

TECHNICAL
REPORTS:
METHODS

10.1002/2015JA022132

Key Points:

- Developed neural network model to infer upper hybrid resonance line from plasma wave observations
- Applied model to 2425 orbits of Van Allen Probes data to determine electron number density
- Initial analysis of electron number density database as a function of L and MLT

Correspondence to:

I. S. Zhelavskaya,
irina.zhelavskaya@gmail.com

Citation:

Zhelavskaya, I. S., M. Spasojevic, Y. Y. Shprits, and W. S. Kurth (2016), Automated determination of electron density from electric field measurements on the Van Allen Probes spacecraft, *J. Geophys. Res. Space Physics*, 121, 4611–4625, doi:10.1002/2015JA022132.

Received 5 NOV 2015

Accepted 4 MAY 2016

Accepted article online 13 MAY 2016

Published online 28 MAY 2016

Automated determination of electron density from electric field measurements on the Van Allen Probes spacecraft

I. S. Zhelavskaya¹, M. Spasojevic², Y. Y. Shprits^{1,3}, and W. S. Kurth⁴

¹Department of Earth, Planetary, and Space Sciences, University of California, Los Angeles, California, USA, ²Department of Electrical Engineering, Stanford University, Stanford, California, USA, ³Helmholtz Centre Potsdam, GFZ German Research Centre For Geosciences and University of Potsdam, Potsdam, Germany, ⁴Department of Physics and Astronomy, University of Iowa, Iowa City, Iowa, USA

Abstract We present the Neural-network-based Upper hybrid Resonance Determination (NURD) algorithm for automatic inference of the electron number density from plasma wave measurements made on board NASA's Van Allen Probes mission. A feedforward neural network is developed to determine the upper hybrid resonance frequency, f_{uhr} , from electric field measurements, which is then used to calculate the electron number density. In previous missions, the plasma resonance bands were manually identified, and there have been few attempts to do robust, routine automated detections. We describe the design and implementation of the algorithm and perform an initial analysis of the resulting electron number density distribution obtained by applying NURD to 2.5 years of data collected with the Electric and Magnetic Field Instrument Suite and Integrated Science (EMFISIS) instrumentation suite of the Van Allen Probes mission. Densities obtained by NURD are compared to those obtained by another recently developed automated technique and also to an existing empirical plasmasphere and trough density model.

1. Introduction

The electron plasma density, n_e , is a fundamental parameter of plasma. The distribution of electron density in the inner magnetosphere is a critical parameter for wave-particle interactions and is necessary for quantifying the role of plasma waves in the formation and decay of the Earth's radiation belts [e.g., Thorne *et al.*, 2013]. The electron density is required for the prediction of the evolution of the space environment [e.g., Artemyev *et al.*, 2013; Summers and Thorne, 2003] and is also important for the analysis of satellite anomalies [e.g., Reeves *et al.*, 2013].

Numerous empirical models of plasma electron density have been developed, and Carpenter and Anderson [1992] and Sheeley *et al.* [2001] are the most widely used in recent years. Carpenter and Anderson [1992] was based on electron density measurements deduced from sweep frequency receiver (SFR) radio measurements on the International Sun-Earth Explorer (ISEE-1) spacecraft as well as ground-based whistler measurements. The model covers the range of $2.25 < L < 8$ and the local time interval of 0 to 15 MLT. The plasmasphere model in Carpenter and Anderson [1992] is a saturated density model and, as such, represents the density distribution after several days of refilling. On the other hand, the Sheeley *et al.* [2001] plasmasphere and plasma trough models are statistical averages based on the densities derived from the Combined Release and Radiation Effects Satellite (CRRES) swept frequency receiver by identifying the upper hybrid resonance frequency and cover the range of $3 \leq L \leq 7$ for all local times. Sheeley *et al.* [2001] also provide the standard deviation of measurements for the plasmasphere and trough models to represent depleted or saturated density levels.

Although these models are commonly used in space physics simulations, in some events such as geomagnetic storms, empirical density models cannot provide reliable density estimates since they are parameterized only by L (and by MLT for trough density models), and density levels, particularly in the outer plasmasphere, are known to be variable during storm times [e.g., Park and Carpenter, 1970; Park, 1974; Carpenter *et al.*, 1993; Moldwin *et al.*, 1995]. The empirical models are also based on relatively limited data sets. Therefore, it is of continuing interest to collect reliable databases of electron density measurements over a wide range of geomagnetic conditions.

One method to determine the electron density is to derive it from intense upper hybrid band radio emissions [Mosier *et al.*, 1973]. This method is considered to be the most reliable passive technique when the electron

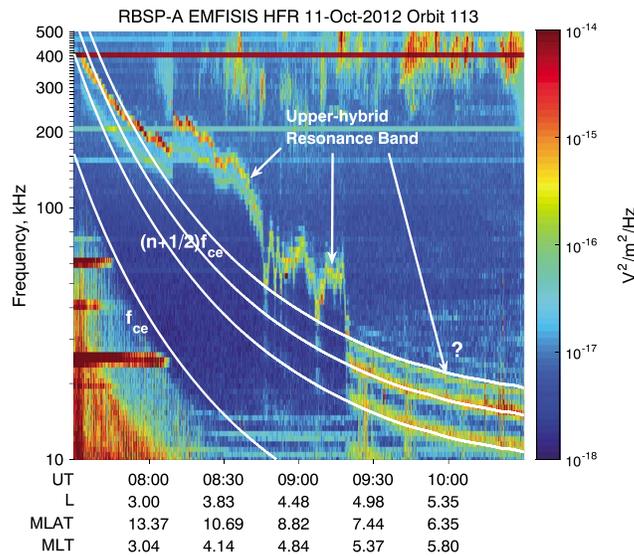


Figure 1. An example spectrogram from the EMFISIS HFR showing the electric power spectral density as a function of frequency (log scale) and time for the outbound portion of orbit 113 from 11 October 2012. The upper hybrid resonance band is identified along with the gyrofrequency and its half harmonics corresponding to $n = 2, 3,$ and 4 .

permittivity of free space. The UHR noise band typically extends from f_{uhr} down to f_{pe} with a generally less intense emission extending down to the z mode cutoff frequency. The upper limit of the upper hybrid noise band is often the most pronounced cutoff in spacecraft plasma wave data. f_{uhr} is generally considered to be the upper frequency edge of the upper hybrid band enhancement rather than the emission peak intensity [e.g., *Beghin et al., 1989; Benson et al., 2004*].

In August 2012, the Van Allen Probes were launched with the primary goal of studying the dynamics of the Earth’s radiation belts [*Mauk et al., 2012*]. The Electric and Magnetic Field Instrument Suite and Integrated Science (EMFISIS) [*Kletzing et al., 2013*] on Van Allen Probes makes routine electric field measurements in the frequency range of 10 to 487 kHz in order to identify the upper hybrid resonance band, thus providing an accurate estimation of the electron density. An example of data from the EMFISIS High Frequency Receiver (HFR) on board RBSP-a (Radiation Belt Storm Probes) is shown in Figure 1 for the outbound portion of orbit 113. HFR provides the electric field power spectral density as a function of frequency and time. The upper hybrid resonance band is indicated. The background magnetic field strength is directly measured by the magnetometer on board the spacecraft, and thus, f_{ce} can be directly determined and is indicated on the spectrogram.

From 8:45 to 9:15 UT, there are fluctuations in the UHR band. These occur in the outer plasmasphere and indicate the presence of density irregularities. It is important to capture density irregularities in any density extraction algorithm. In the example in Figure 1, the UHR band is fairly clear. At other times, the spectrum might be contaminated with a variety of plasma emissions [e.g., see *Kurth et al., 2015, Figures 10 and 7*]. The main challenge of developing a robust automated algorithm is to separate the upper hybrid line from different types of contaminating emissions. Often, the UHR band in the plasmasphere is fairly clear, but there is more noise in the plasma trough. Specifically in the trough, it can be difficult to separate the UHR band from “ $(n + 1/2) f_{\text{ce}}$ ” emissions often observed between the harmonics of f_{ce} (but not necessarily midway between them as shown in *LaBelle et al. [1999]*). These emissions are also referred to as banded emissions [e.g., *LaBelle et al., 1999; Benson et al., 2001*]. *Benson et al. [2001]* argue that these emissions occur both above and below f_{uhr} and that emissions at f_{uhr} are not always observed with the banded emissions. In particular, in Figure 1 it is hard to make a reliable conclusion regarding f_{uhr} starting from 9:20 UT without a proper spectral interpretation.

