QUANTILE NEURAL NETWORKS
— FORECASTING EXTREME EVENTS —

A Disturbance Storm Time index model to describe and forecast its
density function conditioned on some geomagetic data, using
Quantile Neural Networks

Arnau Perich Iglesias

Bachelor's Final Project – Mathematics Degree

Universitat Autònoma de Barcelona

Supervisor: Isabel Serra Mochales

June 2024

UAB
Universitat Autònoma
de Barcelona

Abstract

Solar storms are disturbances of the Earth's magnetic field measured by the *Disturbance Storm Time* (*Dst*) index, and they can cause significant damage to electrical and telecommunication devices. For this reason, precise monitoring of these storms is crucial in a society as technologically advanced as the one we live in. However, there are still gaps in understanding the behavior of these extreme events. This project proposes a model that combines linear regression, neural networks, and the concept of quantile regression to not only estimate the severity of solar storms but also to describe and predict the entire distribution of the *Dst* index conditioned on various geomagnetic data. To achieve this, the project mathematically and rigorously formulates the mentioned models and provides a perspective different from that found in the literature to understand the relationship between them and introduce the concept of quantile regression focused on neural networks. The results of applying this model give, on the one hand, a detailed description of the left tail of the distribution to make inferences about extreme events and, on the other hand, provide a general model for the entire conditional distribution of the Dst.

Preface

This work is the result of persistence and work capacity carried out over all these years, along with the more creative side that has always characterized me and motivated me to find new ways to face problems. The document captures all this essence that characterizes me: it combines mathematical rigor with the practicality of data science, together with a different perspective on addressing the problem, to provide a creative solution to a real problem.

However, before delving into this work, I would like to thank my tutor for guiding me in this world that I previously knew little about. And to all the people who have shared moments with me all these academic years, allowing me to continue having the same eagerness to explore and keep learning, just as I did when I was a child.

Contents

Introduction

Statistics is the connection between mathematical rigor and data science practicality, which has gained significant importance nowadays. These two characteristics make this field a very powerful tool for adding value to real data. In this project, this idea has been leveraged with the objective of describing and predicting the behaviour of solar storms using statistical and deep learning models.

In spite of the fact that a solar storm could be the responsible of damaging electric and electronic devices, satellites, telecommunication tools and other systems that rely on the Earth's magnetic field as a point of reference [1], there is currently a lack of knowledge regarding the description and forecasting of such events, especially the most extreme ones [2]. In this document a solution to this problematic is proposed. For this reason, the mathematical basis of the models has been studied so as to implement them for analysing the *Disturbance Storm Time* (*Dst*) index: an index that measures the intensity of the geomagnetic storms [3].

The main objective of this work, therefore, is to design and implement a model to describe the *Dst* in more detail. To achieve this goal, the following steps have been proposed:

- Define rigorously the two models that will be useful to predict and describe the *Dst* index: the linear model and the neural networks.

- Give special emphasis in the hypotheses of each model and understand the similarities between them.

- Define the idea of quantile regression and provide some own examples in order to understand its advantages.

- Design and implement a model to predict and describe the *Dst* distribution and density function conditioned on some known data.

In the first chapter, a precise mathematical definition of the linear model is provided, exemplifying the case where the data are normal, independent and identically distributed. The second chapter focuses on defining the most important and powerful deep learning model: neural networks. Additionally, a different perspective from the conventional one is given to relate it to the linear model. Next, in the third chapter, quantile regression is presented: an idea that goes further and uses the previous models not only to describe a single value but to define the entire distribution. Finally, in the last chapter, the implemented model is applied to a real data set to provide a more detailed description and prediction of solar storm occurrences.

Chapter 1

Linear Regression
The simplest statistical model

A statistical model is a non-deterministic mathematical model. That is, some of the model parameters do not have a specific value, but rather have an associated probability distribution; it has stochastic variables. Formally [4]:

Definition 1 (Statistical model)**.** *A statistical model is defined as the pair $(\mathcal{S}, \mathcal{P})$, where $\mathcal{S}$ is the set of possible observations, i.e., sample space, and $\mathcal{P}$ is the set of probability distributions of $\mathcal{S}$.*

Thanks to this generalization, we can find the parameter values of any parametric statistical model by maximizing the likelihood function, $\mathcal{L}$. With this process, we find the values that make the observed data, $\mathcal{S}$, most likely, following one of the distributions in $\mathcal{P}$.

The following chapter is focused on explaining the basis of the simplest statistical model: the linear regression. The main definitions have been extracted from *Linear Models in Statistics* book [5] and, then, some deductions have been done.

