







Article

Convolutional Neural Networks for Automated ULF Wave Classification in Swarm Time Series

Alexandra Antonopoulou ^{1,2,*} , Georgios Balasis ^{1,*} , Constantinos Papadimitriou ^{1,2,3} ,
Adamantia Zoe Boutsis ^{1,2} , Athanasios Rontogiannis ⁴, Konstantinos Koutroumbas ¹, Ioannis A. Daglis ^{2,5} 
and Omiros Giannakis ¹ 

- ¹ Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing, National Observatory of Athens, Metaxa and Vas. Pavlou St., 15236 Penteli, Greece; constantinus.0@gmail.com (C.P.); zboutsis@noa.gr (A.Z.B.); koutroum@noa.gr (K.K.); og@noa.gr (O.G.)
- ² Department of Physics, National and Kapodistrian University of Athens, Panepistimiopolis, 15784 Athens, Greece; iadaglis@phys.uoa.gr
- ³ Space Applications & Research Consultancy (SPARC), Aioulou St. 68, 10551 Athens, Greece
- ⁴ School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechniou St. 9, 15780 Athens, Greece; tronto@noa.gr
- ⁵ Hellenic Space Centre, Leoforos Kifissias 178, 15231 Athens, Greece
- * Correspondence: alexantonop@gmail.com (A.A.); gbalasis@noa.gr (G.B.)



Citation: Antonopoulou, A.; Balasis, G.; Papadimitriou, C.; Boutsis, A.Z.; Rontogiannis, A.; Koutroumbas, K.; Daglis, I.A.; Giannakis, O.

Convolutional Neural Networks for Automated ULF Wave Classification in Swarm Time Series. *Atmosphere* **2022**, *13*, 1488. <https://doi.org/10.3390/atmos13091488>

Academic Editors: Sergey Pulinetz and Alexei Dmitriev

Received: 15 July 2022

Accepted: 5 September 2022

Published: 13 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Ultra-low frequency (ULF) magnetospheric plasma waves play a key role in the dynamics of the Earth's magnetosphere and, therefore, their importance in Space Weather phenomena is indisputable. Magnetic field measurements from recent multi-satellite missions (e.g., Cluster, THEMIS, Van Allen Probes and Swarm) are currently advancing our knowledge on the physics of ULF waves. In particular, Swarm satellites, one of the most successful missions for the study of the near-Earth electromagnetic environment, have contributed to the expansion of data availability in the topside ionosphere, stimulating much recent progress in this area. Coupled with the new successful developments in artificial intelligence (AI), we are now able to use more robust approaches devoted to automated ULF wave event identification and classification. The goal of this effort is to use a popular machine learning method, widely used in Earth Observation domain for classification of satellite images, to solve a Space Physics classification problem, namely to identify ULF wave events using magnetic field data from Swarm. We construct a Convolutional Neural Network (ConvNet) that takes as input the wavelet spectrum of the Earth's magnetic field variations per track, as measured by Swarm, and whose building blocks consist of two alternating convolution and pooling layers, and one fully connected layer, aiming to classify ULF wave events within four different possible signal categories: (1) Pc3 wave events (i.e., frequency range 20–100 MHz), (2) background noise, (3) false positives, and (4) plasma instabilities. Our preliminary experiments show promising results, yielding successful identification of more than 97% accuracy. The same methodology can be easily applied to magnetometer data from other satellite missions and ground-based arrays.

Keywords: ULF waves; space weather; Swarm satellites; convolutional neural networks; classification; machine learning

1. Introduction

Ultra-low frequency (ULF) waves are produced by processes in the Earth's magnetosphere and solar wind [1]. The broader group of them, defined as Continuous Pulsations, is characterized by quasi-sinusoidal signals that persist for multiple periods. These geomagnetic pulsations are divided into five subcategories (Pc1-5) according to their frequency [2]. Magnetospheric ULF waves have a significant impact on charged particle dynamics in the radiation belts [3,4]. In particular, ULF waves can accelerate electrons with MeV energies in the radiation belts, which in turn can penetrate spacecraft shielding and produce a gradual

increase of static charge in their electrical components causing damage to subsystems, or eventually cause the total loss of a satellite [3]. Since the varying conditions in the radiation belts have potentially adverse effects on satellites and astronauts in space, ULF waves are of major importance for space weather.

The analysis of ULF pulsations is an active area of space research and much remains to be discovered about their generation and propagation processes. Recent studies for the analysis of ULF waves [5,6] have stimulated much progress in this area. In Balasis et al., 2013 [5], a wavelet-based spectral analysis tool has been developed for the classification of ULF wave events using data from the low-Earth orbit (LEO) CHAMP satellite, in preparation of the Swarm mission, while in Balasis et al., 2019 [6], a machine learning technique based on Fuzzy Artificial Neural Networks has been employed in order to detect ULF waves in the time series of the magnetic field measurements on board CHAMP. The analysis of ULF pulsations, and in particular the Pc3 ULF wave events (20–100 mHz) that a topside ionosphere mission such as Swarm can track with high accuracy, can help to unravel the processes that play a critical role in the generation of these waves and their key propagation characteristics [7].

LEO observations of ULF waves can only be reliably done (i.e., without too much spatial aliasing) for the Pc1 (0.2–5 Hz) and Pc3 waves. Due to the fast motion through field lines in a LEO orbit the lower frequency Pc4–5 waves (1–10 mHz) cannot be accurately determined, their period being longer than the spacecraft transition time through the wave region.

Swarm is a satellite constellation of three identical spacecraft launched on 22 November 2013. Each of the three Swarm satellites performs high-precision and high-resolution measurements of the strength, direction and variation of the magnetic field, accompanied by precise navigation, accelerometer, plasma and electric field measurements. Two of the Swarm satellites, Swarm A and Swarm C, are flying nearly side-by-side in near-polar orbits at an initial altitude of about 465 km, while the third satellite, Swarm B, flies in a slightly higher orbit, at about 520 km initial altitude [8]. Each satellite is equipped with magnetic sensors, measuring a combination of various contributing sources: the Earth's core field, magnetised rocks in the lithosphere, external contributions from electrical currents in the ionosphere and magnetosphere, currents induced by external fields in the Earth's interior and a contribution produced by the oceans. Each satellite carries an Absolute Scalar Magnetometer (ASM) measuring Earth's magnetic field intensity, and a Vector Fluxgate Magnetometer (VFM) measuring the magnetic vector components [9].

In Balasis et al., 2015 [7] we proceeded with the analysis of 1 year of Swarm mission ULF wave observations to generate wave power maps at the satellites' orbiting altitudes. We compared these maps against the wave power map derived from more than 9 years of CHAMP data. Prior to this, we had used the electron density data recordings on board Swarm and CHAMP to correct the ULF wave power maps for contamination by the post-sunset equatorial spread-F (ESF) events [10]. These instabilities also called plasma bubbles [11] are generally accompanied by local depletions of the electron density. Thus the availability of electron density measurements provides a key way to discriminate between ESF events and ULF wave events since the latter are not associated with plasma depletions. In general, the latitudinal distribution of the ESF magnetic signatures is symmetrical about the dip equator, peaking at a distance of $\sim\pm 10^\circ$ in latitude from the dip equator. Furthermore, ESF events appear to have low occurrence rates in June solstice concentrating in the African and Pacific sector and high occurrence rates above the Atlantic sector during December solstice.

By correcting Swarm and CHAMP magnetic field data for ESF signatures, we had obtained the pure compressional (field-aligned) Pc3 wave signal [7]. From Swarm and CHAMP Pc3 wave power maps [7], we were able to confirm earlier work (based on a single satellite observations that CHAMP mission was) and provided new observations based on the three-satellite measurements. Swarm maps presented features of the ULF wave power in the topside ionosphere with unprecedented detail showing subtle differences in the wave

power observed between the upper satellite and the lower pair of satellites and between Swarm and CHAMP missions. We found excellent agreement between the ULF wave power characteristics observed by Swarm A and C and those seen by the upper satellite, which indicates that the satellites probe the same wave events. The similarities between the maps can be attributed to the strong Universal Time (UT) dependence of the Pc3 activity. Furthermore, Swarm ULF wave power maps showed that the Pc3 wave power peaks around noon, which seems to be a general feature of both magnetospheric compressional and ground Pc3 pulsations. A puzzling enhancement, not predicted by current ULF wave theories, of compressional Pc3 wave energy was revealed by Swarm in the region of South Atlantic Anomaly (SAA). Probably, compressional Pc3 wave occurrence is favored by a low magnitude of the geomagnetic field, and consequently of Alfvén velocity, in this region.

