing

Implementation of a CNN for wave identification involves three steps: training, validation, and testing. Training and validation are used to select the best CNN architecture. Testing is used to assess the ability of the CNN to identify waves from background turbulence and magnetometer artifacts. I first trained and validated a CNN using a 10-fold cross-validation technique, which will be described in detail in Section II-C, using a training and validation data set. The training and validation data set contained waves, background turbulence, and magnetometer artifacts, similar to the examples shown in Fig. 2(a)–(d), respectively. During training, the CNN learns features within waves, background turbulence, and magnetometer artifacts that enables it to distinguish them. For testing the network, similarly, I prepared a testing data set which also included waves, background turbulence, and magnetometer artifacts that were not included in the training and validation data set.

To construct the training, validation, and testing data sets of waves, I manually searched the Cassini fluxgate magnetometer [32] data from 2004 to 2011, only selecting times when Cassini was located upstream of the bow shock to identify the upstream whistler precursor waves similar to the one shown in Fig. 2(a). In particular, Cassini 1-second averaged magnetometer data were used for the manual identification of these waves. I was able to identify 282 whistler precursor waves (the times where these waves are observed are provided at <https://github.com/sruhunusiri/Supplementary-Material-for-Identification-of-Plasma-Waves-at-Saturn-Using-Convolutional-Neural-Nets>). Since the time duration for a single wave packet is typically 2 min or less, I selected 2-min time intervals centered on these wave packets and converted the B_x , B_y , and B_z components of the magnetic field to 1-D 8-bit RGB JPEG images (see Fig. 3 for an example of these JPEG images). In these

images, the red, green, and blue channels contain the information of B_x , B_y , and B_z respectively. Each image contains 120 pixels because an image was constructed from a 2-min interval of the magnetic field data which had a cadence of 1 s. Each pixel value in a color channel depicts the normalized magnetic field component amplitude as defined by $[B_i(t) - \min(B_i(t))] / \max[B_i(t) - \min(B_i(t))]$ scaled to 8-bit value, where $i = x, y$, or z components of the magnetic field. These JPEG images were used as inputs to the CNN. I used 230 JPEG images of waves (~80%) for training and validation and the remainder (i.e., 52 wave images or ~20%) for testing.

For preparing the training, validation, and testing data sets of background turbulence and magnetometer artifacts, I selected 2-min time intervals in the magnetic field time series that has no upstream whistler precursor waves, other low-frequency waves [similar to the ones shown in Fig. 1(a)]. Then, I converted the B_x , B_y , and B_z magnetic field time series to 8-bit RGB JPEG images as described above. The training and validation data set contained 8000 images while the testing set contained 2000 images (80% and 20% of images, respectively). These images of background turbulence and magnetometer artifacts are provided at <https://github.com/sruhunusiri/Supplementary-Material-for-Identification-of-Plasma-Waves-at-Saturn-Using-Convolutional-Neural-Nets>. Henceforth, I will refer to the data set containing both background turbulence and magnetometer artifacts simply as the data set of the turbulence and the accuracy associated with the classification of these data by the CNN as the turbulence identification accuracy for convenience.

A reader may ask at this point, what was the motivation for using images rather than using the B_x , B_y , and B_z directly as inputs to the CNN. The motivation was twofold. First, the 1-s averaged magnetic field data represent a large data set. Thus, instead of directly loading this data set to MATLAB workspace, the datastore functionality [33] was used which creates a repository for a collection of data that are too large to fit in memory and images are a convenient format to use with this datastore functionality. Second, some spacecraft instrument data are distributed in image format. For example, electric field and magnetic field fluctuations measured by the Cassini radio and plasma wave instrument [34] are provided as spectrogram images (these data are available at <https://pdspipi.igpp.ucla.edu>). Thus, the aim was to develop a CNN that can be quickly adapted to use with other data sets as well.