Definition 2. *Let $Y \in \mathbb{R}$ and $X \in \mathbb{R}^m$ be random variables. We define the input set and the output set, respectively, as*

$$\boldsymbol{x} := [x_1, \ldots, x_N], \qquad \boldsymbol{y} := [y_1, \ldots, y_N],$$

where x_i and y_i with $i = 1, \ldots, N$ are a realization of X and Y.

Linear regression is a statistical model that assumes the response variable is a linear combination of the predictors, as shown in the expression

$$Y \sim X\beta + \varepsilon,$$

where $\varepsilon \sim P$ is a random variable representing the error term. It is assumed to have zero mean, constant variance and be independent. In this way, we can describe the conditional distribution: $Y|X \sim P_\theta$, where θ represents the model parameters.

The model is based on describing a measure of $Y|X$ as a linear function, F, of the predictors

$$F(X, \beta) := X\beta = \beta_0 + \beta_0 X_1 + \cdots + \beta_m X_m.$$

Finally, the parameter values are adjusted by maximizing the likelihood function

$$\mathcal{L}(\beta; y, x) = \mathbb{P}(Y \mid X = x, \beta).$$

Due to the *Central Limit Theorem* [6], there are many cases where it is reasonable to think that the distribution is normal. This is why the most common example is to assume $\varepsilon \sim \mathcal{N}(0, \sigma)$. In this case, $Y|X \sim \mathcal{N}(F(X, \beta), \sigma)$ and, due to the independence assumption, the likelihood function to maximize is

$$\mathcal{L}(\beta; y, x) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(y_i - F(x_i, \beta))^2}{2\sigma^2} \right\}.$$

Although the *log-likelihood* is usually used

$$l(\beta; y, x) \propto N \log \sigma^2 + \frac{1}{\sigma^2} \sum_{i=1}^{N} -(y_i - F(x_i, \beta))^2.$$

This last expression allows us to recover the usual notion of using the residual sum of squares (RSS) as a function to minimize. Finally, as we have mentioned, a statistical model is based on describing a measure of $Y|X$. In the example we were showing, theoretically, we are trying to minimize the error variance

$$\mathbb{E}(\varepsilon^2) = \mathbb{E}[(Y - F(X, \beta))^2] = \mathbb{E}[(Y + \mathbb{E}[Y|X] - \mathbb{E}[Y|X] - F(X, \beta))^2] =$$
$$\mathbb{E}[(Y - \mathbb{E}[Y|X])^2] + \mathbb{E}[(\mathbb{E}[Y|X] - F(X, \beta))^2] + 2\mathbb{E}[(Y - \mathbb{E}[Y|X])(\mathbb{E}[Y|X] - F(X, \beta))].$$

Given that $\mathbb{E}[Y|X]$ can be thought of as the projection of Y onto the subspace generated by X, then $(Y - \mathbb{E}[Y|X])$ is orthogonal to X, and to $(\mathbb{E}[Y|X] - F(X, \beta))$ since $F(X, \beta)$ is in the subspace generated by X. Therefore, we obtain

$$\mathbb{E}[(Y - \mathbb{E}[Y|X])^2] + \mathbb{E}[(\mathbb{E}[Y|X] - F(X, \beta))^2].$$

And we can conclude that the best choice for F is the conditional expectation: $F(X, \beta) = \mathbb{E}[Y|X]$. Therefore, in this case, we are describing the conditional expectation as a linear function of the predictors.

Thanks to the assumption of the distribution and the simple structure of the linear model, we can make inferences about the coefficients β, the variance σ and the variable Y. Although the following calculations can be made with the general case, to delve into theoretical results analytically, we will reduce to the case of simple linear regression ($m = 1$). The expression is

$$Y \sim \beta_0 + \beta_1 X + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, \sigma)$.

1.1   Estimation

As it has been explained, the coefficients of the regression can be estimated using the likelihood function. The following results show this.

Preposition 1. *Assume normally distributed errors. Then, the maximum likelihood estimators are*

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \qquad \hat{\beta}_1 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}, \qquad \hat{\sigma}^2 = \frac{\sum_{i=1}^{N}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2}{N - 1}.$$

Proof. The expression for the *log-likelihood* is

$$l(\beta; y, x) \propto N \log \sigma^2 + \frac{1}{\sigma^2} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 x_i))^2.$$