In previous missions, the UHR band has been manually identified [e.g., *LeDocq et al., 1994*], and a number of semi-automated routines have been developed. Works on semi-automated routines development date back to the ISEE-1 era [e.g., *Trotignon et al., 1986*], and the research in this field is ongoing. More recent works include *Trotignon et al. [2010]*, *Denton et al. [2012]*, and *Goldstein et al. [2014]*. In *Trotignon et al. [2010]* determination

plasma frequency, f_{pe} , is greater than the electron cyclotron frequency, f_{ce} [e.g., *Beghin et al., 1989; Benson et al., 2004*]. The upper hybrid resonance (UHR) frequency, f_{uhr} , is a combination of the electron cyclotron frequency, f_{ce} , and the electron plasma frequency, f_{pe} :

$$f_{\text{uhr}} = \sqrt{f_{\text{ce}}^2 + f_{\text{pe}}^2}. \quad (1)$$

The electron cyclotron frequency is a function of the magnetic field strength, B , and the electron plasma frequency is a function of the electron number density, n_e . In the International System (SI) these frequencies are given as

$$f_{\text{ce}} = \frac{|q_e|B}{2\pi m_e}, f_{\text{pe}} = \frac{1}{2\pi} \sqrt{\frac{q_e^2 n_e}{m_e \epsilon_0}}, \quad (2)$$

where q_e is the charge of electron, m_e is the mass of an electron, and ϵ_0 is the

of the electron density was performed on the wave spectra from the Waves of High frequency and Sounder for Probing of Electron density by Relaxation (WHISPER) instrument on the four-satellite Cluster mission. The work of *Denton et al.* [2012] was based on the measurements of the electron density n_e found with the automated detection routine from passive radio wave observations by the Radio Plasma Imager (RPI) instrument on the IMAGE (Imager for Magnetopause-to-Aurora Global Exploration) satellite. In *Goldstein et al.* [2014], a comparison study between manual method of determining plasma density and semi-automated extraction algorithm described in *Kurth et al.* [2015] was presented.

Recently, *Kurth et al.* [2015] developed the Automated Upper hybrid Resonance detection Algorithm (AURA) for EMFISIS HFR data. At each time step in the HFR data, AURA starts with an initial estimate of f_{uhr} from spacecraft potential measurements from the Electric Fields and Waves (EFW) instrument [*Wygant et al.*, 2013] and then applies a restricted search on the HFR spectrum to detect a nearby spectral peak. It also relies on the assumption that each successive spectrum contains a spectral peak associated with f_{uhr} near the previously identified peak. Once AURA is run on the HFR data for an orbit, an operator inspects the resulting f_{uhr} profile and corrects intervals where the algorithm has deviated from the visually observed upper hybrid band. AURA helps speed up processing of the HFR data but still requires significant manual intervention (see further discussion in section 3.3).

We propose an alternative algorithm for automatic inference of f_{uhr} using neural networks. Neural networks are one of the most widely used statistical learning models and are well suited for a very broad class of nonlinear approximations and mappings. For this application, we use a feedforward neural network, which is a type of neural network that is very effective in solving nonlinear regression problems with a large number of inputs. The properties of a neural network are initially determined using a set of training data, that is, a data set for which both the inputs (in this case, HFR spectra and geophysical parameters) and the outputs (f_{uhr}) are known. Once the neural network has been trained, validated, and tested, it can be applied to a large data set for which the f_{uhr} is not known. In this case, we use a training set developed from the AURA algorithm (provided courtesy of W. Kurth and the EMFISIS team). This data set obeys the condition $f_{\text{pe}} > f_{\text{ce}}$ in 99.86%, which allows us to confidently use the upper hybrid resonance band to determine plasma density [e.g., *Beghin et al.*, 1989; *Benson et al.*, 2004]. The training data set consists of 1091 orbits, i.e., represents a significant set of example data. After training, we apply the neural network to a database of 2425 orbits and use the output to obtain a database of electron number density. The neural network model and the resulting density distribution are analyzed and compared to the results obtained in *Kurth et al.* [2015] and to the empirical model by *Sheeley et al.* [2001].

2. Background on Neural Networks

In this section, we provide a brief introduction to neural networks and give a high-level description of the three main stages of the neural network design process: training, validation, and testing. Our goal is to introduce basic notions related to the field that may be useful in understanding our specific application to plasma wave data presented in section 3, and we refer the reader to works cited herein for further details on neural networks.

2.1. Brief Introduction to Neural Networks

Artificial neural networks are an attempt to model the information-processing abilities of biological neural networks in the brain [e.g., *McCulloch and Pitts*, 1943; *Hebb*, 1949; *Marr and Poggio*, 1976]. Artificial neural networks are a family of mathematical models that are quite effective at solving problems such as the approximation of functions, classification, pattern recognition, and clustering. Neural networks can be used to reconstruct a mapping between an input set of data X and corresponding outputs Y (assuming that this mapping exists) using a subset of data for which the outputs are known, and this subset is usually referred to as a training set.

An artificial neuron is a very simple computational unit and is the basic component of any artificial neural network. Similar to a real neural cell, an artificial neuron has input channels, a cell body (in which some simple computations happen), and an output channel. Simply put, a neuron receives some input data and then computes an output. Figure 2 shows the structure of an artificial neuron with N inputs. Each input channel has an associated weight, which can be considered as its “importance” (the larger the weight the more the influence of the corresponding input on the output). A neuron also has a bias. Formally, it can be considered

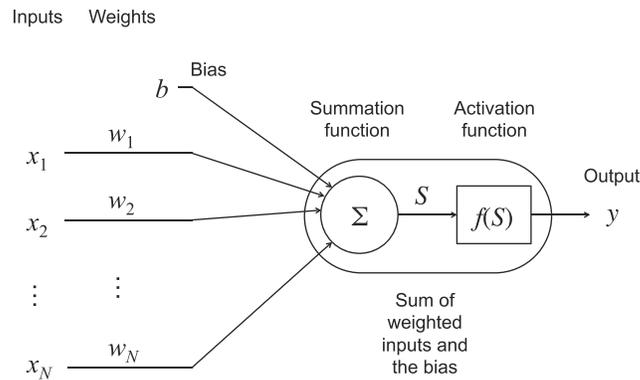


Figure 2. An artificial neuron is the building block of a neural network. The neuron takes N input variables, x_n , multiplies them each with a weight, w_n , and sums them along with a bias, b . The output of the summation, S , is applied to an activation function, $f(S)$, which becomes the output of the neuron.

as an internal parameter of a neuron, which affects the neuron’s output. For simplicity, bias is usually represented as an additional input to the neuron x_0 that is always equal to 1 with the weight equal to the value of the bias $w_0 = b$. A neuron integrates incoming information by adding the sum of input signals multiplied by their weights. A neuron has a fixed activation function, which acts on the calculated sum. The resulting value of the function is the output of the neuron. The activation function can be arbitrary and defines the type of a neuron. The most commonly used activation functions are sigmoid, binary (step function), and linear [e.g., *Sibi et al., 2013*].

Using one neuron, a limited range of simple problems can be solved, such as classification of linearly separable pattern sets [Auer et al., 2008]. More complex problems (e.g., classification of linearly nonseparable pattern sets) cannot be solved with just one neuron [Minsky and Papert, 1969]. Fortunately, they can be solved using neural networks, which consist of a number of neurons. An output of one neuron can serve as an input to the other neuron. This is a main idea behind artificial neural networks.

There are many different topologies of neural networks (ways to connect neurons into network). In this work, we use feedforward neural networks (FNNs). A FNN is the most basic and widely used artificial neural network and is well suited for a very broad class of nonlinear approximations and mappings. FNNs have achieved success in a number of domains [e.g., Salakhutdinov and Hinton, 2009; Glorot et al., 2011; Krizhevsky and Hinton, 2011; Mohamed et al., 2012]. Its topology is shown in Figure 3. It consists of a number of artificial neurons that are arranged into a layered configuration. There are three types of layers: input, output, and hidden layers. The input layer consists of the inputs to the network. No computations happen in this layer. Then follow any number of hidden layers consisting of any number of neurons placed in parallel. Outputs of one hidden

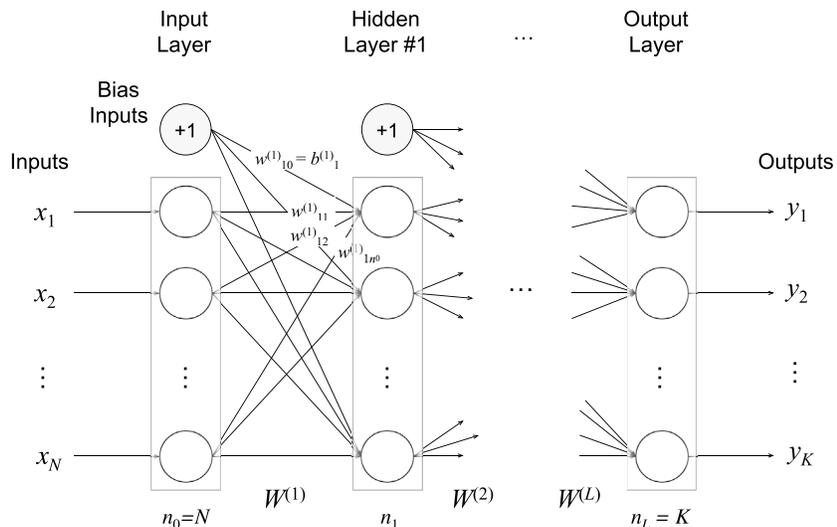


Figure 3. A feed forward neural network is one of the most commonly used neural network architectures. It consists of a number of artificial neurons arranged into layers. There are three types of layers: input, output, and hidden. Each layer has a common activation function. Outputs from one layer serve as inputs to the subsequent one as shown in the diagram. All connections from neurons of one layer to neurons of the other layer have their own weights. Layer i can be parameterized with the matrix of weights $W^{(i)}$. The weights are usually adjusted using training data for which both the input and output are known in order to minimize a given cost function.

layer serve as inputs to the subsequent layer. There might be several hidden layers in the network. The output layer (network output) is formed by another weighted summation of the outputs of the neurons in the last hidden layer [Lippman, 1987]. Neurons of one layer have a common activation function. Different layers may have different activation functions. Mathematically, feedforward neural networks can be defined as a composition/superposition of L activation functions f_1, \dots, f_L for a network with L hidden layers.