B. Convolutional Neural Network Architecture

I adapted a MATLAB CNN [35] that is originally intended for a character recognition task for identifying waves at Saturn. I will briefly describe the network architecture below. The MATLAB codes used for constructing, training and validation, and testing the CNN are provided at <https://github.com/sruhunusiri/Supplementary-Material-for-Identification-of-Plasma-Waves-at-Saturn-Using-Convolutional-Neural-Nets>.

The CNN contains eight layers: an image input layer, a convolutional layer, a rectified linear layer, a max-pooling layer, a dropout layer, a fully connected layer, a softmax layer, and a classification layer (see Fig. 4 for a diagram of the CNN architecture). A layer is a dedicated subunit in the CNN that performs a specific operation on an image input.

The image input layer size was set to 3×120 as the inputs are RGB JPEG images which have three separate channels (red, green, and blue). I experimented with several values for the filter size N and the number of filters M in the convolutional layer to determine the values that yielded the highest wave identification accuracy as will be described later in Section II-C. I selected the smallest stride value allowable which is 1 for the convolutional layer. I selected a variable value of R as the pool size and a value of 1 as the stride size for the max-pooling layer. I also experimented with several values of R to determine the values that yielded the highest wave identification accuracy. A dropout layer was used with a 0.5 dropout probability, as a use of such a layer can prevent the CNN from overfitting [36]. The overfitting is a phenomenon in which the network memorizes the inputs rather than learning the general features in the inputs to perform the classification task. A detailed description of the CNN architecture is provided in the Appendix for the interested reader.

C. Training, Validation, and Testing the CNN

The CNN was trained, validated, and tested on a PC with a single GPU (NVIDIA GeForce GTX 750 Ti). The CNN was trained using an initial learning rate of 0.0001 for a maximum epoch number of 400 using the stochastic gradient descent with momentum algorithm [37]. An epoch number of 400 was selected because the training accuracies for waves and turbulence begin to flatten beyond an epoch number of 350 as can be seen in Fig. 5. Selection of larger values can lead to overfitting.

A 10-fold cross validation [38] was used to determine the best architecture of the CNN that yields the highest accuracies. The motivation for doing a 10-fold or generally a k -fold cross validation is to remove any biases due to the selection of the training and validation data. To create the cross-validation data set, first, the 230 wave images and the 8000 turbulence images in the training and validation data set were separately randomized. Then, 10 subsets were created each containing 23 wave images and 800 turbulence images. Nine subsets were used for training and one subset was used to validate the CNN. This was repeated 10 times by selecting a different combination of nine subsets for training and one subset for validation for the same CNN architecture. The validation accuracies for the 10-fold cross validation is shown for a representative CNN architecture in Fig. 6. This 10-fold cross validation was then repeated for different CNN architectures. Discrete values for max-pooling layer pool size R , filter size N , and a number of filters M were selected for the cross validation as it was not feasible to explore all the possible values for these parameters. Specifically, pool sizes of 4, 8, 16, and 32 were selected. For each of the pool sizes, the filter size was varied from 2 to 64 in powers of 2 and the number of filters was varied from 2 to 64 in powers of 2. Thus, 10-fold cross validation was performed for 144 different CNN architectures and the results are displayed in Fig. 7; these results will be discussed in Section III.

After selecting the CNN with the highest accuracy from the 10-fold cross validation, it was trained using all the train and validation data sets. Then, this CNN was tested using the testing set of images using bootstrap with replacements [38]. This is very useful for scenarios with small data set sizes, such as the case here, where the data set size of the waves is small. In this bootstrap method, the 52 images of waves in the testing data set were bootstrapped to create 1000 different samples consisting of 52 wave images (these different

samples contain the same number of images in the original data set because they contain replacements as mentioned before). The 2000 images of turbulence in the testing data set were used to create 1000 different samples of 2000 turbulence images. Then, the separate 1000 samples of waves and turbulence were combined to create 1000 testing data sets. Finally, the CNN was tested using the 1000 testing data sets of images one at a time. Histograms of these testing accuracies for wave and turbulence identification are depicted in Fig. 8. These bootstrap results will be discussed in Section III.