To obtain the parameter values that maximize the above expression, we differentiate with respect to them and set to zero

$$\frac{\partial l}{\partial \beta_0} \propto \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 x_i))^2 \implies \bar{y} = \beta_0 + \beta_1 \bar{x}.$$

$$\frac{\partial l}{\partial \beta_1} \propto \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 x_i)) x_i \implies \sum_{i=1}^{N} y_i x_i = \beta_0 \sum_{i=1}^{N} x_i + \beta_1 \sum_{i=1}^{N} x_i^2.$$

$$\frac{\partial l}{\partial \sigma^2} \propto \frac{N}{\sigma^2} - \frac{1}{\sigma^4} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 x_i))^2 \implies N - \frac{1}{\sigma^2} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 x_i))^2 = 0.$$

With the first two equations, we obtain a linear system that allows us to find the values of $\hat{\beta}_0$ and $\hat{\beta}_1$ *. On the other hand, the last equation leads us to an expression similar to the one shown in the statement; note that the derived expression has N as the denominator, instead of $N - 1$. It is simply usually taken as $N - 1$ to obtain an unbiased estimator [7]. Finally, it can be verified that these are absolute maxima with the second derivative and the fact that there are no other relative extrema. □

Preposition 2. *The maximum likelihood estimators are unbiased and have variance*

$$\mathbb{E}[\hat{\beta}_0] = \beta_0, \qquad Var(\hat{\beta}_0) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum (x_i - \bar{x})^2} \right).$$

$$\mathbb{E}[\hat{\beta}_1] = \beta_1, \qquad Var(\hat{\beta}_1) = \frac{\sigma^2}{\sum (x_i - \bar{x})^2}.$$

Proof. Following the assumption of the linear model, we can write $y_i \sim \beta_0 + \beta_1 x_i + \varepsilon_i$. Then, from the expressions of preposition 1 and the assumptions about ε_i, we obtain the following results.

For the estimator of β_1 we have that

$$\mathbb{E}[\hat{\beta}_1] = \mathbb{E} \left[\frac{\sum_{i=1}^{N} (x_i - \bar{x})(\beta_0 + \beta_1 x_i + \varepsilon_i - (\beta_0 + \beta_1 \bar{x} + \bar{\varepsilon}_i))}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \right]$$

$$= \beta_1 + \mathbb{E} \left[\frac{\sum_{i=1}^{N} (x_i - \bar{x}) \varepsilon_i}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \right] = \beta_1.$$

$$Var(\hat{\beta}_1) = Var \left(\beta_1 + \frac{\sum_{i=1}^{N} (x_i - \bar{x}) \varepsilon_i}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \right) = \sum_{i=1}^{N} Var \left(\frac{(x_i - \bar{x}) \varepsilon_i}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \right)$$

$$= \frac{\sum_{i=1}^{N} (x_i - \bar{x})^2 \sigma^2}{(\sum_{i=1}^{N} (x_i - \bar{x})^2)^2} = \frac{\sigma^2}{\sum_{i=1}^{N} (x_i - \bar{x})^2}.$$

---

*The system of equations formed by the first two expressions is called *normal equations*.

For the estimator of β_0 we have that

$$\mathbb{E}[\hat{\beta}_0] = \mathbb{E}[\beta_0 + \beta_1\bar{x} + \bar{\varepsilon}_i - \hat{\beta}_1\bar{x}] = \beta_0.$$

$$\mathrm{Var}(\hat{\beta}_0) = \mathrm{Var}(\beta_0 + \beta_1\bar{x} + \bar{\varepsilon}_i - \hat{\beta}_1\bar{x}) = \mathrm{Var}(\bar{\varepsilon}_i) + \mathrm{Var}(\hat{\beta}_1\bar{x}) = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}.$$

$\square$

Preposition 3. *The asymptotic distribution of the coefficients is a normal distribution with mean and variance the same as that of the estimator. Specifically, we have that*

$$\hat{\beta}_0 \overset{a}{\sim} \mathcal{N}\left(\beta_0, \sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2}\right)\right), \qquad \hat{\beta}_1 \overset{a}{\sim} \mathcal{N}\left(\beta_1, \frac{\sigma^2}{\sum(x_i - \bar{x})^2}\right).$$

Proof. Given a parameter θ and $\hat{\theta}$ its maximum likelihood estimator, Wilks' Theorem [8] ensures the asymptotic convergence of the maximum likelihood estimators to a normal distribution with the same mean and variance as that of the estimator. That is to say,

$$\hat{\theta} \overset{a}{\sim} \mathcal{N}(\mathbb{E}[\hat{\theta}], \mathrm{Var}(\hat{\theta})).$$

Therefore, by preposition 2, we obtain the result stated. $\square$

1.2   Inference

With these results, we can provide additional information about both the estimated coefficients and the obtained values. Below, each of these results is detailed.