The description of FNNs given above is based on the assumption that the weights and biases are known. In practice, however, the weights and biases are unknown, and we need to determine them using a set of training data, that is, a data set for which both the input and output are known. Determining the weights and biases reduces to an optimization problem to minimize a given cost function. The cost function is generally defined based on the type of problem we are solving. The optimization problem can be solved with a variety of backpropagation techniques [Rumelhart et al., 1986]. In this work, we use a scaled conjugate gradient backpropagation algorithm (SCG) [Moller, 1993].

SCG performs well over a wide variety of problems, particularly for neural networks with a large number of weights. The SCG algorithm is faster for large networks than any of the other algorithms presented in the Matlab Neural Network Toolbox, and it has relatively modest memory requirements [Mathworks.com, 2015]. SCG is based upon a class of optimization techniques called conjugate gradient methods (CG). CG is the most popular iterative method for solving large systems of linear equations of the form $A\mathbf{x} = \mathbf{b}$, where \mathbf{x} is an unknown vector, \mathbf{b} is a known vector, and A is a known, square, symmetric, positive-definite matrix. In the basic backpropagation algorithm, the weights (of a neural network) are adjusted in the steepest descent direction (negative of the gradient). This is the direction in which the error function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms, a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. Another advantage of SCG is that it avoids the line search (performed in other CG algorithms to determine the optimal distance to move along the current search direction) by using a step size scaling mechanism. A detailed explanation of the algorithm can be found in Moller [1993].

2.2. Overview of the Neural Network Design Process

Next, we provide a high-level description of the main stages of the neural network design process: training, validation, and testing, as shown in Figure 4. Details on our specific implementation can be found in section 3.2. It is important to mention that our reference to training, validation, and testing as stages in neural network design workflow is solely for the purpose of organizing ideas and may not be considered as a part of a common vocabulary in this field.

As previously mentioned, a set of training data for which the output is known is required to design the neural network. First, the training set must be verified and preprocessed. For example, we need to make sure first that all the data points are consistent and there are no missing values in the training set. After that we split the data into three parts in a user-predefined ratio as the (1) training data set, (2) validation data set, and (3) testing data set.

Normalization is a standard preprocessing procedure that is done before training the neural network. All the variables are normalized to a range from 0 to 1, that is, $x_{\text{new}} = \left(\frac{x_{\text{old}} - \min_1}{\max_1 - \min_1} \right) (\max_2 - \min_2) + \min_2$, where x_{old} is the original value, x_{new} is the new value, \min_1 and \max_1 are the minimum and maximum of the original data range, and \min_2 and \max_2 are the minimum and maximum of the new data range (here 0 and 1). One reason to normalize input variables is to eliminate possible skewness in data that could happen due to different input units that could make the range of some variables much larger than others. A large value input can dominate the input effect and influence the model accuracy of the neural network system [Li et al., 2000]. The normalization is applied to the training data set independently of validation and test data sets, and the normalization parameters from the training set are saved and applied to the validation and test sets (rather than recalculated on these sets).

In the *training stage*, we first select the parameters of the neural network such as number of hidden layers, the number of neurons in each layer, and the activation function to be used. Parameter selection is an empirical process. An optimization algorithm is run to determine the weights for the neural network by minimizing the error between the output of the neural network and the known output of the training data set. Typically, we

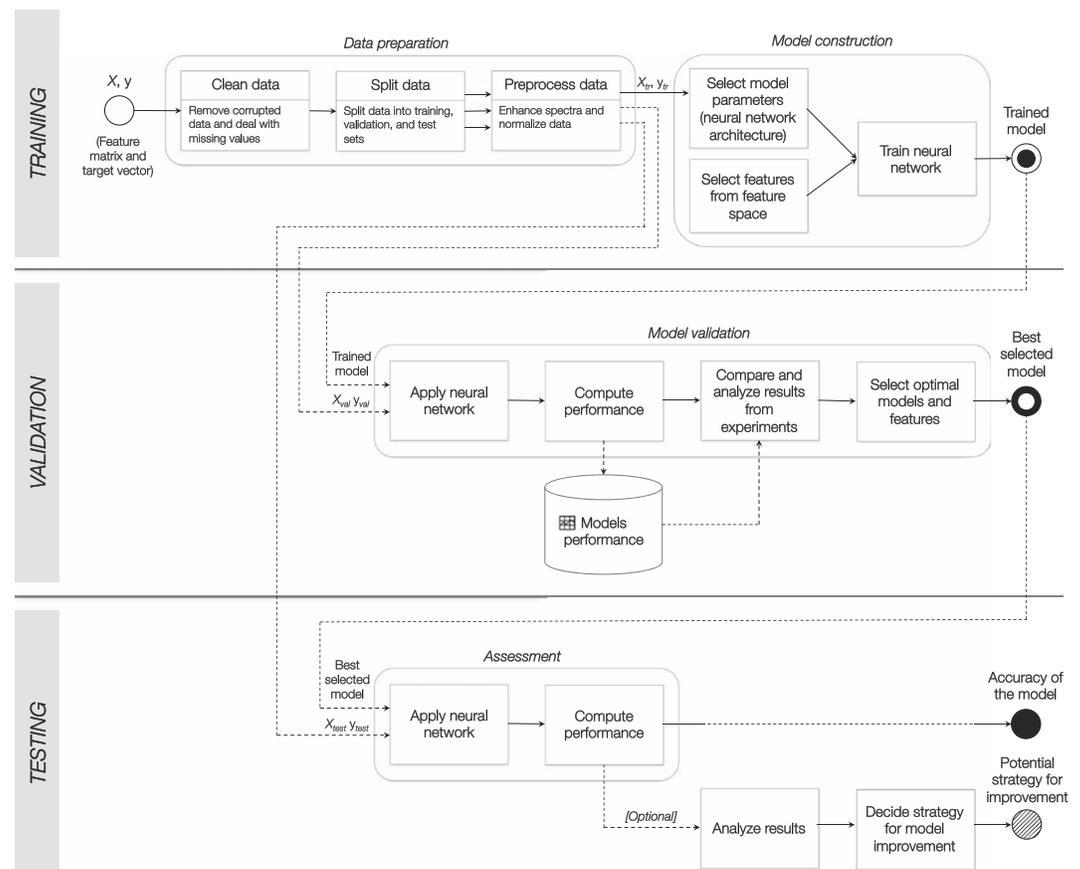


Figure 4. Neural network design workflow (adapted from Klavdianos *et al.* [2013]). Three main stages of the workflow are training, validation, and testing. First, the data set is verified and preprocessed. Then, the parameters of neural network are selected and the optimization algorithm is run to determine the weights for the particular neural network by minimizing the error between the output of the neural network and the known output of the training data set. Usually, we train several neural networks with different parameters, and in the validation stage, we select the one that yields the smallest error on the validation set. Finally, we calculate the accuracy of the model selected in the validation stage by comparing the model output to the known output of the test set.

train multiple neural networks with different parameters, and in the validation stage, we select the one that yields the smallest error on the validation set.

The goal of the *validation stage* is to ensure that a neural network is capable of reconstructing the relationship between the input and output variables on data it has not seen yet, that is, data that was not used to train the model. This is referred to as the generalization ability of the neural network. Using the validation data set, we apply it to each neural network constructed in the training stage and select the model with the highest accuracy by comparing the output of the model to the known output of the validation set.

Finally, in the *testing stage*, we calculate the accuracy of the model selected in the validation stage by comparing the model output to the known output of the test set. The calculated error is then claimed to be the accuracy of the whole system. Now, the neural network is ready to be applied to input data for which the output is not known.

3. NURD

The Neural-network-based Upper hybrid Resonance Determination (NURD) algorithm developed here consists of the steps shown in Figure 5. These steps are described in detail in the following subsections.