III. Results

The mean validation accuracies for wave and turbulence identification along with the average accuracies, determined from the 10-fold cross validation, are depicted in Fig. 7 as a function of the three CNN architecture parameters: filter size N , number of filters M , and pool size R . The error bars depict the minimum and maximum accuracies for the 10-fold cross validation. The wave identification accuracy is the percentage ratio of the number of wave images in a validation data set that is accurately identified by the CNN to the total number of wave images in that validation data set. Similarly, the turbulence identification accuracy is the percentage ratio of the number of turbulence images in a validation data set that is accurately identified by the CNN to the total number of turbulence images in that validation data set. The average accuracy is the average of the wave and turbulence identification accuracies.

The cross-validation results, shown in Fig. 7, enable to identify the CNN architecture that yields the maximum accuracy for both wave and turbulence identification. Note that having a high accuracy for turbulence identification is crucial as the images of turbulence that are not identified as turbulence are classified by the CNN as waves. Thus, having a high accuracy for turbulence identification minimizes the number of false positives.

As can be seen in Fig. 7, the wave identification accuracies often lie between 70% and 100%, whereas the turbulence identification accuracies always take values between 99% and 100%. For a given pool size and filter size, the wave identification accuracies tend to increase as the number of filters is increased. Also, for a given pool size and number of filters, the wave identification accuracies first tend to increase as the filter size is increased up to values between 8 and 16, then the accuracies begin to decrease as the filter size is increased beyond those values.

To find the architecture parameters that yield the highest accuracies for both wave and turbulence identifications, first, for a fixed number of pool size, the 2-D parameter space of accuracies was searched to find the values of the filter size and the number of filters which correspond to the maximum average accuracy. These results are summarized in Table I. As can be seen in Table I, the CNN parameters that yield the highest average accuracy, as well as the largest separate accuracies for wave and turbulence identification, are pool size of 4, filter size of 16, and 16 filters. This CNN will henceforth be referred to as CNN1.

The bootstrap results for testing of CNN1 indicate that it has excellent ability to distinguish waves and turbulence (see Fig. 8). In particular, it has a wave identification accuracy of 94%

$\pm 2\%$ and turbulence identification accuracy of $99.9\% \pm 0.1\%$. The uncertainties are determined from the lower and upper quartiles of the accuracy distributions shown in Fig. 8.

IV. Discussion

In addition to the CNN architecture described above, another CNN architecture that can be used for pattern recognition tasks is an architecture which contains multiple convolutional, rectified linear, and max-pooling layers, which is referred to as a deep neural network [31]. I did not experiment with these networks comprising of complex architectures, as the simple architecture of CNN1 has the capability to identify waves and turbulence with a very high accuracy.

V. Summary

The feasibility of using ANNs to identify the plasma waves at Saturn was demonstrated by training, validating, and testing a CNN to identify the whistler precursor waves, which are a type of waves that are observed in the upstream region of Saturn, using the Cassini magnetometer data. Testing of the CNN revealed that it can identify these waves with a very high accuracy demonstrating that such ANNs can indeed be used to identify plasma waves at Saturn. Similar ANNs will be useful for identifying plasma waves in other planetary and lunar plasma environments using spacecraft data.

Acknowledgement

The author would like to thank A. Sulaiman for providing Cassini bow shock crossing times and also for useful discussions. He would also like to thank M. K. Dougherty for Cassini magnetometer data. The Cassini magnetometer data are publicly available at <https://pds-ppi.igpp.ucla.edu>.

This work was supported by NASA through the Cassini Data Analysis Program under Grant NNX17AD40G. The review of this paper was arranged by Senior Editor S. T. Lai.

Appendix

CNN Architectur: A Detailed View

Here, I provide a detailed description of the CNN architecture. The first layer or the image input layer is responsible for reading each pixel value from the JPEG images which are used as network inputs. Thus, it should have a size equal to the number of channels by image size. Since the images, I use, have a size of 1×120 and have the three color channels, the image input layer size was set to 3×120 (see Fig. 4).