With almost 9 years of Swarm data currently available, the employment of big data analysis techniques is essential to fully deploy the science discovery capabilities of this unique data set. In this work we mainly use data from the Swarm VFM instrument and apply a wavelet-based technique in order to extract the image of the power spectrum of the measured magnetic field intensity along a satellite track, which is finally the input to our machine learning (ML) classification model.

Artificial Intelligence has been successfully introduced in the fields of space physics and space weather since the 1990s and has yielded promising results in modeling and predicting many different aspects of the geospace environment.

For example, in the 1990s and 2000s, several efforts have been made to use artificial intelligent techniques to predict geomagnetic indices and radiation belt electrons [12–22].

More recently, studies have been conducted that use neural networks to forecast global total electron content [23], Dst index [24] and Kp index [25], Bayesian probabilistic approaches for solar corona investigations [26], and multiple machine learning techniques for solar flare forecasting [27] and solar energetic particle events prediction [28,29].

Moreover, another ML technique that seems to become very popular in the Space Weather community recently giving satisfying results is the Long Short-Term Recurrent Neural Networks (LSTM RNN), which is a deep neural architecture [30–33].

What makes space weather a perfect fit for the machine learning framework is the vast number of freely accessible data which are generally of high quality and require little preprocessing effort and time [34].

A promising machine learning method for solving image classification problems is Convolutional Neural Networks (ConvNets). The ConvNets are not a new idea; for example, in 1989, LeCun et al. [35] used a ConvNet to recognize handwritten digits, while in 2012, Krizhevsky et al. [36] trained a large, deep ConvNet to classify images in a large number of categories. Moreover, ConvNets have not only performed well in image classification problems, but they are also applied successfully on non-image tasks, e.g., in natural language processing [37].

In the field of space physics also, an effort has been made recently to use ConvNets to predict the magnetic flux rope's orientation from in situ solar wind observations, giving promising results [38].

ConvNets, or in general Neural Networks, are very interesting machine learning methods. They have a multilayer structure, and each layer can learn an abstract representation of the input. This process is called Feature Learning or Representation Learning [39]. Feature Learning is a method that allows a machine to learn from raw data. The machine takes the raw data as input and automatically learns the required features to solve the machine learning problem. For instance, in image classification, the raw data is an image represented by a matrix of numbers, each corresponding to an image pixel. These matrices are fed into the ConvNet, and the ConvNet learns useful features from these images to solve the classification problem.

Many different ConvNet architectures can be found in the literature, but in general their basic components remain the same. Considering the famous LeNet-5 [40], it consists of three types of processing components, namely convolutional, pooling, and fully connected.

In this work, we develop a ConvNet model, based on these three principle processing components, to classify ULF wave events. We train the model using as input spectral images, extracted by processing Swarm magnetic field data. Specifically, we focus on the Pc3 ULF waves, which are detected in the frequency range of about 20–100 mHz. The goal is to automatically detect Pc3 ULF pulsations and to discriminate them from other signals detected in the same frequency range but with different characteristics. Specifically, the four different categories we aim to identify are: “Pc3 ULF waves”, “Background noise”, “False Positives”, and “Plasma Instabilities”.

The present paper is organized as follows: At the beginning of Section 2, the basic theory of neural and convolutional neural networks, as well as two more classifiers that are used in this study (k-Nearest Neighbors and Support Vector Machines) as benchmarks for assessing the performance of the proposed ML model, is introduced (Sections 2.1–2.5), while the dataset and the methodology are described in Sections 2.6 and 2.7, respectively. In Section 3, we present the results of this study and, finally, Section 4 summarizes our findings.

2. Materials and Methods

2.1. Image Classification

In an image classification problem, the goal is to assign to an image a label out of a known set of classes, indicating the class in which the image belongs. One way to achieve this is to adopt a suitable model. The adopted model takes as input a single image and outputs probabilities associated with the relevance of the input image with each of the known classes.

Hence, the task is to turn a considerable amount of numbers (i.e., a matrix representing the intensity of the pixels of the image) into a single label, such as “ULF wave Event” in the present framework. The complete image classification pipeline can be described as follows [41]:

Input: a set of N images, each labeled with one out of K different class labels. This is referred to as the training set.

Learning: use the training set to built within the adopted model internal representations of each class. This step is referred to as training a classifier or learning a model.

Evaluation: evaluate the performance of the model by asking it to predict class labels for images with a priori known labels that were not used for the training of the model (they form the test set). This is because we want to compare the true labels of these images with those predicted by the classifier. High classification rates indicate that the model generalizes well on images that have not been used for its training, and thus, it can be trusted for operational purposes, where we do not know the label of a particular image and we rely on the classification performed by the classifier.

2.2. Artificial Neural Networks (ANNs)

Artificial neural networks are models inspired by the way the human brain is constructed. Their building block is the artificial neuron, whose basic structure is shown in Figure 1. The node (neuron) receives an input and uses some internal parameters (i.e., weights (W) and biases (b) that are learned during training) to compute the output. Such nodes can be considered as simple examples of “learning machines” whose free parameters are updated by a learning algorithm, in order to learn a specific task, based on a set of training data [42,43]. Returning to Figure 1, let us describe the basic operation of a single neuron. We can see two distinct operations: the first one is a linear combination of input features and parameters, the other is a nonlinear operation, performed by an activation function, such as the Sigmoid:

$$S(z) = \frac{1}{1 + e^{-z}} \quad (1)$$

The most well-known class of ANNs are the Multilayer Feedforward Neural Networks (FNN), which in some contexts are also called Multilayer Perceptrons (MLP), containing one or more hidden layers with multiple hidden units (neurons) in each of them [42].

The training of the model consists then, of optimizing the network’s parameters so that to minimize the output estimation error.

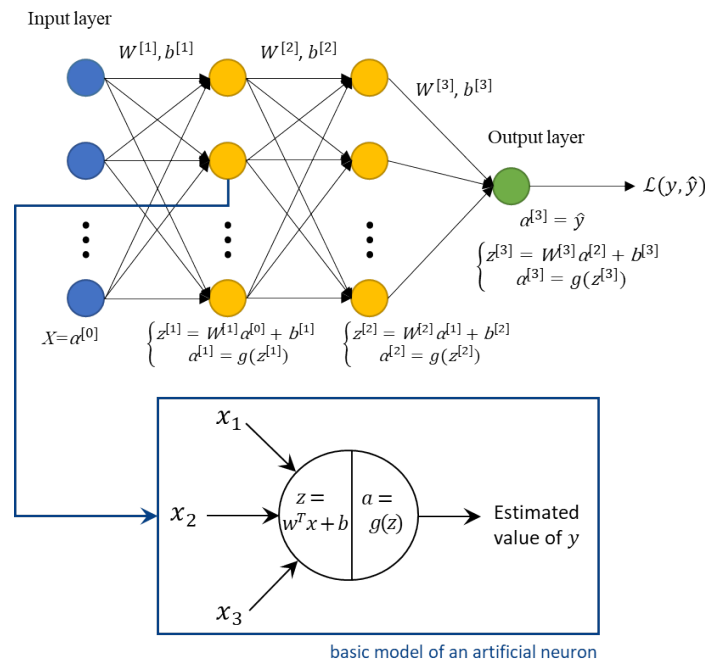


Figure 1. Basic architecture of a Neural Network (top) and “zoom-in” a single unit of it, the artificial neuron (bottom).

2.3. Convolutional Neural Networks (ConvNets)

Convolutional neural networks are a class of feedforward neural networks: they are made up of neurons with learnable parameters (weights and biases). Each neuron receives an input, performs a dot product, and the result usually passes through a nonlinear function. The difference is that in ConvNet architectures the inputs are images (i.e., matrices of at least two dimensions), allowing us to encode certain properties in the architecture of the model that enable a more efficient implementation of forward propagation and significantly reduce the number of parameters in the network [41].

In general, an image needs to be flattened into a single array to be used as an input into a ANN. In ConvNets, on the other hand, the input comprises the raw images and therefore, their architecture is designed in a more sophisticated manner. In particular, unlike an ANN, the layers of a ConvNet have neurons arranged in three dimensions: width, height, depth. The neurons in a ConvNet layer are only connected to a small portion of nodes of the previous layer that are spatially close to each other, rather than to all neurons as in ANNs (see Figure 2). Moreover, the final output layer is a one-dimensional vector, whose length equals to the number of classes. Its i^{th} position is the probability with which the input image belongs to the i^{th} class.