3.1. Input Data

We use data from the EMFISIS instrumentation suite [Kletzing *et al.*, 2013] on the Van Allen Probes to train a neural network for plasma density inference. In order to train a neural network (hereinafter referred to as

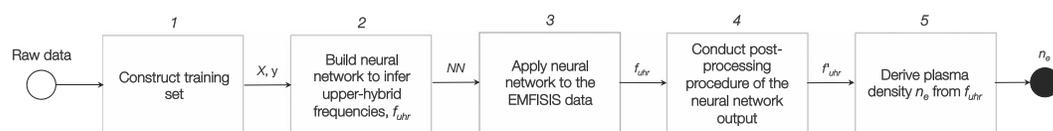


Figure 5. Diagram of the proposed algorithm.

the model), a user must have a set of example data, which consists of the input variables and known target variables (outputs of the model). Data used as inputs to the model are described below and summarized in Table 1. Our neural network model has one output, which is also described here. Construction of the training set in a form suitable for training is discussed in the next subsection.

The first input to the model is the EMFISIS HFR survey mode electric field power spectral density covering the frequency range of 10 to 487 kHz. The data are binned into 82 logarithmically spaced bins and has a 6s temporal resolution. The frequency spacing provides about 5% spectral resolution ($\Delta f/f$) and determines the resulting density resolution ($\Delta n/n$). As previously discussed, n_e is proportional to f_{pe}^2 , which results in 10% density resolution. Also, the upper frequency range (487 kHz) limits the maximum density that can be derived to about 2900 cm^{-3} , and the lower frequency range (10 kHz) limits the minimum density to about 1 cm^{-3} . The logarithm of the electric field power spectral density for the 82 frequency bins is used as an input to the model.

We also use a measurement of background magnetic field, $|B|$, from the EMFISIS fluxgate magnetometer to determine electron cyclotron frequency, f_{ce} ($f_{ce} = 28|B|$, where f_{ce} is measured in hertz and $|B|$ is in nanotesla). The logarithm of the cyclotron frequency is used as an input to the model.

Plasma density is known to vary spatially, so we use two input parameters related to the location of the measurement. The first is the dipole L value, which identifies the equatorial crossing of the magnetic field line where the measurement was made in units of Earth radii. We also use magnetic local time (MLT) in decimal hours.

The Kp index is an indicator of global geomagnetic disturbances and is used as an input to the model since the plasma density is known to vary as a function of geomagnetic conditions.

Finally, after visual inspection of the spectrograms, it was noticed that a frequency bin with the highest power spectral density from the HFR spectrum could be used as a rough initial approximation of f_{uhr} ; thus, it was also incorporated into the training set. This assumption might introduce errors to the neural network, since as mentioned in section 1, the upper hybrid frequency f_{uhr} is generally the upper frequency edge of the upper hybrid band rather than its maximum emission peak [Benson *et al.*, 2004]. However, the conclusion in Benson *et al.* [2004] was based on active and passive observations from the Radio Plasma Imager (RPI) on the IMAGE satellite. This instrument has a higher frequency resolution compared to EMFISIS HFR on board the Van Allen Probes, which allowed to make extremely accurate determinations. In the case of EMFISIS HFR, the coarse frequency resolution of the instrument does not allow us to capture that edge accurately. We will discuss the error of density determination in section 3.3.

As mentioned before, inherently, to use a neural network we need to have known responses to the inputs of the model in order to train it. In our case, the output or target variable for the neural network model is the upper hybrid frequency value for each set of input variables. Identifying these values manually for a substantial set of data is a challenging and an extremely time-consuming task, which also might be potentially erroneous. For this reason, for the training data set (including validation and testing), values of the upper

Table 1. List of Input Variables to the Neural Network Model

	Name	Description	Units
1 – 82	\log_{10} spectrum	Decimal logarithm of the spectrum	$\log_{10} \text{ V}^2/\text{m}^2/\text{Hz}$
83	$\log_{10} f_{ce}$	Decimal logarithm of electron cyclotron frequency	$\log_{10} \text{ Hz}$
84	L	Magnetic field line	Earth radii
85	Kp index	Geomagnetic index	Unitless (0–9)
86	MLT	Magnetic local time	Hours (0–24)
87	$f_{bin_{max}}$	Frequency bin with the highest power spectral density from the HFR spectrum	Unitless (1, 2, ..., 82)

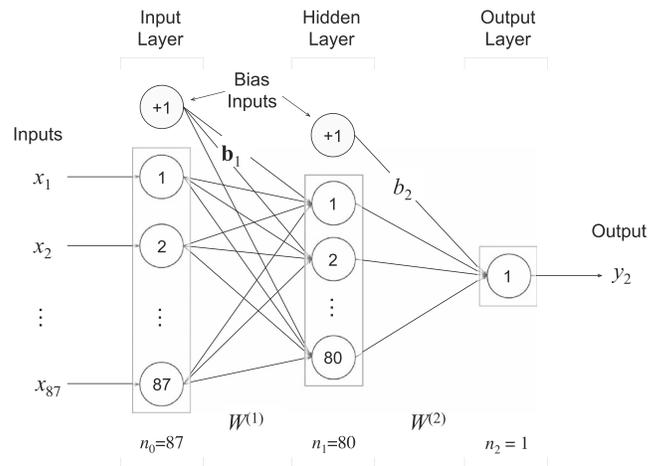


Figure 6. Architecture of the feedforward neural network selected in the validation stage. It has 87 inputs, 80 neurons in the hidden layer, and one output. The weights of the neural network can be represented as $80 \times 87 W^{(1)}$ and $1 \times 80 W^{(2)}$ matrices, and the biases can be represented as 80×1 vector \mathbf{b}_1 for the hidden layer and scalar b_2 for the output layer.

hybrid frequency for 1091 orbits was obtained from the AURA algorithm [Kurth et al., 2015] and made available to us through the courtesy of W. Kurth and the EMFISIS team. This data set had been manually inspected, and the data points, for which the AURA algorithm failed to identify the upper hybrid frequency correctly, had been corrected by the operator [Kurth et al., 2015]. The ultimate accuracy of the resulting electron density obtained with AURA is $\approx 10\%$, which bounds the accuracy of the resulting neural network model. With the 9h orbital period and the 6s cadence of the HFR measurements, this data set yields about 5,900,000 measurements.

Not all of these measurements can be used to train the neural network, since the UHR frequency may be above the upper limit of the HFR frequency range particularly near perigee. We exclude the out of frequency range measurements when the network is trained and use a special procedure when the resulting neural network is being applied to the data. The procedure is as follows. Each orbit is considered separately. First, we determine the left and right boundaries of the upper hybrid resonance line by finding the bins with maximum values of the spectral density in the highest frequency spectral bin at 480 kHz nearest perigee. The neural network is then applied only to the measurements inside these boundaries.

After excluding the out of frequency range portion of the data, the data set for training, validation, and testing comprises 4,027,610 measurements. This results in a $1 \times 4,027,610$ vector of target values (measurements of f_{uhr}) and an $87 \times 4,027,610$ matrix of input values from Table 1 (in machine learning, the matrix of input values is usually referred to as feature matrix).

3.2. Neural Network Implementation

The Matlab Neural Network Toolbox is used for all operations on neural networks. We use a feedforward neural network for UHR frequencies inference, which is a type of artificial neural network that is traditionally used for solving multiple nonlinear regression problems and is more effective on large input parameters space than regular regression models.

The architecture of the feedforward neural network used is shown in Figure 6. It consists of an input layer, one hidden layer, and an output layer. Each layer consists of neurons. An artificial neuron consists of inputs, which are multiplied by weights, summed, and then computed by a mathematical function, which determines the output of the neuron (more background on neural networks was given in section 2). The number of neurons in the input layer should be equal to the number of dimensions in the input vector (87 as was discussed in subsection 3.1). In this application, the validation stage has shown that 80 neurons in the hidden layer are well suited for reasonable performance of the neural network (details about the validation stage are given below). The output layer consists of just one neuron, as we are to infer only one variable, f_{uhr} .

Each neuron in one layer is connected to the neurons of the subsequent layer. Each connection has its own weight, and each neuron has its own bias. For convenience, these weights can be represented as $80 \times 87 W^{(1)}$ and $1 \times 80 W^{(2)}$ matrices, and the biases can be represented as 80×1 vector \mathbf{b}_1 for the hidden layer and scalar b_2 for the output layer.

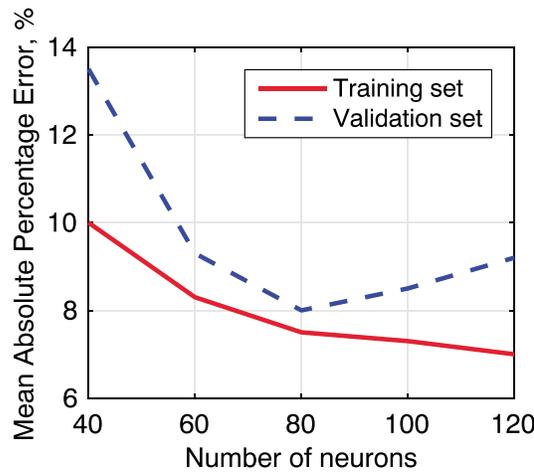


Figure 7. This figure shows the mean absolute percentage error (MAPE, on the x axis) of the neural networks with different number of neurons in the hidden layer (on the y axis) being applied on the training data (red solid curve) and on the validation data, i.e., data not used to train the network (blue dashed curve). While MAPE usually decreases on the training set, when the complexity of the network (number of neurons) increases, it stops decreasing after a certain point on the validation data, which indicates that the network has been overfit. That way, we can select the optimal number of neurons in the hidden layer.