The second layer or the convolutional $\times$ layer has a set of filters that are encoded with features that enable to distinguish waves from turbulence. A filter is a set of weights F_i that are responsive to certain features in the images that produces a high activation for those features. For each image channel, there are separate filters. These filters do not need to be user specified and the CNN learns the features during training and adjusts the filter weights F_i . The size of the filters and the number of filters in the convolutional layer, however, need to be user specified. Since the image input layer of the CNN is 1-D, the filter size also needs to be 1-D. Thus, I specified $1 \times N$ as the filter size and selected M number of filters. I

experimented with several values for N and M to determine the values that yielded the highest wave identification accuracy as described in Section II-C.

The outputs of the convolutional layer are M feature maps or a feature map for each of the M number of filters. A feature map is generated by computing the dot product of the pixel value read by the image input layer with the filter weight F_i for a subset of consecutive N image inputs and adding a bias b_i i.e., $(\sum_{i=1}^N R_i F_i + G_i F_i + B_i F_i) + b_i$ (see Fig. 4). Here, R_i , G_i , and B_i are the pixel values for the red, green, and blue channels, respectively. This step is repeated for all the inputs from the image input layer by advancing across the inputs by a user-selectable step size referred to as a stride. I selected the smallest stride value allowable which is 1. In general, a larger stride value should be used for images with a large number of pixels to decrease the training time. The computation of the dot product and striding along the image inputs generate a feature map that is $1 \times Q$ in size, where $Q = 1 + (120 - N)/\text{stride}$. For example, if $N=10$ and M is the convolution layer outputs 40 feature maps with a size of $1 \times [1 + (120 - 10)/1 = 111]$.

The third layer or the rectified linear layer inspects each of the feature maps generated by the convolution layer and sets each element within a feature map to zero if the value is negative but leaves the element value unaltered if it is positive (see Fig. 4). Thus, the rectified linear layer outputs a modified set of feature maps (with a size of $1 \times Q$) that have element values of either 0 or greater.

The fourth layer in the CNN or the max-pooling layer generates an output by selecting the maximum number in a subset of inputs in the output from the rectified linear layer (see Fig. 4). This is repeated for all the elements in the rectified linear layer output by advancing every number of steps referred to as a stride. The size of the subset referred to as the pool size, and the stride needs to be user specified. I selected a variable value of R as the subset size and a value of 1 as the stride size. Thus, the max-pooling layer produces M number of outputs each with a size of $1 \times Z$ where $Z = 1 + (Q - R)/\text{stride}$. For example, for a CNN with $N=10$, and $M=40$, setting $R=10$ yields $Z = 1 + (111 - 10)/1 = 102$. Thus, in this example, the max-pooling layer produces 40 outputs each with a size of 1×102 .

The fifth layer is the dropout layer and it contains neurons that are connected to all the elements of the max-pooling layer output and randomly sets the input elements to zero based on a user-specified probability to reduce overfitting. Here, I used a dropout probability of 0.5.

The sixth layer or the fully connected layer contains neurons that are connected to all the elements of the dropout layer output. The seventh layer or the softmax layer uses the outputs from the fully connected layer to generate probabilities for the image input to be either a wave or turbulence. The final layer is the classification layer and it classifies the image input as either a wave or turbulence using the probabilities output by the softmax layer.

In summary, layers 1 through 4 of the CNN extract useful features from the images and simplify them for the subsequent layers. Layers 5 through 8 use the image feature information from the preceding layers to classify the inputs.