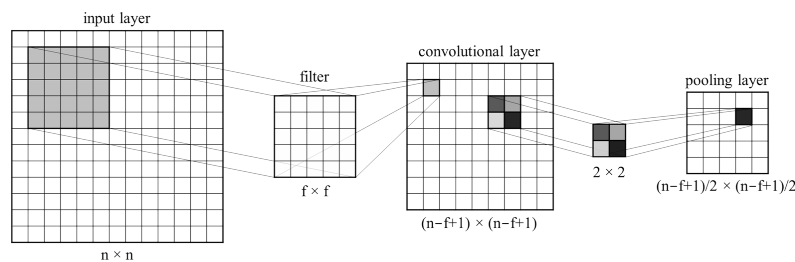


Figure 2. Example of the two major processes in a ConvNet (in 2-D): convolution and max-pooling.

2.4. Layers Used to Build ConvNets

The ConvNet architecture consists of three types of layers: convolution, pooling, and classification. These layers are combined together to form a complete ConvNet architecture.

Convolutional layer: It is the main building block of a Convnet that performs most of the heavy computation and whose parameters constitute sets of learnable filters. Each filter is spatially small (along width and height), but spans the entire spectral depth of the input volume (i.e., the spectral bands of the input image). During the forward propagation, each filter is convolved across the width and height of the input volume, and dot products are computed between the filter's entries and the input at any position. In other words, the convolution operation computes a dot product of the filter coefficients with a local region of the image, which has exactly the same dimension as the filter (Figure 2 left). Each filter in a convolutional layer generates a separate 2-dimensional "activation map". These activation maps are stacked along the depth dimension to produce the output activation map also called the output feature map. Three hyperparameters control the size of the output volume (i.e., the output activation map): depth, stride and padding. The depth of the output volume corresponds to the number of filters, each learning to identify different aspects of the input. The stride is the step with which we slide the filter within the image. Larger strides produce smaller output volumes. Zero-padding is used to control the spatial size of the output volumes, and usually to preserve the spatial size of the input so that the width and height of the input and output matches exactly. The amount of padding depends on the size of the filter. The spatial size of the output volume is calculated as a function of the size of the input volume N , the filter size F , the stride with which the filters are applied S , and the amount of zero-padding P used on the border. The number of neurons that "fit" is given by [41]:

$$\frac{N - F + 2P}{S} + 1, \text{ where } P = (F - 1)/2. \quad (2)$$

Pooling layer: In a ConvNet architecture, the pooling layer is periodically placed in-between successive convolutional layers. Its role is to merge semantically similar features into one [35]. More precisely, its function is to gradually reduce the spatial size of the representation in order to reduce the number of parameters and computations in the network, and, consequently, also to control "overfitting", an effect that will be discussed again later. The pooling layer operates independently on every depth slice of the input and resizes it spatially, using most commonly the max operation (other operations are also used, such as the average or the L2-norm functions). The depth dimension remains unchanged.

Classification layer: It is basically an ordinary fully connected multilayer FNN, where each neuron is connected to every other neuron in the layer before it.

It should be mentioned that the only difference between fully connected (FC) and convolutional (CONV) layers is that the neurons in the CONV layer are connected only to a local region of its input and that they share parameters (this greatly reduces the number of parameters). Nevertheless, the neurons in both FC and CONV layers compute dot products, implying that their functional form would be the same [41].

2.5. The k -Nearest Neighbors (k -NN) and the Support Vector Machines (SVM) Classification Algorithms

In order to assess the performance of the proposed classification model, we compared our results with other classifiers. To do so, we run various versions of the k -Nearest Neighbor (k NN) classifier and the Support Vector Machines (SVM) classifier, which are both described briefly in the following sections.

k -NN

k -NN is one of the most popular and simple classifiers [44,45]. k -NN is a non-parametric classifier and, as a consequence, it does not have a training stage [45]. It assigns an unlabeled sample in a class according to its distance from the labeled samples.

Specifically, for the data point q to be classified, we identify first its k nearest neighbors, among an available set of classified samples (Figure 3). Then, we assign q to the class where most of the k -nearest neighbors of q belong. The technique is referred to as k -Nearest Neighbour (k -NN) Classification since k nearest neighbors are used in determining the class [46].

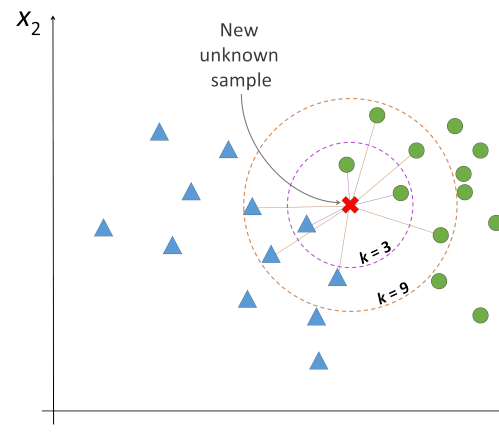


Figure 3. k -NN example: among the three nearest neighbors of the unclassified sample, two belong to the green class and one to the blue class. Thus, the 3-NN classifier classifies the sample to the green class. Similar rationale is applied for the $k = 9$ case.

The performance of a k -NN classifier is mainly affected by the choice of k as well as the distance metric applied [44], in order to quantify the “similarity” between two vectors. Popular measurements used for quantifying the similarity between data points are the Euclidean Distance and the Manhattan Distance. These are special cases of the Minkowski Distance metric, defined in Equation (3). For each $x_i \in D$ we can calculate the distance between q and x_i as follows:

$$MD_p(q, x_i) = \left(\sum_{f \in F} |q_f - x_{if}|^p \right)^{\frac{1}{p}} \quad (3)$$

where q_f is the unknown sample, $f \in F$ (where F is a set of features by which the examples are described), x_{if} is the training sample of the dataset D , $x_i \in D$ and $i \in [1, |D|]$, each labeled with a class label $y_j \in Y$.

Specifically, the L1 Minkowski distance where $p = 1$ is the Manhattan distance while the L2 distance where $p = 2$ is the Euclidean distance [46].

k -NN has several advantages: First of all, it is one of the simplest and straightforward machine learning techniques. It is a method that has a single hyperparameter that has to be defined (k), and its performance is competitive in various applications [44,47]. On the other hand, k -NN has also disadvantages: It has a high computational cost, since for each data point to be classified, one has to compute its distance from all the available classified data points (the higher the number of the latter, the higher the computational cost becomes). Moreover, it is quite sensitive to irrelevant or redundant features, as all features contribute to similarity and subsequently to the classification. [45].

SVM

SVM is considered as the method of choice on various fields of applications. Fast implementation and ease of use are some of the reasons for their success. SVM requires only a few architectural decisions to be taken [48]. In its most primitive form, SVM is a linear classifier and it is studied for the case where the two classes in a two-class classification problem are linearly separable.

The basic idea of SVM classification relies on the determination of the decision boundaries. The decision boundary (a hyperplane in its primitive form) that separates the “positive” training data samples (i.e., label = 1, “above” the decision boundary) from the “negative” training data samples (i.e., label = −1, “below” the decision boundary) (Figure 4). The best hyperplane is the one that gives the largest space (maximum margin) between the nearest points of the two classes. These points are called the support vectors.

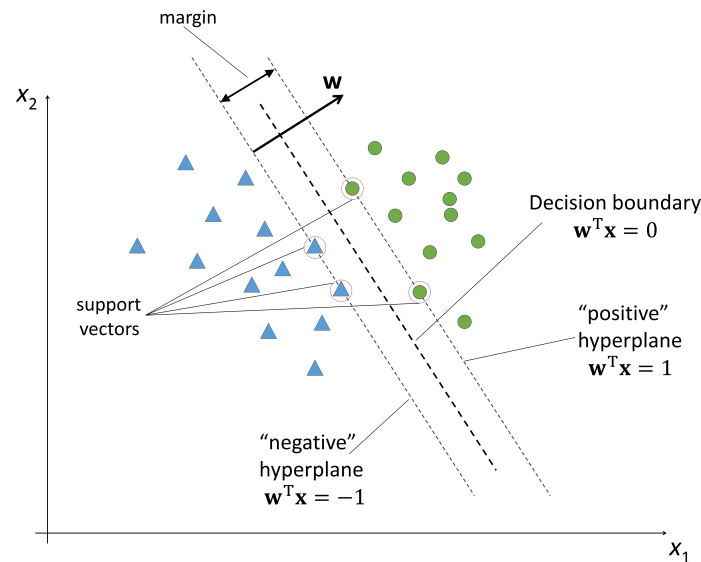


Figure 4. In the SVM method the goal is to maximize the margin between the two classes.