Next, we use the validation data set to identify the most suitable neural network architecture. The error measure used to evaluate the performance of neural networks is the mean absolute percentage error (MAPE) [Armstrong, 1985] given by

$$MAPE = \frac{1}{M} \sum_{i=1}^M \frac{|f_i^{AURA} - f_i^{NURD}|}{f_i^{AURA}} \times 100\%, \quad (5)$$

where M is the size of the validation set, f_i^{NURD} are f_{uhr} values predicted using our neural networks, and f_i^{AURA} are f_{uhr} values provided by the EMFISIS team (they are referred to as ground truth values). Figure 7 shows the plot of the MAPE as a function of the number of neurons in the hidden layer. The red solid curve shows the MAPE when the neural networks are applied on the same data used to train the network. As the number of neurons in the hidden layer increases, the MAPE decreases; that is, when more free parameters are available, the model is better able to fit the data. On the other hand, when the neural networks are applied on a validation data

The transfer function for the hidden layer is the hyperbolic tangent sigmoid function (transforms any input to the range of -1 to $+1$) and linear transfer function for the output layer. The output of the hidden layer is a vector

$$\mathbf{a}_1 = \tanh(\mathbf{b}_1 + W^{(1)}\mathbf{x}), \quad (3)$$

where \mathbf{x} is the 87×1 input vector characterizing f_{uhr} . The output of the network is then

$$a_2 = b_2 + W^{(2)}\mathbf{a}_1. \quad (4)$$

The training set consists of 4,027,610 input vectors, and this set is randomly split into three parts: training, validation, and test sets in the following proportion, respectively, 34% : 33% : 33%. A large fraction of the test data, 66%, was reserved for validation and testing since there is a large amount of training data available. Such split ratio allows for better generalization ability of the network.

Using the training set, we trained five different neural networks varying the number of neurons in the hidden layer from 40 to 120. As discussed in section 2.1, we train the network using a conjugate gradient (SCG) backpropagation algorithm.

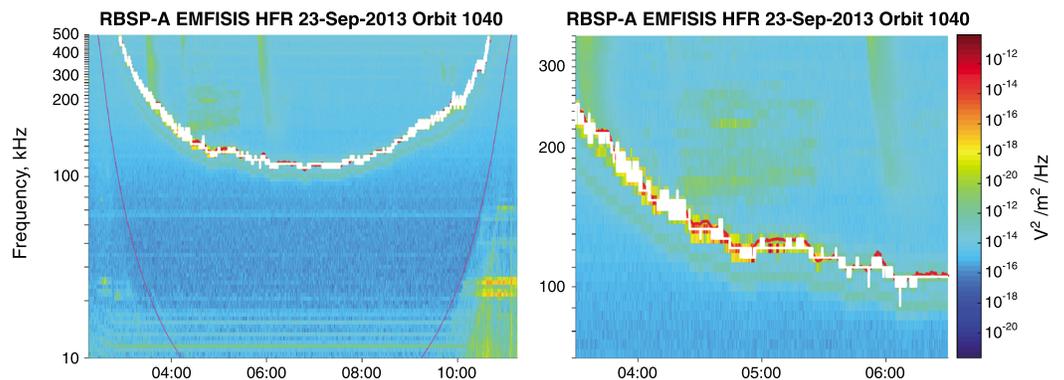


Figure 8. An example of the output of the model on the spectrogram of Waves HFR data before (red curve) and after postprocessing step (white curve). (left) The spectrogram for orbit 1040 and (right) the zoom-in version of it from 3:30 until 6:30 UT. Postprocessing removes the possible noise in the densities inferred with the neural network.

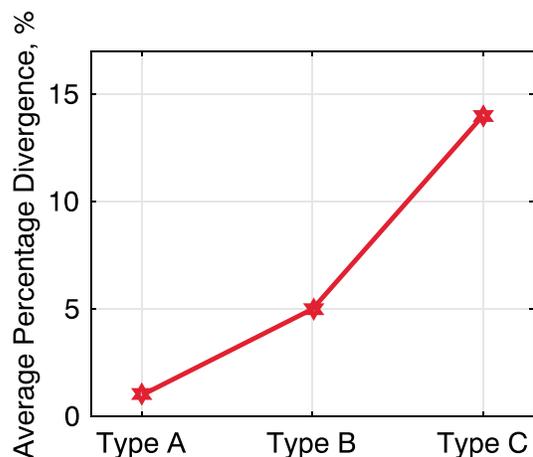


Figure 9. Average percentage divergence of the electron density n_e inferred with the proposed algorithm from densities obtained with AURA versus types.

set (that is, data that were not used to train the network), then we find that as the number of neurons in the hidden layer increases the error on the validation set decreases until a certain point. After this point the error on the validation set starts to increase, which means that the neural network has been overfit. When a model is overfit, it does not generalize well; that is, there is low error on the training data but high error when new data are applied to the model. Therefore, we select the neural network for which the MAPE is the smallest on the validation set. As can be seen in Figure 7, the selected model has 80 neurons in the hidden layer.

Next, we use the test data set and apply it to the model selected in the validation set. Eventually, we find that the MAPE on the test set is ~8%, and this is considered the overall error of our model.

3.3. Applying the Neural Network

The neural network constructed in the previous section with 80 neurons in the hidden layer was applied to a database of 2425 orbits from October 2012 to March 2015. After the neural network is applied to the input data, a postprocessing step is applied (step 4 in the diagram in Figure 5) in order to reduce the noise in the prediction of f_{uhr} (the output of the neural network is of the form $\mathbf{y} = f(W, \mathbf{x}) + \epsilon$, where ϵ is random noise). The predicted f_{uhr} value is rounded to the closest EMFISIS HFR frequency bin. Then we consider this bin and two adjacent bins (above and below) in the HFR spectrum. The resulting f_{uhr} is determined as the frequency corresponding to the bin with the highest power spectral density of these three bins. An example of the “raw” output of neural network and the postprocessed one is presented in Figure 8. It can be seen from this figure that the neural network diagnosis (shown in red) might slightly fluctuate from the actual upper hybrid line. Postprocessing step reduces possible fluctuations. However, the resulting upper hybrid line can still switch between several adjacent spectral bins in the spectrum (e.g., see Figure 10), which might introduce a certain level of uncertainty in the density determination. Also, we need to take into account errors introduced by using the frequency of the maximum emission intensity as the resulting f_{uhr} instead of the upper cutoff of the upper hybrid resonance band [Beghin et al., 1989; Benson et al., 2004]. As was already mentioned, the coarser frequency resolution of the EMFISIS HFR compared to the IMAGE RPI might not allow us to capture that edge accurately. Thus, we can evaluate the error introduced due to that by estimating the error in density determination when the derived value of f_{uhr} is off by one frequency bin from its true value. After some simple calculations, we obtain the following formula for this error:

$$\frac{\Delta n}{n} \approx 2 \frac{\Delta f}{f} \sqrt{\left(\frac{f_{\text{pe}}}{f_{\text{ce}}}\right)^{-2} + 1}, \tag{6}$$

where, $\frac{\Delta f}{f}$ is the frequency resolution (in case of 1 bin), which is $\approx 5\%$ as discussed before, and ratio $\left(\frac{f_{\text{pe}}}{f_{\text{ce}}}\right)$ is always greater than 1, which makes the expression under the square root always less than 2 but more than 1, thus making the whole expression always more than $\approx 10\%$ and less than $\approx 14\%$. Hence, the error which might be introduced due to toggling between bins (two bins in this case) can vary from about 10 to 14%. It is worth noting that this kind of error should be accounted mostly in the trough region since for the plasmasphere determination of the upper hybrid line is quite straightforward most of the time.

Finally, the electron densities are derived from the resulting f_{uhr} using formulae (1) and (2).

In order to evaluate the performance of our NURD algorithm in comparison with densities provided by the EMFISIS team, we must first introduce a classification of orbits into types. In Kurth et al. [2015], the orbits are divided into three types: A, B, and C:

1. Type A consists of orbits where less than 25% of the f_{uhr} points required manual correction and constitutes 70% of the orbits processed by AURA.