1.2.1   Confidence Interval for Estimators

Using the results from preposition 3 and substituting σ for the estimation from preposition 1, we obtain the statistic

$$t = \frac{\hat{\beta}_i - \beta_{i_0}}{\hat{\mathrm{Var}}(\hat{\beta}_i)} \sim \text{t-Student}_{n-2}.$$

Thanks to this result, on the one hand, the confidence intervals with a level of $1 - \alpha$ are

$$\left[\beta_0 \pm t_{n-2,\alpha/2}\hat{\sigma}\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2}}\right], \qquad \left[\beta_1 \pm t_{n-2,\alpha/2}\hat{\sigma}\sqrt{\frac{1}{\sum(x_i - \bar{x})^2}}\right].$$

On the other hand, we can design the test $\mathcal{H}_0 : \beta_i = \beta_{i_0}$ vs $\mathcal{H}_1 : \beta_i \neq \beta_{i_0}$ to prove whether there is significant evidence to say that the coefficient is different from β_{i_0}. This test is especially important in the case of the coefficient β_1 and $\beta_{1_0} = 0$, as it allows us to see if X has an effect on the response variable Y.

1.2.2   Confidence Interval for $\mathbb{E}(Y|X = x_0)$

According to the linear model, the conditional mean is described as a linear combination of the predictors

$$\mathbb{E}[Y \mid X = x_0] = \beta_0 + \beta_1 X.$$

Therefore, the estimate of this value is

$$\hat{\beta}_0 + \hat{\beta}_1 x_0.$$

By preposition 2 and 3, the estimator is normal with mean and variance

$$\mathbb{E}[\hat{\beta}_0 + \hat{\beta}_1 x_0] = \beta_0 + \beta_1 x_0,$$

$$\text{Var}(\hat{\beta}_0 + \hat{\beta}_1 x_0) = \text{Var}(\hat{\beta}_0) + \text{Var}(\hat{\beta}_1 x_0) = \sigma^2 \left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum (x_i - \bar{x})^2} \right).$$

Therefore, following the same argument as for the previous confidence interval, the statistic has the same form and the confidence interval with a level of $1 - \alpha$ is

$$\left[\hat{\beta}_0 + \hat{\beta}_1 x_0 \pm t_{n-2,\alpha/2} \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum (x_i - \bar{x})^2}} \right].$$

1.2.3   Confidence Interval for $Y | X = x_0$

According to the linear model, the prediction of Y given a value of X is modeled as

$$y_0 \sim \beta_0 + \beta_1 x_0 + \epsilon.$$

The prediction of y_0 is

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_0.$$

Similarly as before, the estimator is normal with the same mean but different variance. In this case, the prediction error is the difference between the value that the random variable Y can take and the fitted value $\hat{y}_0$. Therefore, we have

$$\mathbb{E}[Y - \hat{y}_0] = 0,$$

$$\text{Var}(Y - \hat{y}_0) = \text{Var}(Y) + \text{Var}(\hat{y}_0) = \sigma^2 \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum (x_i - \bar{x})^2} \right).$$

Thus, we have that the confidence interval with a level of $1 - \alpha$ is

$$\left[\hat{\beta}_0 + \hat{\beta}_1 x_0 \pm t_{n-2,\alpha/2} \hat{\sigma} \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum (x_i - \bar{x})^2}} \right].$$

Notice that the main difference between both results is that the prediction interval is wider. This result is logical since there is more uncertainty when predicting a new value than when calculating its mean.

Chapter 2

Neural Networks
A different perspective to define them

The general idea of a Neural Network is based on approximating an unknown function that models the relationship between a given data set. The structure always consists of an input layer, an output layer and at least one hidden layer. Each layer is composed of different units, called neurons. These are connected between the different layers through weights, and their function is to give more importance to relevant features. The value of these coefficients can be determined by minimizing a loss function or by thinking from a probabilistic perspective to maximize a likelihood.

Next, a mathematical formulation of the basics of neural networks is given, and an unusual viewpoint is presented to relate it to linear models. Mainly, the content explained in the *An Introduction to Statistical Learning* book [9] and the article [10] has been followed, as well as own contributions has been provided in order to explain this different perspective.

2.1   Mathematical Formulation

We recall the initial definition already presented in Chapter 1.

Definition 3. *Let $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}^m$ with $i = 1, \ldots, N$. We define the input set and output set of size N, respectively, as*

$$\boldsymbol{x} \coloneqq [x_1, \ldots, x_N], \qquad \boldsymbol{y} \coloneqq [y_1, \ldots, y_N].$$

It is worth noting that, to avoid adding more notation, sometimes we will refer to X and Y as the random variables from which the above sets are drawn.