For the two-dimensional case, the equation of the decision boundary can be written as

$$w^T x = w_1 x_1 + w_2 x_2 = 0 \quad (4)$$

where $w = [w_1, w_2]^T$, $x = [x_1, x_2]^T$. The margin is defined by the two parallel boundaries $w^T x = 1$ and $w^T x = -1$, shown in Figure 4, and its size turns out to be: $\frac{2}{\sqrt{w_1^2 + w_2^2}}$.

With more than two features (i.e., for an f -dimensional space with $f > 2$), this distance generalizes to $\frac{2}{\|w\|}$ [49], where $\|w\| = \sqrt{w_1^2 + \dots + w_f^2}$. SVM have also the ability to deal well with nonlinear classification problems. Such problems are solved by first mapping the f -dimensional input space into a high dimensional feature space, where the problem is likely to become “more linear”, for a suitably chosen nonlinear transformation map. Then in this high dimensional feature space a linear classifier is constructed, which acts as nonlinear classifier in input space [50].

2.6. Swarm-Tailored Methodology

Here, we describe the methodology pipeline one can follow to identify and classify automatically Pc3 ULF waves, using Swarm magnetic field data and our trained ConvNet model (Figure 5):

The first step is to take data from the Swarm Vector Field Magnetometer (VFM) [9], North-East-Center (NEC) local Cartesian coordinate frame, 1 Hz sampling rate, and calculate the total magnitude.

Our team has worked thoroughly with both ASM and VFM instruments, but for the scope of this work, after the subtraction of the CHAOS model and the application of the HP filter, the results are almost identical (especially for the frequency range of Pc3 waves that are studied here).

The second step is to keep only the external contributions of the geomagnetic field by subtracting the internal part of the CHAOS-6 magnetic field model [51] from the total magnitude. The produced time series are then segmented into low and mid-latitudinal

tracks, i.e., keep only the parts of the orbit for which the satellite lies in the region from -45° to $+45^\circ$ in magnetic latitude. This is done in order to exclude the influence of polar field aligned currents (FAC) that might affect the measurements [52]. Each satellite track between $\pm 45^\circ$ corresponds to a 15-minutes time interval. In addition, the magnetic field time series is filtered using a high-pass Butterworth filter with a cut-off frequency of 20 mHz, to focus only on the Pc3 ULF waves (from 20 to 100 mHz approximately). Next step is to apply wavelet transform [6,7] on the filtered Swarm tracks, to end up with spectral images which are then passed through the trained ConvNet model. So given a wavelet power spectrum image as input, the network outputs a probability for each one of the four classes. All the Swarm data are available on the online Swarm database [53].

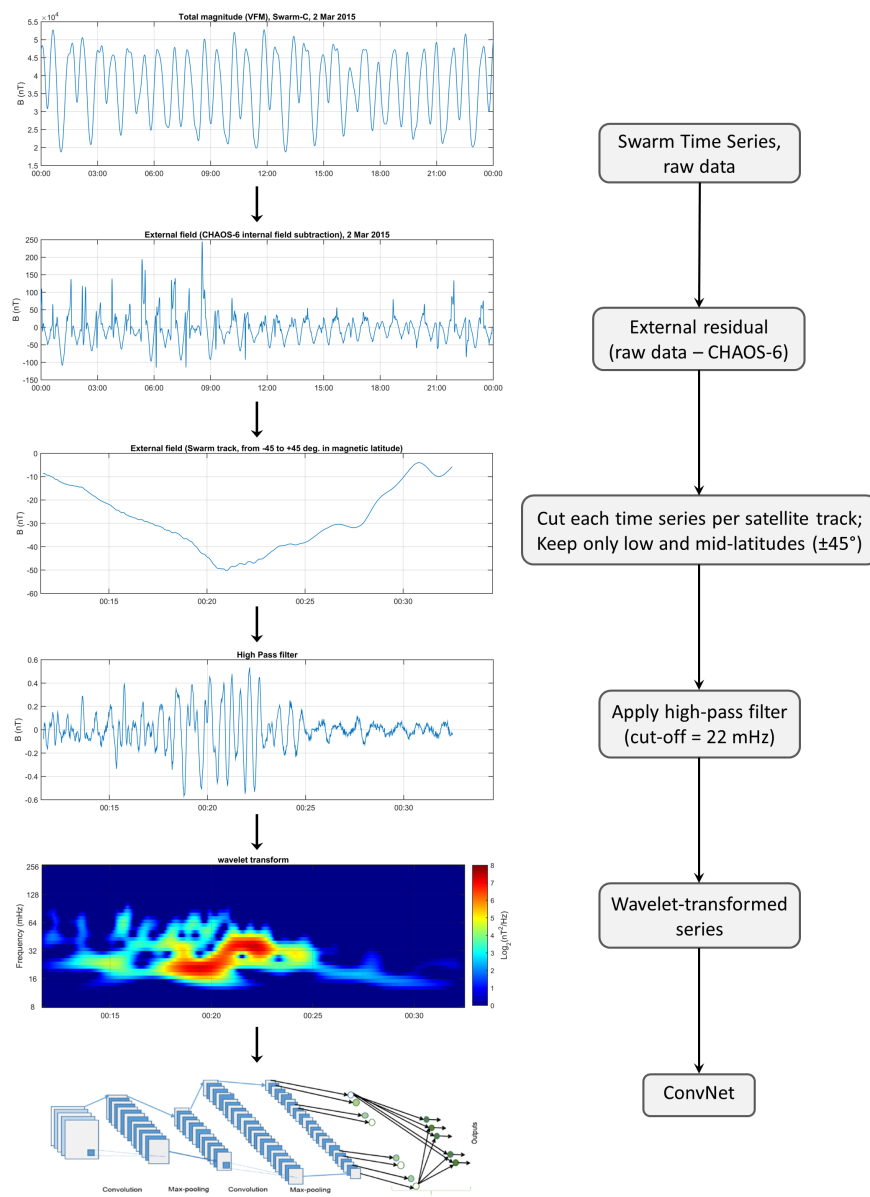


Figure 5. The proposed methodology, begins from Swarm VFM magnetic field measurements, includes the use of CHAOS-6 model, segmentation into low and mid-latitudinal satellite tracks, filtering and wavelet transform. The final step involves the use of the trained ConvNet model, that takes as input the produced wavelet images, and gives as output a probability measure for each of the 4 classes.

As already mentioned, the constructed ConvNet model classifies the satellite tracks in four categories. In more details these classes are (see Figure 6):

1. Pc3 ULF Wave Events, detected in the frequency range 20–100 mHz,
2. Background Noise, i.e., tracks without significant wave activity,
3. False Positives (FP's), i.e., signals that exhibit wave power in the Pc3 range but are not true ULF pulsations, containing measurements contaminated by short lived anomalies, such as spikes or abrupt discontinuities due to instrument errors, and
4. Plasma Instabilities (PI's), attributed primarily to ESF events which are predominantly present in the nightside tracks and have similar characteristics to Pc3 waves even though they are not true ULF pulsations [7].

Classification Process: The criteria used for the manual classification of the power spectra can be found in Papadimitriou et al., 2018 [54]. For each spectrum, the maximum power per second is calculated and all segments of consecutive points that exceed a threshold of $0.5 \text{ nT}^2/\text{Hz}$ are labeled “candidate events”. Each candidate is tested against a series of criteria that help rule out artificial signals that might result from instrument or telemetry errors. Specifically, for the candidate event:

- it must exhibit a duration of at least 2 times its peak period,
- it must have an amplitude that does not exceed certain limits (10 nT),
- and it must be smooth enough to constitute a continuous pulsation, so its difference series must always be smaller than 1 nT.

For more details the reader is referred to [54].

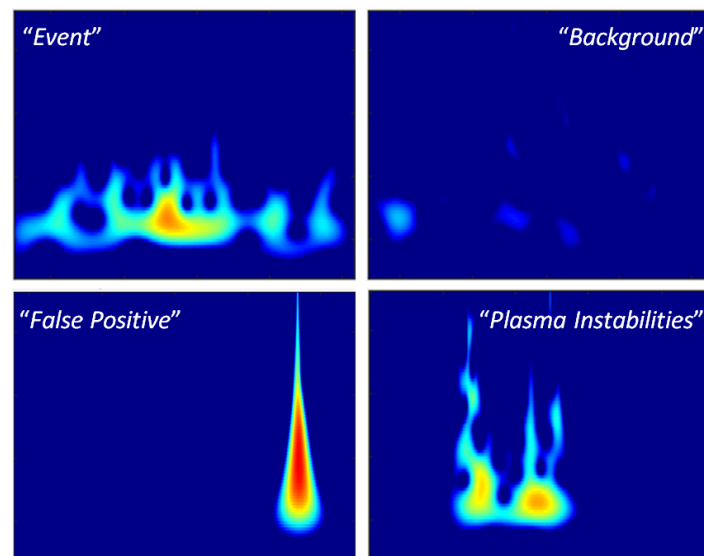


Figure 6. In this work we solve a 4-class image classification problem, with the four categories to be: “Pc3 ULF wave Events” (top left), “Background Noise” (top right), “False Positives” (bottom left), and “Plasma Instabilities” (bottom right).