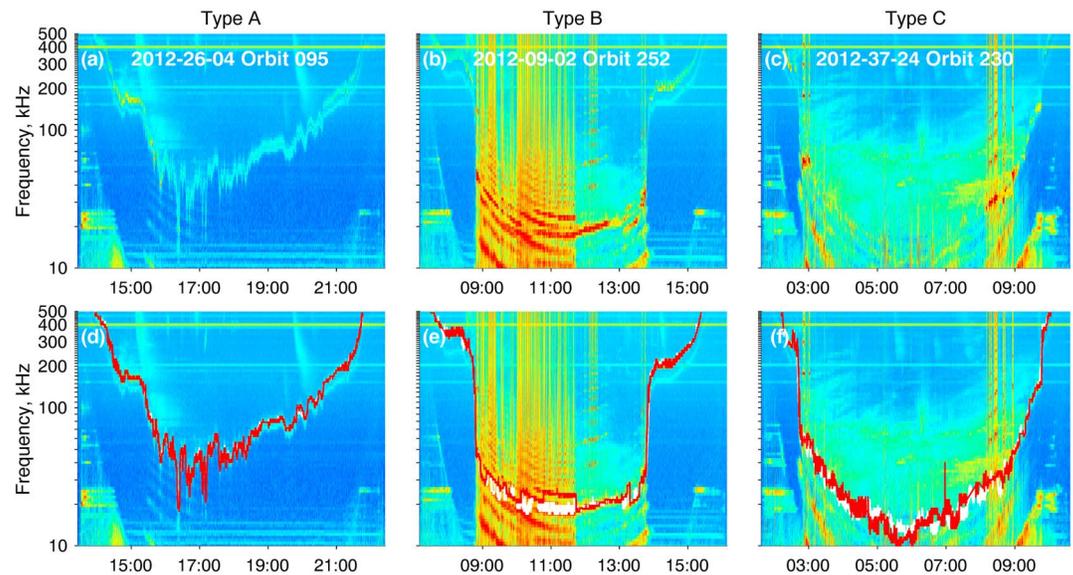


Figure 10. (a–c) Examples of typical orbits of each type and (d–f) resulting upper hybrid frequency f_{uhr} profiles determined with the NURD algorithm (white) and AURA (red).

2. Type B are orbits where 25 to 50% of the data points require manual correction, constituting 20% of the orbits.
3. Type C are orbits where there are “interpretational difficulties in finding the upper hybrid band” [Kurth *et al.*, 2015], and this type constitutes 10% of the orbits.

To compare results from AURA and our NURD algorithm, we compute the mean absolute percentage error for each orbit type. This evaluation simply shows to what extent the densities inferred with the NURD method differ from the densities identified with AURA for each type. It does not actually demonstrate the real error rate of either method since the ground truth densities are ambiguous, particularly for type C. Thus, it is more appropriate to call this evaluation measure as average divergence. It is also worth keeping in mind that while the data set of electron densities obtained with AURA and used in training is the most reliable resource of the electron plasma density for the Van Allen Probes currently, there are still cases of a great uncertainty in density determination which cannot be resolved even by the manual inspection and can lead to larger errors, especially in the plasma trough where the densities are low due to recent geomagnetic activity (please refer to section 2.2 of Kurth *et al.* [2015] for discussion about potential errors introduced due to the interpretational issues). Thus, it is quite likely that there are errors in the training set, which might affect the neural network performance.

Figure 9 shows the average divergence for different types of orbits. One can see from this plot that AURA and NURD yield nearly identical results for orbits of type A since this type corresponds to orbits with f_{uhr} profiles that are the easiest to identify. The difference for type B is also not very significant (about 5%), but the difference for type C is about 14%. This is due to the uncertainty of the upper hybrid frequency determination particularly during the geomagnetically active times when the plasma density in the trough is very low and there are strong electron cyclotron harmonic emissions present. Figure 10 demonstrates typical examples of orbits of each type and shows the f_{uhr} profiles obtained with the aid of AURA (used in the training, marked with red) and the resulting f_{uhr} profiles obtained with the aid of the NURD algorithm (indicated in white). As we can see from this plot, the upper hybrid lines derived for type A orbit with both algorithms almost overlap. For more complicated types of orbits B and C, the resulting upper hybrid line inferred with the NURD algorithm almost overlaps with the AURA results in the plasmasphere regions (small L shells) but differs in the trough (high L shells). Indeed, in the plasmasphere the density determination is quite straightforward. Differences in the trough region make the largest contribution to the average divergence for these complicated types of orbits. For these cases, it can be hard to quantify which of the algorithms performed better. In this particular example of orbit of type B, we can see that NURD's f_{uhr} tends to toggle more between adjacent bins than AURA's. But NURD's f_{uhr} tends to follow one resonance band, which seems to be selected among the intense banded emissions as the one that is a continuation of the intense emission extending beyond the

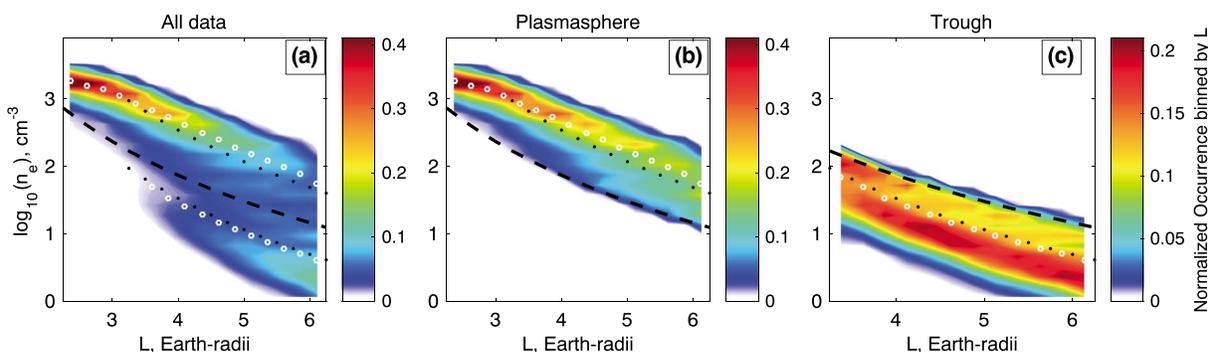


Figure 11. The log of the electron density (n_e) versus L , where the color scale indicates the occurrence normalized by the number of measurements in different L bins. The black dashed line shows the separation between trough-like and plasmaspheric-like data as $n_b = 10(6.6/L)^{-4}$ (as in *Sheeley et al. [2001]*). The upper black dotted line is the plasmasphere model, and the lower black dotted line is the trough model of *Sheeley et al. [2001]* for MLT = 0. The upper and lower white dotted lines show the mean of the log of the electron density (n_e) obtained with the NURD algorithm for plasmasphere and trough correspondingly.

intense banded emissions, while AURA's f_{uhr} might switch from band to band. Again, it is extremely hard to judge visually which of these diagnoses is correct, and in some cases it might happen that both algorithms could be wrong. Most of these cases usually refer to the regions of type C orbits, when the plasma density in the trough is very low and there are strong electron cyclotron harmonic emissions present. For example, in the vicinity of 07:30 in Figures 10c and 10f, there is a possibility that both the NURD and AURA algorithms could be wrong and that the f_{uhr} band could belong to the moderately strong emission near 40 kHz. However, it is difficult to make such conclusion without a proper spectral interpretation of the emissions. *Benson et al. [2001]* show in an analysis of AMPTE/IRM (Active Magnetospheric ParticleTracer Explorers/Ion Release Module) banded magnetospheric emissions that emissions at f_{uhr} are not always observed with the banded emissions. *Benson et al. [2001]* also show that under low-density conditions different passive and active techniques of density determination could yield significantly different results (up to a factor of 3.5). Besides that, *Benson et al. [2001]* discuss cases when different magnetospheric emissions in passive spectra have been misinterpreted leading to errors in density determination. These findings illustrate the difficulties affecting the reliability of the density determination. At the current stage, both NURD and AURA may experience difficulties in the low-density regions. Thus, while use of the densities marked "questionable" in the resulting electron density data set obtained with NURD is safe in statistical studies, one should be careful when using the exact diagnoses.

Using the new statistical data set, we are planning to develop a neural network that will predict global plasma density based only on solar wind conditions and geomagnetic indexes. Comparisons of model results with observations will show how well the plasma density can be predicted by such a model and how much additional information that can help current NURD predictions is contained in geomagnetic indices or solar wind parameters.

4. Results and Discussion

After applying the NURD algorithm, we produced a database of 10,252,612 measurements of electron density from October 2012 to March 2015. Here we perform initial analysis of this database and compare our distributions to the empirical plasmasphere and trough density models of *Sheeley et al. [2001]*. In our analysis, we separate density values into plasmaspheric-like data and trough-like data using the same criteria as in *Sheeley et al. [2001]*. A threshold density is defined as

$$n_b = 10(6.6/L)^{-4}. \tag{7}$$

Density readings at or above n_b (given also in cm^{-3}) for the given L shell are considered plasmaspheric-like. Readings below n_b are considered trough-like.