Definition 4. *The function $f : \mathbb{R}^n \to \mathbb{R}^m$ represents the true relationship of $\boldsymbol{y}$ with $\boldsymbol{x}$. On the other hand, the function $F : \mathbb{R}^n \to \mathbb{R}^m$ represents the network activity.*

In the context of neural networks, the network activity is a function that relates the input values to the output values and is made up of the connection of a set of values through weights. These values are called *neurons* and are grouped by *layers*. The structure of F will be determined by the number of neurons, the number of layers and how these neurons are connected. All of this is called the *network architecture*.

To understand this network activity, it is essential to define each of these parts.

2.1.1   Network Architecture

Neurons are the basic unit of a neural network. Each neuron receives a signal $s = (s_1, \ldots, s_k)$ from $k \in \mathbb{N}$ inputs (which will depend on the previous choice) and a vector modeling the weights w of the connection between the previous neurons and the current one. A function h interacts with the inputs, weights and an activation function g to generate a 1-dimensional output. There are different ways in which neurons process these inputs, although we can distinguish between the two main types:

- **Neurons representing a dot product.** One way to handle the signal s that the neuron receives is through the dot product between the inputs and the weight vector w. Therefore, the neuron's activity can be described as

$$h(s) = g(\langle s, w \rangle),$$

  where $\langle \cdot, \cdot \rangle$ represents the dot product.

- **Neurons that compute a distance.** On the other hand, neurons can also compute a distance between s and w. In this case, the activity would be described by

$$h(s) = g(||w - s||),$$

  where $|| \cdot ||$ represents a norm.

One of the critical points when designing the neural network to perform at its best level is to correctly choose the activation function. This choice will depend on our data set, the objective to be achieved, etc.

Definition 5. *The most common activation functions are:*

Sigmoid

$$g(z) = \frac{1}{1 + e^{-z}}.$$

Hyperbolic tangent

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

Rectified Linear Unit (ReLU)

$$g(z) = \max(0, z).$$

Softplus

$$g(z) = \ln(1 + e^z).$$

A very important characteristic of these functions is non-linearity as it allows capturing more complex relationships and interactions. Moreover, if we used a linear function, we would end up with a linear model. In the early days of this area, the *sigmoid* function was generally used as the preferred activation function. Currently, the preferred activation function is *ReLU*. This function can be computed and stored more efficiently than a

sigmoid. Furthermore, although the threshold is zero, because it is applied to a linear function, the constant term w_{k0} will shift this inflection point. Therefore, it is a more complex function than it seems and yields great results.

Each processing unit, the neurons, is organized into layers. Every neural network has a first layer that contains all n input variables and a last layer that contains the response. The number of neurons in this will depend on the objective: typically, for regression problems, we will only have $m = 1$ neurons; in the case of a categorical variable with m classes, we will have one neuron for each class. Additionally, every neural network has at least one hidden layer. These hidden layers contain neurons that perform the process explained before.

The activity of a neural network mainly depends on the choice of the number of neurons, layers, activation function and weights for each neuron. But also on how the neurons and layers are connected. The architecture of a neural network is the choice of all these characteristics. The most common neural network architectures are:

- **Feedforward Neural Network (FNN)**: it is the simplest neural network. It consists of one or more hidden layers. The neurons in each layer are connected to the neurons in the next layer, so no cycles are created. Therefore, as its name suggests, information moves forward through the neurons of each layer, sequentially and orderly. The neurons used are of the first type, with the dot product as the operation.

- **Convolutional Neural Network (CNN)**: it is used, especially for image classification, object detection and image segmentation.

- **Recurrent Neural Network (RNN)**: it is designed to process structured data, where the order of the input data matters. The data input and calculation of the hidden layer neurons are done sequentially, allowing interaction of neurons and input data at all times. It is used for tasks such as speech recognition, text translation and time series processing.

2.1.2 Training the Network

Our goal is to train the network by adjusting the weights of the function F to best approximate f. This problem can be approached in two different ways:

- **Functional perspective**. The usual approach is based on minimizing a *risk function* R that depends on the chosen model F and is computed by summing over a *loss function* L. That is

$$R(F) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, F(x_i)).$$

- **Statistical perspective**. A new and own point of view is presented. Note that, for any architecture, the function F representing the network activity satisfies

$$Y = F(X) + \varepsilon,$$

where ε represents the error term. If we assume that the data come from random variables, then $\varepsilon \sim P$ is another random variable. Thus, we can formulate the optimization problem with a likelihood. In the same way as in the linear model, we have $Y|X \sim P_\theta$ and seek to maximize

$$\mathcal{L}(\theta; x, y) = \mathbb{P}(Y = y \mid X = x, \theta).$$

Recall that when considering the case where $\varepsilon \sim \mathcal{N}(0, \sigma)$, we are modeling the expectation, meaning the best choice for F is the conditional expectation: $F(x) = \mathbb{E}[Y|X = x]$.