2.7. Data & Training of the Network

The data used for the training of our ML model is the total magnitude derived from the Swarm Vector Field Magnetometer (VFM) measurements [9], North-East-Center (NEC) local Cartesian coordinate frame (where B_N is the component towards geographic North, B_E is the component towards geographic East, and B_C is the component towards the center of the Earth), 1 Hz sampling rate (MAGX_LR_1B Product), for February, March and April of the year 2015. The constructed image dataset underwent manual annotation with four labels, corresponding to our four different classes. The whole dataset consists of 2620 samples, from which the 80% were used for the training set (i.e., 2096 samples). We validated the ConvNet on a test dataset that consists of the rest 20% of total samples (i.e.,

524 samples), which was also annotated manually. The dataset was first shuffled and then split. Thus, the split of the samples in the training and test sets is completely random with no regards to time.

During the training phase, the cross-validation method has been employed in order to decrease the statistical fluctuation in the final error estimations. Cross-validation is a statistical method commonly used to gain a more accurate estimate of the performance of a machine learning model. The cross-validation comprises of the following steps:

- Divide the training dataset into k subsets and perform training k times in total. Each time use $k - 1$ subsets for training and the remaining one for testing.
- For each one of the k' times, compute the accuracy on the training and the test set (i.e., $[ac_{tr(1)}, ac_{te(1)}], \dots, [ac_{tr(k)}, ac_{te(k)}]$).
- Finally, compute the mean (μ) and standard deviation (σ) values of the accuracies of the training subsets $[tr(1), \dots, tr(k)]$ and the test subsets $[te(1), \dots, te(k)]$.

Intuitively, small variation shows that we have good quality of data and that the dataset is well-labeled. In any other case, the dataset should be examined again. Moreover, another use of cross-validation is to evaluate different settings (i.e., different values for the hyperparameters of the model) on each batch. Then, use the hyperparameters that achieved the highest accuracy in a specific batch, to train the model (using the whole training set). This way we can better tune hyper-parameters such as the kernel size or the number of kernels in the convolution layers, kernel size of the pooling layers, learning rate, dropout percentage, number of epochs, etc., and achieve better performance of the estimator on the test set.

We built a ConvNet architecture consisting of 3 major stages (Figure 7): two alternating convolution and max-pooling layers, responsible for feature extraction, and one fully connected (FC), which plays the role of the classifier. In the FC layer we use a regularization technique in order to avoid “overfitting”, which occurs when the model fits very well on the peculiarities of the data set at hand, which, inevitably, leads to poor generalization performance on unseen data. So, to prevent complex co-adaptations on the training data, we use the “Dropout” technique [55]: it is a technique meant to prevent overfitting the training data by dropping out units in a neural network. In practice, neurons are either dropped with probability p or kept with probability $1 - p$ [41].

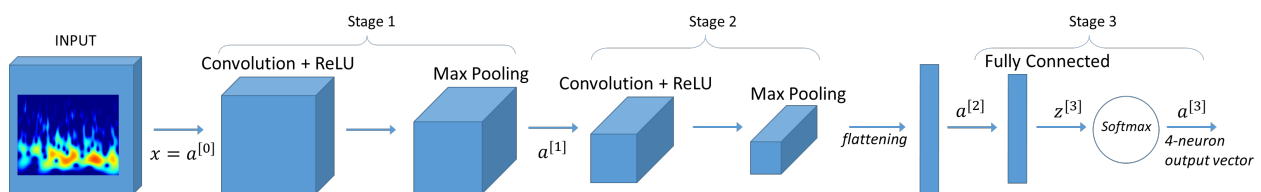


Figure 7. An abstract representation of our ConvNet architecture. Layers are presented as volumes, with the input volume to be the labeled images, followed by two alternating convolution and pooling layers, and one fully connected layer.

The non-linear activation function used in the convolutional layers is the rectified linear function (ReLU),

$$\text{relu}(z) = \max(z, 0). \quad (5)$$

Another critical parameter of the model is the “learning rate”: it indicates at which pace the weights get updated. This can be fixed or adaptively changed. The current most popular method is called Adam, which is a method that adapts the learning rate [41].

The training was performed for 100 epochs, using Adam optimization [56], i.e., we update the parameters in the optimization loop 100 times.

Other important characteristics and details of the ConvNet architecture is presented in Table 1.

The Swarm data analysis was implemented using the framework of the MATLAB programming environment, while the ConvNet model was implemented using Python and its machine learning framework, TensorFlow.

To summarize all the information on the model's training:

- Data used: total magnitude, Swarm VFM, NEC local Cartesian coordinate frame, 1 Hz sampling rate (MAGX_LR_1B Product), for February, March and April of the year 2015.
- Number of total samples: 2620 samples, manually annotated with 4 labels.
- Input: pairs of wavelet power spectra images with their annotation (class label)
- Training set—Test set split: 80% (2096 samples)—20% (524 samples) of total sample
- Layers: 2 convolutional, 2 max-pooling, 1 fully connected.
- Parameter initializer: Xavier Initialization [57]
- Activation functions: ReLU, Softmax [58]
- Cost function: Cross-entropy (Log Loss) [59]
- Optimizer: Adam Optimization
- Extra: Dropout Regularization.

The selected time interval of study is centered around the strongest magnetic storm of solar cycle #24, i.e., the storm which occurred on 17 March 2015 with a minimum Dst index of -223 nT [60–63].

Table 1. Information about the network's architecture (number of filters on each CONV layer, kernel size (f), stride (s)).

Layers	Details
Convolutional layer 1	8 filters, $f = 4, s = 1$
Max-Pooling layer 1	$f = 8, s = 8$
Convolutional layer 2	16 filters, $f = 2, s = 1$
Max-Pooling layer 2	$f = 4, s = 4$
Fully Connected layer	4-neuron output

3. Results

After the training of the network by following all the necessary steps described in Section 2, we obtained an overall accuracy on the training set equal to 98.3% and on the test set 97.3%.

Another measure we use to verify our results is the Heidke Skill Score (HSS), which measures the fraction of correct predictions after eliminating those which would be correct due purely to random chance [64], $HSS = 96.2\%$. The Heidke Skill Score is computed based on a generalized multi-category contingency table showing the correlations between predictions and observations, where $n(F_i, O_j)$ denotes the number of predictions in category i that had observations in category j , $N(F_i)$ denotes the total number of predictions in category i , $N(O_j)$ denotes the total number of observations in category j , and N is the total number of predictions [65]:

$$HSS = \frac{\frac{1}{N} \sum_{i=1}^K n(F_i, O_i) - \frac{1}{N^2} \sum_{i=1}^K N(F_i)N(O_i)}{1 - \frac{1}{N^2} \sum_{i=1}^K N(F_i)N(O_i)} \quad (6)$$

To visualize the performance of our classification model, we calculate the confusion matrix (Figure 8 left), and the Precision and Recall metrics [66,67] derived from it (Figure 8 right). In the confusion matrix we have two sections: the predicted classifications (rows) and the actual classifications (columns), subsectioned for each class. Looking at the Figure 8 (left), we can see that each entry in the confusion matrix denotes the number of predictions that have been classified correctly (in the diagonal positions of the matrix) or incorrectly (in the rest positions of each row, for each class). This way we can not only calculate the

accuracy for each class, but also observe which classes are confused with each other and which show good performance being well-separated from the others.

The True Positives is the number of correctly classified samples for each class. In general, Precision is calculated by $TP/(TP+FP)$ and Recall $TP/(TP+FN)$, where TP is the True Positives, FP is the False Positives, and FN the False Negatives.

For example, for the “Background noise” class, True Positives (TP) is the top-left cell in the confusion matrix (167). False positives (FP) are all the cells where other types of signals (“Events”, “PI’s”, or “FP’s”) are predicted as “Background noise”. These would be the cells to the right of the True Positives cell ($2 + 3 + 0$). False Negatives (FN) would be any occurrences where “Background noise” signals were classified as either “Events”, “PI’s”, or “FP’s”. These would be the cells below the True Positives cell in the confusion matrix ($1 + 0 + 0$).