In Figure 11, we show two-dimensional plots of normalized occurrence as a function of electron density and L . We divided the data set into 26 bins in the logarithm of density and 16 bins in L . Each bin contains the number of measurements in that range of L and n_e divided by the total number of measurements in that range

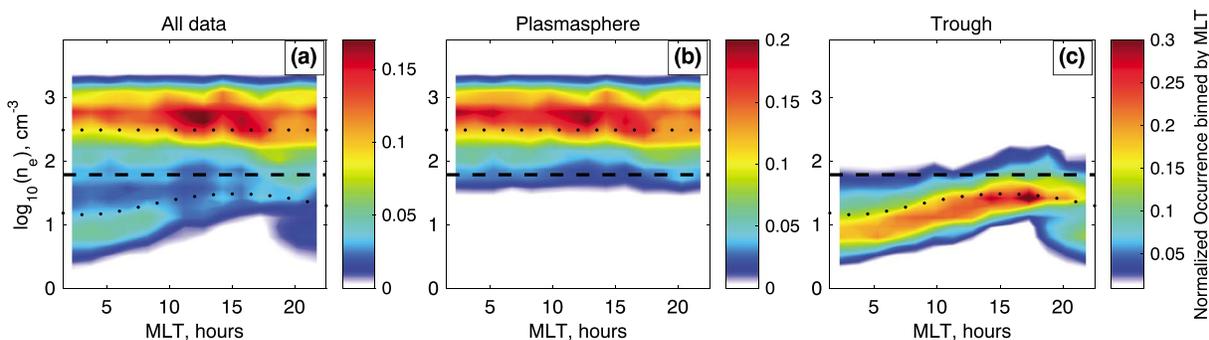


Figure 12. The log of the electron density (n_e) versus MLT, where the color scale indicates the occurrence normalized by the number of measurements in different MLT bins for the data in the range of $3 < L < 5$. The black dashed line is the density threshold for the median of L for both plasmaspheric-like and trough-like data, which is $L \approx 4.2$. The upper black dotted line is the plasmasphere model at the median of L for plasmasphere-like data ($L \approx 4.1$), and the lower black dotted line is the trough model of *Sheeley et al.* [2001] at the median of L for trough-like data ($L \approx 4.6$).

of L . Thus, the color scale indicates the normalized occurrence in different L bins. The minimum number of measurements required to calculate the occurrence is 10, and the median number of measurements in each bin in Figure 11 is 19,063. A contour function is applied to plot the binned occurrence.

Figure 11a shows the normalized occurrence for all measurements. The black dashed line is the threshold density, n_b . The upper black dotted line is the plasmasphere model of *Sheeley et al.* [2001], and the lower black dotted line is the trough model of *Sheeley et al.* [2001] for $MLT = 0$. The upper and lower white dotted lines are the mean of the log of the electron density derived with the NURD algorithm for plasmasphere and trough correspondingly. The intense red color in the range of $L = 2$ to 3.5 indicates that at these L values, the measurements tend to be clustered around a fairly narrow range of densities. However, at higher L , we begin to see a bimodal structure with clear separation between plasmasphere and trough measurements.

In Figure 11b, we examine just the plasmasphere-like measurements. We find generally good agreement with the Sheeley plasmasphere model, but we also find that occurrence distributions measured by Van Allen Probes tend to peak at slightly higher density than the average value obtained with CRRES data. The shift in the mean density value is $\approx 280 \text{ cm}^{-3}$ (≈ 0.14 on a logarithmic scale) in average. In contrast, in the trough illustrated in Figure 11c, the occurrence distributions peak at densities slightly lower than the empirical model. The peak of the occurrence is shifted $\approx 5.32 \text{ cm}^{-3}$ down in average (≈ 0.05 on a log scale).

Next, in Figure 12 we examine the normalized occurrence in different MLT bins. Here we limit the data to the range of $3 < L < 5$ and divide the data into 26 bins in the logarithm of density and 14 bins in MLT. As previously, each bin contains the number of measurements in that bin divided by the total number of measurements at that MLT, such that the color scale indicates the normalized occurrence in different MLT bins. Figure 12 shows (a) all data, (b) plasmasphere-like data, and (c) trough-like data. The black dashed line is the density threshold for the median of L for both plasmasphere-like and trough-like data, which is $L \approx 4.2$. The upper dotted line is the plasmasphere model of *Sheeley et al.* [2001] at the median of L for plasmasphere-like data ($L \approx 4.1$), and the lower black dotted line is the trough model of *Sheeley et al.* [2001] at the median of L for trough-like data ($L \approx 4.6$).

The relatively good agreement between the density measurements obtained with the NURD algorithm and the empirical model of *Sheeley et al.* [2001] can be seen from both of these figures. Since the neural network in the NURD algorithm has been trained on the data set obtained with AURA, we need to mention that this relative agreement should also be accounted for the proficient quality of the AURA density set. For sure, special measures should be employed to obtain the optimal neural network, which would perform the best on this given data set (e.g., choosing the appropriate number of neurons discussed above). Thus, we might conclude that the agreement with the empirical model of *Sheeley et al.* [2001] was reached due to both AURA, which provided a proficient data set, and NURD, which was optimally trained to infer the density with a sufficient accuracy.

5. Summary

An algorithm for automated determination of plasma densities from electric field measurements made on Van Allen Probes spacecraft has been developed. The algorithm employs a feedforward neural network to infer upper hybrid frequency profiles from the HFR spectrograms and electric field measurements obtained from the EMFISIS instrumentation suite. Plasma densities are derived then from the inferred upper hybrid frequencies.

The algorithm is described in full detail, and analysis of the obtained densities is presented. The proposed NURD algorithm is applicable for L shells $2 \leq L \leq 6$ and covers all local times.

The algorithm was applied to the database available covering 2425 orbits. The analysis of the resulting densities has shown an agreement with the Sheeley model for trough and plasmaspheric-like densities and, on the other hand, there is a large variability in data. Using the proposed NURD algorithm, we are able to determine the density in a much finer resolution than using existing empirical models.

Comparison with AURA [Kurth *et al.*, 2015] showed that densities obtained with the proposed method are in a good agreement with the densities obtained with AURA. To illustrate this, we used the classification of orbits spectrograms into three levels of difficulty introduced in Kurth *et al.* [2015]: type A (upper hybrid frequency is easy to identify, 70% of all orbits), type B (some interpretation is needed to be done by an expert, 20% of all orbits), and type C (concealed signal, 10% of all orbits). The mean average percentage difference between densities by these two methods was $\sim 1\%$ for type A, $\sim 5\%$ for type B, and $\sim 14\%$ for type C. Comparison with the statistical model of Sheeley *et al.* [2001] also showed relatively good agreement with the densities set obtained with NURD. The ultimate error rate of the derived density is $\sim 14\%$. However, the error still might be larger in a number of cases when there is high uncertainty in the density determination, especially in the plasma trough where the densities might be low due to recent geomagnetic activity. Indeed, the NURD algorithm does not save us from the uncertainty in density determination. But first, it can still produce reasonable estimates for the regions of uncertainty, which can be safely used in statistical studies, and second, it is automated, which means that it can remove if not all then definitely a significant part of the manual aspects of the density determination.

Using the created framework, the neural network model can be retrained on other data sets for which input parameters can be modified if needed. At the moment NURD is tuned specifically for the Van Allen Probes data, but with it we were able to show that neural networks, and thus possibly other machine learning algorithms, might be applied and produce good results on such data.

In the future, we are planning to work on the development of a global predictive neural network that will be based on solar wind parameters and geomagnetic indices. If, using this new predictive system, we can determine density just from the solar wind and geomagnetic indices, that will be a good indication that certain dependencies between the plasma density and the solar wind and geomagnetic indices exist. Thus, using the developed system and passive observations, we could determine the plasma density values better than inferring them from the passive observations alone. The future extension of such a network will be a data assimilative predictive network that will use all available measurements and will blend them with the predictive model.

Acknowledgments

The work at UCLA is supported by NASA awards NNX10AK99G and NNX13AE34G, NSF award AGS-1243183, UC Lab Fee award 116720, and Horizon 2020 award 637302, the International Space Science Institute (ISSI). The work at Stanford University was supported under NASA awards NNX14AC04G and NNX15AI94G. The research at University of Iowa was supported by JHU/APL contract 921647 under NASA prime contract NAS5-01072. All Van Allen Probes data were accessed through the EMFISIS website hosted at the University of Iowa, and we graciously thank the EMFISIS Principal Investigator, Craig Kletzing, and the EMFISIS team. The K_p index was provided by the World Data Center. The electron density data set obtained in this paper is available from <ftp://rbm.epss.ucla.edu/ftpdisk1/NURD>.