References

- [1]. Orlowski DS , Russell CT , and Lepping RP , “Wave phenomena in the upstream region of Saturn,” J. Geophys. Res., Space Phys, vol. 97 no. A12, pp. 19187–19199, 1992.
- [2]. Bertucci C et al., “Low-frequency waves in the foreshock of Saturn: First results from Cassini,” J. Geophys. Res., Space Phys, vol. 112, no. A9, p. A09219, 2007.
- [3]. Andrés N , Gómez DO , Bertucci C , Mazelle C , and Dougherty MK , “Saturn’s ULF wave foreshock boundary: Cassini observations,” Planet. Space Sci, vols. 79–80, pp. 64–75, 5 2013.
- [4]. Smith EJ and Tsurutani BT , “Saturn’s magnetosphere: Observations of ion cyclotron waves near the Dione L shell,” J. Geophys. Res., Space Phys, vol. 88, no. A10, pp. 7831–7836, 1983.
- [5]. Barbosa DD , “Theory and observations of electromagnetic ion cyclotron waves in Saturn’s inner magnetosphere,” J. Geophys. Res., Space Phys, vol. 98, no. A6, pp. 9345–9350, 1993.
- [6]. Leisner JS , Russell CT , Dougherty MK , Blanco-Cano X , Strangeway RJ , and Bertucci C , “Ion cyclotron waves in Saturn’s E ring: Initial Cassini observations,” Geophys. Res. Lett, vol. 33, no. 11, p. L11101, 2006.
- [7]. Russell CT , Leisner JS , Arridge CS , Dougherty MK , and Blanco-Cano X , “Nature of magnetic fluctuations in Saturn’s middle magnetosphere,” J. Geophys. Res., Space Phys, vol. 111, no. A12, p. A12205, 2006.
- [8]. Rodríguez-Martínez M , Blanco-Cano X , Russell CT , Leisner JS , Wilson RJ , and Dougherty MK , “Harmonic growth of ion-cyclotron waves in Saturn’s magnetosphere,” J. Geophys. Res., Space Phys, vol. 115, no. A9, p. A09207, 2010.
- [9]. Meeks Z , Simon S , and Kabanovic S , “A comprehensive analysis of ion cyclotron waves in the equatorial magnetosphere of Saturn,” Planet. Space Sci, vol. 129, pp. 47–60, Sep. 2016.
- [10]. Meeks Z and Simon S , “Magnetic signatures of ion cyclotron waves during Cassini’s high-inclination orbits of Saturn,” Planet. Space Sci, vol. 136, pp. 34–45, Feb. 2017.
- [11]. Chou M and Cheng CZ , “Distribution of water-group ion cyclotron waves in Saturn’s magnetosphere,” Earth, Planets Space, vol. 69, p. 122, Dec. 2017.
- [12]. Akalin F et al., “First whistler observed in the magnetosphere of Saturn,” Geophys. Res. Lett, vol. 33, no. 20, p. L20107, 2006.
- [13]. Menietti JD , Averkamp TF , Kurth WS , Ye S-Y , Gurnett DA , and Cecconi B , “Survey of Saturn electrostatic cyclotron harmonic wave intensity,” J. Geophys. Res. Space Phys, vol. 122, no. 8, pp. 8214–8227, 2017.
- [14]. Menietti JD et al., “Survey analysis of chorus intensity at Saturn,” J. Geophys. Res. Space Phys, vol. 119, no. 10, pp. 8415–8425, 2014.
- [15]. Pířa D et al., “Statistics of Langmuir wave amplitudes observed inside Saturn’s foreshock by the Cassini spacecraft,” J. Geophys. Res. Space Phys, vol. 120, no. 4, pp. 2531–2542, 2015.
- [16]. Pířa D , Santolík O , Hospodarsky GB , Kurth WS , Gurnett DA , and Sou ek J , “Spatial distribution of Langmuir waves observed upstream of Saturn’s bow shock by Cassini,” J. Geophys. Res. Space Phys, vol. 121, no. 8, pp. 7771–7784, 2016.
- [17]. George D and Huerta EA , “Deep neural networks to enable real-time multimessenger astrophysics,” Phys. Rev. D, Part. Fields, vol. 97, no. 4, p. 044039, 2017.
- [18]. George D , Shen H , and Huerta EA . (2017). “Glitch classification and clustering for LIGO with deep transfer learning.” [Online]. Available: <https://arxiv.org/abs/1711.07468>
- [19]. George D and Huerta EA , “Deep Learning for real-time gravitational wave detection and parameter estimation: Results with advanced LIGO data,” Phys. Lett. B, vol. 778, pp. 64–70, Mar. 2018.
- [20]. Pearson KA , Palafox L , and Griffith CA . (2017). “Searching for exoplanets using artificial intelligence.” [Online]. Available: <https://arxiv.org/abs/1706.04319>
- [21]. Schirmermeister RT et al., “Deep learning with convolutional neural networks for EEG decoding and visualization,” Hum. Brain Mapping, vol. 38, no. 11, pp. 5391–5420, 2017.
- [22]. Kaur K , Wadhwa M , and Park EK , “Detection and identification of seismic p-waves using artificial neural networks,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Dallas, TX, USA, Aug. 2013, pp. 1–6.