Figure 9 presents the confusion matrix and the Recall and Precision metrics for the SVM classification method.

Figure 10 presents the results from the comparison of the ConvNet performance with the k-NN and SVM classification performance. For the implementation of these two classification methods, the Python’s scikit-learn tool has been utilized. The k-NN with $k = 5$ and $p = 1$ (Manhattan distance metric) gives 57.5% and the SVM 88.1%.

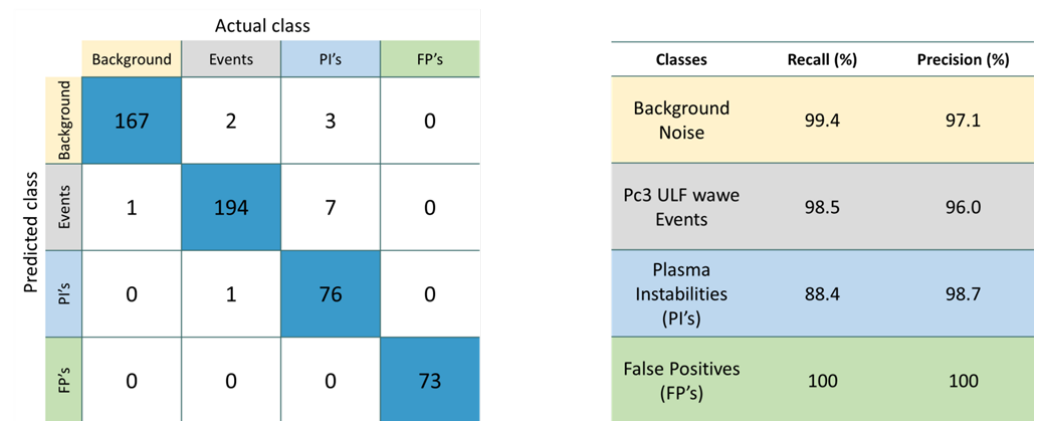


Figure 8. ConvNet performance: the confusion matrix (left) and the Precision and Recall on each class (right) of the test set.

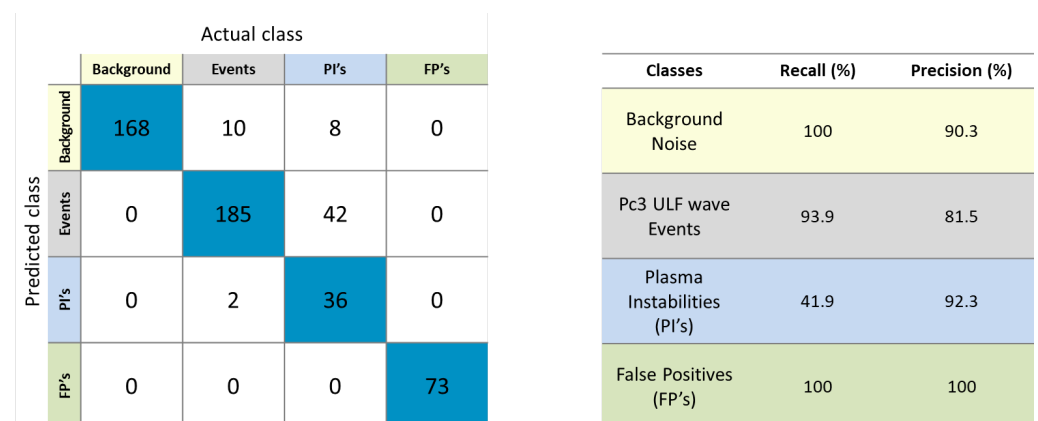


Figure 9. SVM performance: the confusion matrix (left) and the Precision and Recall on each class (right) of the test set.

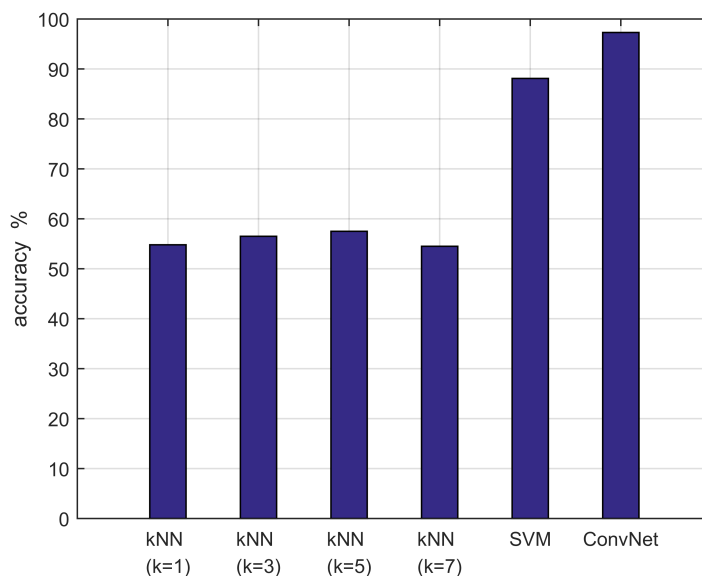


Figure 10. Comparison of the network’s accuracy on the test set with k-Nearest Neighbors and Support Vector Machines classifiers. ConvNet achieves the highest accuracy.

4. Conclusions & Discussion

Herein, we demonstrate the applicability of a machine learning classification method, widely used in Earth Observation for satellite images, to a Space Physics problem, namely we employ a Convolutional Neural Network to identify ULF wave events in Swarm satellite time series. First, we apply a wavelet-based technique to the magnetometer data onboard Swarm in order to infer the image of the power spectrum of the corresponding satellite track. Then, we use the ConvNet to derive the class of the specific image, targeting at the identification of Pc3 waves, which is the most efficiently resolved frequency range (20–100 MHz) of ULF waves, observed by a topside ionosphere mission flying at LEO. To our knowledge, this is the first time that a ConvNet is used within such an upper atmosphere study. Given the enormous wealth of magnetometer data collected over the last decades both from spacecraft and ground-based networks, ConvNet can play a key role in classifying various natural signals (e.g., waves, ESF events) contained within these data. The latter may lead to better understanding of the generation and propagation of wave phenomena and their contribution to the dynamics of the magnetosphere-ionosphere coupling system, including wave-particle interactions. It could also provide more accurate spatial and temporal distributions of ESF events, which is of paramount importance for space weather-related applications.

The methodology used in this work provides promising preliminary results. Figure 8 shows the confusion matrix of the test dataset. Obviously, in an ideal classifier the confusion matrix must be diagonal, with the off-diagonal elements equal zero. We can see that the confusion matrix is almost diagonal, with small numbers in the non-diagonal positions of the matrix. Moreover, a class that presents a considerable number of misclassified samples is the “Plasma Instabilities” class and specifically we can see that they are misclassified as Pc3 ULF Wave Events. The False Positives class presents the highest performance but the result is not so representative due to its small number of samples.

Looking at Figure 10, we compare the ConvNet’s performance with that of two famous classifiers: the k-Nearest Neighbor (kNN) classifier and the very competitive Support Vector Machines (SVM) classifier. It is obvious that the ConvNet method achieves the best results with the highest accuracy.

During the manual annotation of the spectral images for the construction of the labeled dataset, the main challenge was the separation of the “Plasma Instability” class samples, which finally presented the lowest accuracy after the training. To do so, we combined information of the electron density product and the latitude of the satellite tracks, the latter

giving information for the nightside or dayside position of the satellite and, together with the electron density information, giving a critical indication for the image to belong either in the Pc3 ULF wave Event class or in the Plasma Instability class. As a next step, we are planning to introduce this information also inside our machine learning model aiming to reduce the misclassified samples on the specific class and achieve better performance. As a continuation of this work, we aim to apply this method and train the same ML model with much more data, which are available from the Swarm mission, and in particular, we aim to exploit the data from the beginning of the mission onwards. Finally, this new methodology has the potential to be applied to identify ULF waves in other frequency ranges, e.g., Pc1/EMIC, Pc2, Pc4 and Pc5, on observations from other satellite missions or even on ground-based observations. To summarize our findings and conclusions:

- Accuracy on the training set (2096 samples) = 98.3%
- Accuracy on the test set (524 samples) = 97.3%
- Heidke Skill Score (HSS) = 96.2%
- Comparing with the well-known kNN & the very competitive SVM classification methods: kNN ($k = 5$) = 57.5%, SVM = 88.1%, ConvNet gives the best results achieving the highest accuracy.
- This new methodology could be applied to investigate:
 1. other frequency ranges (Pc1/EMIC, Pc2, Pc4, Pc5)
 2. observations from other satellite missions
 3. ground-based observations.