The relationship between the two ways of formulating the problem is that most likelihood functions (or log-likelihood) end up simplifying to minimize L. Therefore, without going into further detail, most times we talk about a loss function, it can be understood as an expression derived from the likelihood of some distribution. To give an example, let be $m = 1$. Choosing the residual sum of squares is the same as considering the error ε to be normal distributed. With this assumption, $Y|X \sim \mathcal{N}(F(X), \sigma)$ and the expression to minimize of the variance deduced from maximizing the likelihood function is

$$\hat{\sigma}^2 \propto \sum_{i=1}^{N} (y_i - F(x_i))^2,$$

the same as the residual sum of squares function.

Finally, since, given a sample, an architecture and a loss function, the estimator only depends on the choice of weights but the complexity of F is very high, the problem of training a neural network can be reduced to the numerical method

$$W^* = \arg \min_{W} \frac{1}{N} \sum_{i=1}^{N} L(y_i - F(x_i; W)),$$

where W is the matrix that models all the weights of the entire network. One of the most common ideas to minimize this function is to iteratively make small changes to W in the direction that achieves the greatest improvement; this matches with the direction in which the gradient vector points. The t-th iteration is

$$W_{j,k}(t) = W_{j,k}(t-1) + \Delta Wj, k(t),$$

where the gradient $\Delta W_{j,k}(t)$ is calculated as

$$\Delta W_{j,k}(t) = -\delta(t) \frac{\partial (y_i - F(x_i; W))}{\partial W_{j,k}}.$$

This method is called gradient descent, also known as *Backpropagation.*

The following sections focus on modeling mathematically two of the main architectures of a neural network, which will be useful for solving the practical case explained in Chapter 4.

2.2   Feedforward Neural Network

The simplest structure of a neural network consists of an input layer with n neurons, an output layer with a single neuron ($m = 1$; continuous response) and a hidden layer with K neurons. Each neuron is connected to all the neurons in the next layer. Formally, first, the neurons of the first hidden layer, activations A_k for $k = 1, \ldots, K$, are computed as functions h_k of the input data

$$A_k = h_k(x) = g(w_{k0} + \sum_{j=1}^{n} w_{kj} x_j),$$

where g is an activation function. Next, these K activations are used to compute the output layer

$$F(x) = \beta_0 + \sum_{k=1}^{K} \beta_k A_k.$$

As a final result, as we have defined, we obtain a function $F : \mathbb{R}^n \to \mathbb{R}^m$ that only depends on the input data.

Following the previous scheme, we can generalize to the case where we have more than one hidden layer. Let $x \in \mathbf{x}$ be a input vector with n components, L the number of hidden layers, K_l the number of activation neurons in layer $l \in [1, \ldots, L]$, $A_k^{(l)}$ the value of neuron $k \in [1, \ldots, K_l]$ of layer l, and $y \in \mathbf{y}$ the response vector with m components. We start by determining the first hidden layer with the input data, in the same way as before. We have

$$A_k^{(1)} = h_k^{(1)}(x) = g(w_{k0}^{(1)} + \sum_{j=1}^{n} w_{kj}^{(1)} x_j),$$

for $k = 1, \ldots, K_1$. Next, for each hidden layer $l = 2, \ldots, L$, we determine the activations

$$A_k^{(l)} = h_k^{(l)}(A^{(l-1)}) = g(w_{k0}^{(l)} + \sum_{j=1}^{K_{l-1}} w_{kj}^{(l)} A_j^{(l-1)}),$$

for $k = 1, \ldots, K_l$. Note that, although $h_k^{(l)}$ is shown as a function of the previous activations, it only depends on x; recursively, we reach the input layer. Finally, we compute the values of the neurons in the output layer. If the response variable is continuous, we simply define the function $F(x) = (F_1(x), \ldots, F_m(x))$ with

$$F_k(x) = \beta_{k0} + \sum_{j=1}^{K_L} \beta_{kj} A_j^{(L)},$$

for $k = 1, \ldots, m$. On the other hand, if the response is categorical, we are interested in the output neurons representing the probability of the classes. That is, $F_k(x) = \mathbb{P}(Y = k | x)$. Therefore, in this case, we will use the *softmax* activation function and compute the last neurons as

$$F_k(x) = \frac{e^{[\beta_{k0} + \sum_{j=1}^{K_L} \beta_{kj} A_j^{(L)}]}}{\sum_{j=1}^{m} e^{[\beta_{j0} + \sum_{j'=1}^{K_L} \beta_{jj'} A_{j'}^{(L)}]}}.$$

Finally, the class with the highest probability is assigned as the image of F: $F(x) = \max_k F_k(x)$.