References

- Armstrong, J. S. (1985), *Long-Range Forecasting: From Crystal Ball to Computer*, 2nd ed., John Wiley, New York.
- Artemyev, A. V., D. Mourenas, O. V. Agapitov, and V. V. Krasnoselskikh (2013), Parametric validations of analytical lifetime estimates for radiation belt electron diffusion by whistler waves, *Ann. Geophys.*, *31*, 599–624.
- Auer, P., H. Burgsteiner, and W. Maass (2008), A learning rule for very simple universal approximators consisting of a single layer of perceptrons, *Neural Networks*, *21*(5), 786–795, doi:10.1016/j.neunet.2007.12.036.
- Beghin, C., J. L. Rauch, and J. M. Bosqued (1989), Electrostatic plasma waves and HF auroral hiss generated at low altitude, *J. Geophys. Res.*, *94*, 1359–1379.
- Benson, R. F., V. A. Osheroovich, J. Fainberg, A. F. Viñas, and D. R. Ruppert (2001), An interpretation of banded magnetospheric radio emissions, *J. Geophys. Res.*, *106*, 13,179–13,190.
- Benson, R. F., P. A. Webb, J. L. Green, L. Garcia, and B. W. Reinisch (2004), Magnetospheric electron densities inferred from upper-hybrid band emissions, *Geophys. Res. Lett.*, *31*, L20803, doi:10.1029/2004GL020847.
- Carpenter, D. L., and R. R. Anderson (1992), An ISEE/whistler model of equatorial electron density in the magnetosphere, *J. Geophys. Res.*, *97*, 1097–1108.
- Carpenter, D. L., B. L. Giles, C. R. Chappell, P. M. E. Décréau, R. R. Anderson, A. M. Persoon, A. J. Smith, Y. Corcuff, and P. Canu (1993), Plasmasphere dynamics in the duskside bulge region: A new look at an old topic, *J. Geophys. Res.*, *98*(A11), 19,243–19,271, doi:10.1029/93JA00922.

- Denton, R. E., Y. Wang, P. A. Webb, P. M. Tengdin, J. Goldstein, J. A. Redfern, and B. W. Reinisch (2012), Magnetospheric electron density long-term (>1 day) refilling rates inferred from passive radio emissions measured by IMAGE RPI during geomagnetically quiet times, *J. Geophys. Res.*, *117*, A03221, doi:10.1029/2011JA017274.
- Hebb, D. O. (1949), *The Organization of Behavior: A Neuropsychological Theory*, vol. 34, John Wiley, Inc., p. 335, New York.
- Li, H., C. L. P. Chen, and H.-P. Huang (2000), *Fuzzy Neural Intelligent Systems: Mathematical Foundation and the Applications in Engineering*, CRC Press, p. 392, Boca Raton, Fla.
- Glorot, X., A. Bordes, and Y. Bengio (2011), Domain adaptation for large-scale sentiment classification: A deep learning approach, in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, edited by L. Getoor and T. Scheffer, pp. 513–520, New York.
- Goldstein, J., S. D. Pascuale, C. Kletzing, W. Kurth, K. J. Genestreti, R. M. Skoug, B. A. Larsen, L. M. Kistler, C. Mouikis, and H. Spence (2014), Simulation of Van Allen Probes plasmopause encounters, *J. Geophys. Res.*, *119*, 7464–7484, doi:10.1002/2014JA020252.
- Klavdianos, P. B., L. M. Brasil, and J. S. S. Melo (2013), *Face Recognition With Active Appearance Model (AAM), Image Processing: Concepts, Methodologies, Tools, and Applications*, vol. 3, IGI Global, pp. 1124–1144, Inf. Sci. Ref., Hershey, Pa., doi:10.4018/978-1-4666-3994-2.ch057.
- Kletzing, C. A., et al. (2013), The Electric and Magnetic Field Instrument Suite and Integrated Science (EMFISIS) on RBSP, *Space Sci. Rev.*, *179*, 127–181, doi:10.1007/s11214-013-9993-6.
- Krizhevsky, A., and G. Hinton (2011), Using very deep autoencoders for content-based image retrieval, in *Proceedings European Symposium on Artificial Neural Networks ESANN-2011*.
- Kurth, W. S., S. De Pascuale, J. B. Faden, C. A. Kletzing, G. B. Hospodarsky, S. Thaller, and J. R. Wygant (2015), Electron densities inferred from plasma wave spectra obtained by the Waves instrument on Van Allen Probes, *J. Geophys. Res. Space Physics*, *120*, 904–914, doi:10.1002/2014JA020857.
- LaBelle, J., D. R. Ruppert, and R. A. Treumann (1999), A statistical study of banded magnetospheric emissions, *J. Geophys. Res.*, *104*(A1), 293–303.
- LeDocq, M. J., D. A. Gurnett, and R. R. Anderson (1994), Electron number density fluctuations near the plasmopause observed by CRRES spacecraft, *J. Geophys. Res.*, *99*, 23,661–23,671, doi:10.1029/94JA02294.
- Lippman, R. P. (1987), An introduction to computing with neural nets, *IEEE ASSP Mag.*, *4*, 4–22, doi:10.1109/MASSP.1987.1165576.
- Marr, D., and T. Poggio (1976), Cooperative computation of stereo disparity, *Science*, *194*(4262), 283–287, doi:10.1126/science.968482.
- Mathworks.com (2015), Choose a Multilayer Neural Network Training Function—MATLAB & Simulink. [Available at: <http://www.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html>, Accessed 27 October 2015.]
- Mauk, B. H., N. J. Fox, S. G. Kanekal, R. L. Kessel, D. G. Sibeck, and A. Ukhorskiy (2012), Science objectives and rationale for the Radiation Belt Storm Probes mission, *Space Sci. Rev.*, *179*(1–4), 3–27, doi:10.1007/s11214-012-9908-y.
- McCulloch, W. S., and W. Pitts (1943), A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.*, *5*(4), 115–133, doi:10.1007/BF02478259.
- Minsky, M., and S. Papert (1969), *Perceptrons*, MIT Press, Cambridge, Mass.
- Mohamed, A., G. Dahl, and G. Hinton (2012), Acoustic modeling using deep belief networks, *IEEE Trans. Audio, Speech, Lang. Process.*, *20*(1), 14–22.
- Moldwin, M. B., M. F. Thomsen, S. J. Bame, D. McComas, and G. D. Reeves (1995), The fine-scale structure of the outer plasmasphere, *J. Geophys. Res.*, *100*(A5), 8021–8029, doi:10.1029/94JA03342.
- Moller, M. F. (1993), A scaled conjugate gradient algorithm for fast supervised learning, *Neural Netw.*, *6*(4), 525–533, doi:10.1016/S0893-6080(05)80056-5.
- Mosier, S. R., M. L. Kaiser, and L. W. Brown (1973), Observations of noise bands associated with the upper hybrid resonance by the Imp 6 radio astronomy experiment, *J. Geophys. Res.*, *78*, 1673–1679.
- Park, C. G. (1974), Some features of plasma distribution in the plasmasphere deduced from Antarctic whistlers, *J. Geophys. Res.*, *79*(1), 169–173, doi:10.1029/JA079i001p00169.
- Park, C. G., and D. L. Carpenter (1970), Whistler evidence of large-scale electron-density irregularities in the plasmasphere, *J. Geophys. Res.*, *75*(19), 3825–3836, doi:10.1029/JA075i019p03825.
- Reeves, G. D., et al. (2013), Electron acceleration in the heart of the Van Allen radiation belts, *Science*, *341*, 991–994.
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986), Learning representations by back-propagating errors, *Nature*, *323*(6088), 533–536, doi:10.1038/323533a0.
- Salakhutdinov, R., and G. Hinton (2009), Semantic hashing, *Int. J. Approximate Reasoning*, *50*(7), 969–978.
- Sibi, P., S. Allwynjones, and P. Siddarth (2013), Analysis of different activation functions using back propagation neural networks, *J. Theor. Appl. Inf. Technol.*, *47*(3), 1344–1348.
- Sheeley, B. W., M. B. Moldwin, H. K. Rassoul, and R. R. Anderson (2001), An empirical plasmasphere and trough density model: CRRES observations, *J. Geophys. Res.*, *106*(A11), 25,631–25,642.
- Summers, D., and R. M. Thorne (2003), Relativistic electron pitch-angle scattering by electromagnetic ion cyclotron waves during geomagnetic storms, *J. Geophys. Res.*, *108*, A01208, doi:10.1029/2002JA009489.
- Thorne, R. M., et al. (2013), Rapid local acceleration of relativistic radiation-belt electrons by magnetospheric chorus, *Nature*, *504*, 411–414, doi:10.1038/nature12889.
- Trotignon, J. G., J. Etcheto, and J. P. Thouvenin (1986), Automatic determination of electron density measured by the Relaxation Sounder on Board ISEE 1, *J. Geophys. Res.*, *91*, 4302–4320.
- Trotignon, J. G., et al. (2010), The WHISPER Relaxation Sounder and the CLUSTER Active Archive, in *The Cluster Active Archive, Studying the Earth's Space Plasma Environment*, edited by H. Laakso, M. Taylor, and C. P. Escoubet, pp. 185–208, Springer, Astrophysics and Space Science Proceedings, Berlin, doi:10.1007/978-90-481-3499-1_12.
- Wygant, J. R., et al. (2013), The Electric Field and Waves instruments on the Radiation Belt Storm Probes mission, *Space Sci. Rev.*, *179*(1–4), 183–220, doi:10.1007/s11214-013-0013-7.