Author Contributions: Conceptualization, G.B.; methodology, A.A. and C.P.; software, A.A.; investigation, A.A., C.P. and A.Z.B.; writing—original draft preparation, A.A., G.B., C.P., K.K. and I.A.D.; writing—review and editing, A.R., I.A.D., O.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study has been supported as part of Swarm DISC (Data, Innovation, and Science Cluster) activities, funded by ESA contract no. 4000109587.

Data Availability Statement: The results presented rely on the data collected by the three satellites of the Swarm constellation. Swarm data can be accessed at <https://swarm-diss.eo.esa.int/> (accessed on 27 January 2022).

Acknowledgments: We thank the referees for the very valuable technical comments that helped us to improve the overall quality of our paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. McPherron, R.L. Magnetic Pulsations: Their Sources and Relation to Solar Wind and Geomagnetic Activity. *Surv. Geophys.* **2005**, *26*, 545–592. [\[CrossRef\]](#)
2. Jacobs, J.A.; Kato, Y.; Matsushita, S.; Troitskaya, V.A. Classification of geomagnetic micropulsations. *J. Geophys. Res.* **1964**, *69*, 180–181. [\[CrossRef\]](#)
3. Mann, I.R. Waves, particles, and storms in geospace: An introduction. In *Waves, Particles, and Storms in Geospace*; Balasis, G., Daglis, I.A., Mann, I.R., Eds.; Oxford University Press: Oxford, UK, 2016; pp. 1–14.
4. Daglis, I.A.; Katsavrias, C.; Georgiou, M. From solar sneezing to killer electrons: Outer radiation belt response to solar eruptions. *Philos. Trans. R. Soc. A* **2019**, *377*, 20180097. [\[CrossRef\]](#)
5. Balasis, G.; Daglis, I.A.; Georgiou, M.; Papadimitriou, C.; Haagmans, R. Magnetospheric ULF wave studies in the frame of Swarm mission: A time-frequency analysis tool for automated detection of pulsations in magnetic and electric field observations. *Earth Planets Space* **2013**, *65*, 18. [\[CrossRef\]](#)
6. Balasis, G.; Aminalragia-Giamini, S.; Papadimitriou, C.; Daglis, I.A.; Anastasiadis, A.; Haagmans, R. A machine learning approach for automated ULF wave recognition. *J. Space Weather Space Clim.* **2019**, *9*, A13. [\[CrossRef\]](#)
7. Balasis, G.; Papadimitriou, C.; Daglis, I.A.; Pilipenko, V. ULF wave power features in the topside ionosphere revealed by Swarm observations. *Geophys. Res. Lett.* **2015**, *42*, 6922–6930. [\[CrossRef\]](#)
8. Olsen, N.; Finlay, C.C.; Kotsiaros, S.; Tøffner-Clausen, L. A model of Earth's magnetic field derived from 2 years of Swarm satellite constellation data. *Earth Planets Space* **2016**, *68*, 124. [\[CrossRef\]](#)

9. Leger, J.-M.; Bertrand, F.; Jager, T.; Le Prado, M.; Fratter, I.; Lalaurie, J.-C. Swarm Absolute Scalar and Vector Magnetometer Based on Helium 4 Optical Pumping. *Procedia Chem.* **2009**, *1*, 634–637. [[CrossRef](#)]
10. Stolle, C.; Lühr, H.; Rother, M.; Balasis, G. Magnetic signatures of equatorial spread F as observed by the CHAMP satellite. *J. Geophys. Res.* **2006**, *111*, A02304. [[CrossRef](#)]
11. Park, J.; Stolle, C.; Noja, M.; Stolle, C.; Lühr, H. The Ionospheric Bubble Index deduced from magnetic field and plasma observations onboard Swarm. *Earth Planet Space* **2013**, *65*, 13. [[CrossRef](#)]
12. Baker, D.N. Linear prediction filter analysis of relativistic electron properties at 6.6 RE. *J. Geophys. Res. Space Phys.* **1990**, *95*, 15133–15140. [[CrossRef](#)]
13. Valdivia, J.A.; Sharma, A.S.; Papadopoulos, K. Prediction of magnetic storms by nonlinear models. *Geophys. Res. Lett.* **1996**, *23*, 2899–2902. [[CrossRef](#)]
14. Sutcliffe, P.R. Substorm onset identification using neural networks and Pi2 pulsations. *Ann. Geophys.* **1997**, *15*, 1257–1264. [[CrossRef](#)]
15. Lundstedt, H. AI techniques in geomagnetic storm forecasting. In *Magnetic Storms*; Tsurutani, B.T., Gonzalez, W.D., Kamide, Y., Arballo, J.K., Eds.; American Geophysical Union: Washington, DC, USA, 1997.
16. Lundstedt, H. Progress in space weather predictions and applications. *Adv. Space Res.* **2005**, *36*, 2516–2523. [[CrossRef](#)]
17. Boberg, F.; Wintoft, P.; Lundstedt, H. Real time Kp predictions from solar wind data using neural networks. *Phys. Chem. Earth Part C* **2000**, *25*, 275–280. [[CrossRef](#)]
18. Vassiliadis, D. System identification, modeling, and prediction for space weather environments. *IEEE Trans. Plasma Sci.* **2000**, *28*, 1944–1955. [[CrossRef](#)]
19. Gleisner, H.; Lundstedt, H. A neural network-based local model for prediction of geomagnetic disturbances. *J. Geophys. Res. Space Phys.* **2001**, *106*, 8425–8433. [[CrossRef](#)]
20. Li, X. Quantitative prediction of radiation belt electrons at geostationary orbit based on solar wind measurements. *Geophys. Res. Lett.* **2001**, *28*, 1887–1890. [[CrossRef](#)]
21. Vandegriff, J. Forecasting space weather: Predicting interplanetary shocks using neural networks. *Adv. Space Res.* **2005**, *36*, 2323–2327. [[CrossRef](#)]
22. Wing, S.; Johnson, J.; Jen, J.; Meng, C.-I.; Sibeck, D.; Bechtold, K.; Carr, S.; Costello, K.; Freeman, J.; Balikhin, M.; Vandegriff, J. Kp forecast models. *J. Geophys. Res.* **2005**, *110*, A04203. [[CrossRef](#)]
23. Cesaroni, C.; Spogli, L.; Aragon-Angel, A.; Fiocca, M.; Dear, V.; De Franceschi, G.; Romano, V. Neural network based model for global Total Electron Content forecasting. *J. Space Weather Space Clim.* **2020**, *10*, 11. [[CrossRef](#)]
24. Park, W.; Lee, J.; Kim, K.-C.; Lee, J.; Park, K.; Miyashita, Y.; Sohn, J.; Park, J.; Kwak, Y.; Hwang, J.; et al. Operational Dst index prediction model based on combination of artificial neural network and empirical model. *J. Space Weather Space Clim.* **2021**, *11*, 38. [[CrossRef](#)]
25. Chakraborty, S.; Morley, S. Probabilistic prediction of geomagnetic storms and the Kp index. *J. Space Weather Space Clim.* **2020**, *10*, 36. [[CrossRef](#)]
26. Arregui, I. Recent Applications of Bayesian Methods to the Solar Corona. *Front. Astron. Space Sci.* **2022**, *9*, 826947. [[CrossRef](#)]
27. Georgoulis, M.K.; Bloomfield, D.S.; Piana, M.; Massone, A.M.; Soldati, M.; Gallagher, P.T.; Pariat, E.; Vilmer, N.; Buchlin, E.; Baudin, F.; et al. The flare likelihood and region eruption forecasting (FLARECAST) project: Flare forecasting in the big data & machine learning era. *J. Space Weather Space Clim.* **2021**, *11*, 39. [[CrossRef](#)]
28. Lavasa, E.; Giannopoulos, G.; Papaioannou, A.; Anastasiadis, A.; Daglis, I.A.; Aran, A.; Pacheco, D.; Sanahuja, B. Assessing the Predictability of Solar Energetic Particles with the Use of Machine Learning Techniques. *Sol. Phys.* **2021**, *296*, 107. [[CrossRef](#)]
29. Aminalragia-Giamini, S.; Raptis, S.; Anastasiadis, A.; Tsigkanos, A.; Sandberg, I.; Papaioannou, A.; Papadimitriou, C.; Jiggins, P.; Aran, A.; Daglis, I.A. Solar Energetic Particle Event occurrence prediction using Solar Flare Soft X-ray measurements and Machine Learning. *J. Space Weather Space Clim.* **2021**, *11*, 59. [[CrossRef](#)]
30. Blandin, M.; Connor, H.K.; Öztürk, D.S.; Keese, A.M.; Pinto, V.; Mahmud, M.S.; Ngwira, C.; Priyadarshi, S. Multi-Variate LSTM Prediction of Alaska Magnetometer Chain Utilizing a Coupled Model Approach. *Front. Astron. Space Sci.* **2022**, *9*, 846291. [[CrossRef](#)]
31. Capannolo, L.; Li, W.; Huang, S. Identification and Classification of Relativistic Electron Precipitation at Earth Using Supervised Deep Learning. *Front. Astron. Space Sci.* **2022**, *9*, 858990. [[CrossRef](#)]
32. Pinto, V.A.; Keese, A.M.; Coughlan, M.; Mukundan, R.; Johnson, W.; Ngwira, C.M.; Connor, H.K. Revisiting the Ground Magnetic Field Perturbations Challenge: A Machine Learning Perspective. *Front. Astron. Space Sci.* **2022**, *9*, 869740. [[CrossRef](#)]
33. Yeakel, K.L.; Vandegriff, J.D.; Garton, T.M.; Jackman, C.M.; Clark, G.; Vines, S.K.; Smith, A.W.; Kollmann, P. Classification of Cassini's Orbit Regions as Magnetosphere, Magnetosheath, and Solar Wind via Machine Learning. *Front. Astron. Space Sci.* **2022**, *9*, 875985. [[CrossRef](#)]
34. Camporeale, E.; Wing, S.; Johnson, J. *Machine Learning Techniques for Space Weather*; Elsevier: Amsterdam, The Netherlands, 2008; p. 454. [[CrossRef](#)]
35. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]

36. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In *Proceedings, Advances in Neural Information Processing Systems 25 (NIPS 2012)*; Pereira, F., Burges, C.J., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NY, USA, 2012; pp. 1090–1098.
37. Collobert, R.; Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML 2008)*, Helsinki, Finland, 5–9 June 2008; pp. 160–167.
38. Narock, T.; Narock, A.; Dos Santos, L.F.G.; Nieves-Chinchilla, T. Identification of Flux Rope Orientation via Neural Networks. *Front. Astron. Space Sci.* **2022**, *9*, 838442. [[CrossRef](#)]
39. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
40. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
41. CS231n: Deep Learning for Computer Vision, Stanford, Spring 2022. Available online: <http://cs231n.stanford.edu/> (accessed on 21 May 2022).
42. Alom, M.Z.; Taha, T.; Yakopcic, C.; Westberg, S.; Hasan, M.; Esesn, B.; Awwal, A.; Asari, V. The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. *arXiv* **2018**, arXiv:1803.01164.
43. Theodoridis, S.; Koutroubas, K. *Pattern Recognition*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2003; p. 689, ISBN 0-12-685875-6.
44. Imandoust, S.B.; Bolandraftar, M. Application of K-nearest neighbor (KNN) approach for predicting economic events theoretical background. *Int. J. Eng. Res. Appl.* **2013**, *3*, 605–610.
45. Güvenç, E.; Çetin, G.; Koçak, H. Comparison of KNN and DNN Classifiers Performance in Predicting Mobile Phone Price Ranges. *Adv. Artif. Intell. Res.* **2021**, *1*, 19–28.
46. Cunningha, P.; Delany, S.J. *k-Nearest Neighbour Classifiers*; Technical Report UCD-CSI-2007-4; University College Dublin: Dublin, Ireland, 2007.
47. Cai, Y.; Ji, D.; Cai, D. A KNN Research Paper Classification Method Based on Shared Nearest Neighbor. In *Proceedings of the NTCIR-8 Workshop Meeting*, Tokyo, Japan, 15–18 June 2010.
48. Chamasemani, F.F.; Singh, Y.P. Multi-class Support Vector Machine (SVM) classifiers—An Application in Hypothyroid detection and Classification. In *Proceedings of the Sixth International Conference on Bio-Inspired Computing: Theories and Applications*, Penang, Malaysia, 27–29 September 2011. [[CrossRef](#)]
49. Applied Machine Learning—INFO-4604. University of Colorado Boulder. 2018. Available online: <https://cmci.colorado.edu/classes/INFO-4604/> (accessed on 24 June 2022).
50. Haasdonk, B. Feature Space Interpretation of SVMs with Indefinite Kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 4. [[CrossRef](#)]
51. Finlay, C.C.; Olsen, N.; Kotsiaros, S.; Gillet, N.; Tøffner-Clausen, L. Recent geomagnetic secular variation from Swarm and ground observatories as estimated in the CHAOS-6 geomagnetic field model. *Earth Planet Space* **2016**, *68*, 112. [[CrossRef](#)]
52. Ritter, P.; Lühr, H.; Rauberg, J. Determining field-aligned currents with the Swarm constellation mission. *Earth Planet Space* **2013**, *65*, 1285–1294. [[CrossRef](#)]
53. Swarm Data Access. Available online: <https://swarm-diss.eo.esa.int/> (accessed on 27 May 2022).
54. Papadimitriou, C.; Balasis, G.; Daglis, I.A.; Giannakis, O. An initial ULF wave index derived from 2 years of Swarm observations. *Ann. Geophys.* **2018**, *36*, 287–299. [[CrossRef](#)]
55. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:207.0580v1.
56. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, CA, USA, 7–9 May 2015.
57. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010; pp. 249–256. Available online: <https://proceedings.mlr.press/v9/glorot10a.html> (accessed on 14 July 2022).
58. Nwankpa, C.E.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *arXiv* **2018**, arXiv:1811.03378v1. <https://doi.org/10.48550/arXiv.1811.03378>.
59. Good, I.J. Rational Decisions. *J. R. Stat. Soc. Ser. B* **1952**, *14*, 107–114. Available online: www.jstor.org/stable/2984087 (accessed on 7 September 2022). [[CrossRef](#)]
60. Papadimitriou, C.; Balasis, B.; Boutsis, A. Z.; Daglis, I. A.; Giannakis, O.; Anastasiadis, A.; De Michelis, P.; Consolini, G. Dynamical Complexity of the 2015 St. Patrick’s Day Magnetic Storm at Swarm Altitudes Using Entropy Measures. *Entropy* **2020**, *22*, 574. [[CrossRef](#)]
61. Balasis, G.; Papadimitriou, C.; Boutsis, A.Z.; Daglis, I.A.; Giannakis, O.; Anastasiadis, A.; De Michelis, P.; Consolini, G. Dynamical complexity in Swarm electron density time series using Block entropy. *EPL* **2020**, *131*, 69001. [[CrossRef](#)]
62. De Michelis, P.; Pignalberi, A.; Consolini, G.; Coco, I.; Tozzi, R.; Pezzopane, M.; Giannattasio, F.; Balasis, G. On the 2015 St. Patrick’s Storm Turbulent State of the Ionosphere: Hints From the Swarm Mission. *J. Geophys. Res. Space Phys.* **2020**, *125*, e2020JA027934. [[CrossRef](#)]

63. Balasis, G.; Daglis, I. A.; Contoyiannis, Y.; Potirakis, S. M.; Papadimitriou, C.; Melis, N. S.; Giannakis, O.; Papaioannou, A.; Anastasiadis, A.; Kontoes, C. Observation of intermittency-induced critical dynamics in geomagnetic field time series prior to the intense magnetic storms of March, June, and December 2015. *J. Geophys. Res. Space Phys.* **2018**, *123*, 4594–4613. [[CrossRef](#)]
64. Forecast Verification Methods Across Time and Space Scales—Heidke Skill Score (Cohen’s k). In Proceedings of the 7th International Verification Methods Workshop, Berlin, Germany, 8–11 May 2017. Available online: <https://cawcr.gov.au/projects/verification/> (accessed on 24 May 2022).
65. Tsagouri, I.; Borries, C.; Perry, C.; Dierckxsens, M.; Georgoulis, M.; Bloomfield, D.S. *Guidelines for Common Validation in the SSA SWE Network*; Technical Note ssa-swe-escdef-tn-5401; European Space Agency: Paris, France, 2019.
66. Goutte, C.; Gaussier, E. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. *Adv. Inf. Retr.* **2005**, *3408*, 345–359. [[CrossRef](#)]
67. Raghavan, V.; Bollmann, P.; Jung, G.S. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.* **1989**, *7*, 205–229. [[CrossRef](#)]