The loss function for neural networks with a continuous response variable commonly used is the *sum of squares*. Therefore, the parameters are chosen to minimize the risk function

$$R(F) = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=1}^{m} (y_{ij} - F_j(xi))^2 \right).$$

On the other hand, for categorical response variables, the multinomial negative log-likelihood, also known as *cross-entropy*, is usually used. Therefore, the risk function to minimize is

$$R(F) = -\frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=1}^{m} y_{ij} \log(F_j(x_i)) \right).$$

2.3   Recurrent Neural Networks

The structure is designed to handle structured data and take advantage of the information provided by this order. Let $x_1, \ldots, x_T$ from the sample $\mathbf{x}$, with $T \leq N$, the input sequence, where $x_t = (x_{t_1}, \ldots, x_{t_n})$. The output vector $y \in \mathbf{y}$ can be either scalar or another sequence. For simplicity, we assume $y \in \mathbb{R}$. The idea is based on recurrently applying a single-layer neural network with K neurons. The values of the neurons obtained in the previous stage, along with the vector x_t, are used as inputs to this same network to update the neuron values. This process is applied recursively until the neurons are updated to the last stage, and thus provide a response.

Formally, the process is shown as follows. We start by calculating the neurons of the first recurrent layer using x_1. That is

$$A_k^{(1)} = h_k^{(1)}(x_1) = g(w_{k0} + \sum_{j=1}^n w_{kj} x_{1_j}),$$

for $k = 1, \ldots, K$. Next, we calculate the response value of stage $t = 1$

$$O_1 = \beta_0 + \sum_{k=1}^K \beta_k A_k^{(1)}.$$

Note that what we are interested in is O_T; that is, the response value corresponding to the last vector x_T. Therefore, for each $t = 2, \ldots, T$, we will recalculate the neuron values using x_t and $A_k^{(t-1)}$ as inputs. We have that

$$A_k^{(t)} = h_k^{(t)}(x_t, A_k^{(t-1)}) = g(w_{k0} + \sum_{j=1}^n w_{kj} x_{1_j} + \sum_{s=1}^K u_{ks} A_s^{(t-1)}),$$

for $k = 1, \ldots, K$. Subsequently, as before, we calculate the response

$$O_t = \beta_0 + \sum_{k=1}^K \beta_k A_k^{(t)}.$$

Note that the idea is based on recurrently applying a *Single Layer Feedforward Neural Network*. This is why the weights do not change in any of the stages; we are applying the same network all the time. Finally, as mentioned, the response value is O_T; we define the function F as

$$F(x_1, \ldots, x_T) = O_T = \beta_0 + \sum_{k=1}^K \beta_k A_k^{(T)}.$$

On the other hand, the reason for calculating the other O_t is that sometimes we are interested in a sequence of size $\bar{T}$ as the response. In these cases, F has the form

$$F(x_1, \ldots, x_T) = (O_{T-\bar{T}}, \ldots, O_T).$$

As mentioned earlier, RNNs are used when working with time series. However, the way the problem is approached is different. Unlike other possible problems that we might face with an RNN, such as using them for text classification, where the sample would consist of several texts (which would translate into different structured data sets), this is different. We only have one sample consisting of a time series with output values, $\{y_t\}$, and another

with input values, $\{x_t\}$. Additionally, the past response values also provide information about the current response value.

To solve this conflict, the idea is based on extracting small sub-series $X = \{X_1, \ldots, X_T\}$ of fixed size T, along with the response variable. These have the form

$$X_1 = \begin{pmatrix} y_{t-T} \\ x_{t-T} \end{pmatrix}, \quad X_2 = \begin{pmatrix} y_{t-T+1} \\ x_{t-T+1} \end{pmatrix}, \quad \ldots, \quad X_L = \begin{pmatrix} y_{t-1} \\ x_{t-1} \end{pmatrix} \quad \text{and} \quad Y = y_t.$$

That is, here the response variable corresponds to the value y_t at the current time, while the structured input set consists of T vectors; each vector is composed of y_t and x_t from previous times. In this way, from a single time series, we can extract different samples to train the network.