- [23]. Arnold R and Miklós P , “Character recognition using neural networks,” in Proc. 11th Int. Symp. Comput. Intell. Inform., Nov. 2010, pp. 311–314.
- [24]. Abdel-Hamid O , Mohamed A-R , Jiang H , Deng L , Penn G , and Yu D , “Convolutional neural networks for speech recognition,” IEEE/ACM Trans. Audio, Speech Language Process, vol. 22, no. 10, pp. 1533–1545, Oct. 2015.
- [25]. Team Siri , “Deep learning for Siri’s voice: On-device deep mixture density networks for hybrid unit selection synthesis,” Mach. Learn. J, vol. 1, no. 4, 2014 [Online]. Available: <https://machinelearning.apple.com/2017/08/06/siri-voices.html>
- [26]. Wu J-G and Lundstedt H , “Geomagnetic storm predictions from solar wind data with the use of dynamic neural networks,” J. Geophys. Res. Space Phys, vol. 102, no. A7, pp. 14255–14268, 1997.
- [27]. Cander LR , “Artificial neural network applications in ionospheric studies,” ANNALI GEOFISICA, vol. 41, pp. 757–766, Nov. 1998.
- [28]. Barkhatov NA and Revunov SE , “Neural network classification of discontinuities in space plasma parameters,” Geomagnetism Aeronomy, vol. 50, no. 7, pp. 894–904, 2010.
- [29]. Muret P and Omid N , “Analysis of steepened magnetosonic waves using wavelet transforms and neural networks,” J. Geophys. Res. Space Phys, vol. 100, no. A12, pp. 23465–23479, 1995.
- [30]. LeCun Y and Bengio Y , “Convolutional networks for images, speech, and time series,” in The Handbook of Brain Theory and Neural Networks, Arbib MA , Ed., 2nd ed Cambridge, MA, USA: MIT Press, 2003, pp. 276–279.
- [31]. Dai W , Dai C , Qu S , Li J , and Das S . (2016). “Very Deep Convolutional Neural Networks for Raw Waveforms.” [Online]. Available: <https://arxiv.org/abs/1610.00087>
- [32]. Dougherty MK et al., “The Cassini magnetic field investigation,” Space Sci. Rev, vol. 114, nos. 1–4, pp. 331–383, 2004.
- [33]. (2018). Getting Started with Datastore. [Online]. Available: https://www.mathworks.com/help/matlab/import_export/what-is-adatastore.html and <https://www.Mathworks.com>
- [34]. Gurnett DA et al., “The Cassini radio and plasma wave investigation,” Space Sci. Rev, vol. 114, nos. 1–4, pp. 395–463, 2004.
- [35]. (2018). Create Simple Deep Learning Network for Classification. [Online]. Available: <https://www.mathworks.com/help/nnet/examples/create-simple-deep-learning-network-for-classification.html> and <https://www.Mathworks.com>
- [36]. Srivastava N , Hinton G , Krizhevsky A , Sutskever I , and Salakhutdinov R , “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res, vol. 15, no. 1, pp. 1929–1958, 2014.
- [37]. (2018). Set Up Parameters and Train Convolutional Neural Network. [Online]. Available: <https://www.mathworks.com/help/nnet/ug/setting-up-parameters-and-training-of-a-convnet.html> and <https://www.Mathworks.com>
- [38]. Kohavi R , “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in Proc. 14th Int. Joint Conf. Artif. Intell, vol. 2, 1995, pp. 1137–1143.

Fig. 1.

(a)–(c) Three examples of the plasma waves observed by the Cassini magnetometer at Saturn. (a) Upstream low-frequency wave with a frequency of few millihertz. (b) Whistler precursor wave observed in the upstream region of Saturn which has a frequency of few tens of millihertz. (c) Water group ion cyclotron wave observed in the magnetosphere which has a frequency of few hundred millihertz. B_x , B_y , and B_z are the magnetic field components in kronocentric solar orbital coordinates, where X points from Saturn to Sun, Z is parallel to the Saturn's orbital plane upward normal, and Y completes the right-handed coordinate system. $|B|$ is the magnetic field magnitude. These plots are based on 1-s averaged magnetometer data.

Fig. 2.

(a) Time segment with a whistler precursor wave. (b) Time segment with background turbulence. (c) and (d) Commonly observed Cassini magnetometer artifacts. Artifact from a spacecraft roll which was used for magnetometer calibration purposes is shown in (c). Another frequently observed magnetometer artifact is shown in (d). A CNN was trained to distinguish waves shown in (a) from background turbulence shown in (b) as well as the magnetometer artifacts shown in (c) and (d).

Fig. 3.

Example of a 1-D JPEG image constructed from the Cassini magnetic field time series data. The red, green, and blue channels of the JPEG image are constructed using the B_x , B_y , and B_z component values of the magnetic field. Similar images were used to train, validate, and test the CNN.

Fig. 4.

Architecture of the CNN employed for the identification of plasma waves at Saturn. The network contains eight layers: an image input layer, a convolutional layer, a rectified linear layer, a max-pooling layer, a dropout layer, a fully connected layer, a softmax layer, and a classification layer.

Fig. 5.

(a) and (b) Training accuracy versus epoch number for wave and turbulence identification, respectively, for two representative CNNs with a filter size of 16, 8 filters, and a pool size of 4 and a filter size of 64, 16 filters, and a pool size of 8. The training accuracies for CNNs do not improve significantly beyond ~350 epochs. Thus, a maximum epoch number of 400 were used to train the CNNs to avoid overfitting

Fig. 6.

(a) and (b) Validation accuracy versus iteration for wave and turbulence identification, respectively, for a representative CNN with a filter size of 4, 4 filters, and a pool size of 4 determined from the 10-fold cross-validation procedure. The 10-fold cross validation was repeated for the 144 different CNN architectures to determine the best CNN architecture that has the highest accuracies.

Fig. 7.

Mean validation accuracies for wave and turbulence identification determined from the 10-fold cross validation procedure for 144 different CNN architectures. The error bars depict the minimum and the maximum accuracies determined from the 10-fold cross validation. These results were used to determine the best CNN architecture that has the highest accuracies.

Fig. 8.

(a) and (b) Distributions of testing accuracies for wave and turbulence identification for CNN1 determined from the bootstrap method. CNN1 has a wave identification accuracy of $94\% \pm 2\%$ and a turbulence identification accuracy of $99.9\% \pm 0.1\%$. The uncertainties are determined from the lower and upper quartiles of the accuracy distributions. The results indicate that CNN1 has a very high accuracy for distinguishing waves from turbulence.

TABLE I

CNN Architecture Parameters That Yield the Highest Validation Accuracies for Wave and Turbulence Identification

R	N	M	Average accuracy (%)	Wave identification accuracy (%)	Turbulence identification accuracy (%)
4	16	16	97.9(93.4,100.0)	96.0(86.9,100.0)	99.9(99.6,100.0)
8	32	32	97.3(93.4,100.0)	94.7(86.9,100.0)	99.8(99.5,100.0)
16	8	16	97.3(93.2,100.0)	94.7(86.9,100.0)	99.9(99.6,100.0)
32	8	32	96.4(91.0,100.0)	93.0(82.6,100.0)	99.8(99.5,100.0)

Notes: The values within the bracket denote the minimum and maximum values of accuracies determined from the 10-fold cross validation.

The CNN with R=4, N=16, and M=16 has the highest average accuracy and the highest separate accuracies for waves and turbulence identification