Chapter 3

Quantile Regression
The idea of Koenker and Basset

In the previous chapters, various models have been defined and detailed culminating in the structure of *Recurrent Neural Networks*, the model that will help us most effectively predict our data set. Additionally, emphasis has been placed on the different assumptions considered at each moment: good control of these is what allows us to identify the best model for each situation. Finally, examples have been given using the mean squared error as a loss function (one of the most employed methods) under the assumptions of normality, independence, etc. But what happens when these starting hypotheses are not met? The presence of heteroscedasticity, structural changes, or outliers are some of the most common circumstances that lead to such violations [11].

In 1978, Koenker and Basset [12] introduced the well-known *Quantile Regression* as a solution to these problems. Since it is based on an estimation method that minimizes weighted absolute deviations with asymmetric weights, on the one hand, it ensures that the model is not affected by extreme data. On the other hand, it allows for a better description of the data set being treated.

3.1 Quantile Estimation

Definition 6. *Let X be a random variable. The theoretical quantile of order p $(0 < p < 1)$ is defined as the value x_p such that*

$$\mathbb{P}(X \leq x_p) = p.$$

Given a sample $x_1, \ldots, x_n$, the sample quantile of order p $(0 < p < 1)$ is defined as the value x_p that leaves a proportion p of observations below x_p.

In the original article by Koenker and Basset, cited above, the sample quantile of order p is defined through the expression to be minimized

$$\min_{x_p \in \mathbb{R}} \left[\sum_{x_i > x_p} p|x_i - x_p| + \sum_{x_i \leq x_p} (1-p)|x_i - x_p| \right].$$

It can be verified that the optimal value x_p is such that $p = \frac{\#\{x_i \leq x_p\}}{n}$. Therefore, it is equivalent to the given definition.

The idea of quantile regression is to describe the quantile of order p of the data, instead of the mean (as usual). This concept can be thought of from two different perspectives.

- **Functional perspective**: The article presents the idea of quantile regression from a functional perspective based on using the above expression as a risk function, instead of the mean squared error. In this way, the property of the data being modeled is the quantile. From this argument, the following definition arises.

Definition 7 (Quantile Loss Function)**.**

$$L(x, x_p) = \begin{cases} p|x - x_p| & \text{if } x > x_p \\ (1-p)|x - x_p| & \text{if } x \le x_p \end{cases}.$$

- **Statistical perspective**: As mentioned in previous chapters, most loss functions can be derived from a likelihood. Below, the deduction of this is made to approach the idea of Koenker and Basset from a statistical perspective.

Definition 8. *A random variable has an $ALaplace(\theta, \sigma, p)$ distribution if its density function is*

$$f(x) = \begin{cases} \frac{p(1-p)}{\sigma} e^{-p\frac{x-\theta}{\sigma}} & , \text{ if } x > \theta \\ \frac{p(1-p)}{\sigma} e^{(1-p)\frac{x-\theta}{\sigma}} & , \text{ if } x \le \theta \end{cases} \qquad 0 < p < 1.$$

Let X be a random variable with an $ALaplace(\theta, \sigma, p)$ distribution with $0 < p < 1$. Consider the sample $x_1, \ldots, x_n$ from X, independent and identically distributed. Next, we formulate the problem of maximizing the likelihood of this sample

$$\mathcal{L}(\theta, \sigma; x) = \prod_{i=1}^{n} f(x_i) \iff l(\theta, \sigma; x) = \sum_{i=1}^{n} \log f(x_i).$$

Using definition 7, we can write this *log-likelihood* in the compact form

$$l(\theta, \sigma; x) = n \log(p(1-p)) - n \log \sigma - \frac{1}{\sigma} \sum_{i=1}^{n} L(x_i, \theta).$$

And by developing the maximization process, we obtain the expression

$$l(\theta, \sigma; x) = n(\log(p(1-p)) - 1 - \log(L(x_i, \theta))) \propto -L(x_i, \theta).$$

We conclude that minimizing the risk function imposed by Koenker and Basset is equivalent to maximizing the likelihood function of a sample with an $ALaplace$ distribution. Therefore, for a fixed value $0 < p < 1$, the maximum likelihood estimator $\hat{\theta}_p$ approximates the quantile of order p of X. To delve deeper into this result, consider the following definition.

Definition 9. *The quantile function is defined as the function $Q : (0, 1) \to \mathbb{R}$ such that*

$$Q(p) = \inf\{x \in \mathbb{R} \,|\, p \le \mathbb{P}(X \le x)\}.$$

We can think that $Q(p) \approx \hat{\theta}_p$, and therefore we can obtain an approximation of the image of Q, considering different values of $p = p_i$. Given that $Q^{-1}(x) = \mathbb{P}(X \le x)$, we also have an approximation of the distribution function of X. Finally, we can obtain an approximation of the density function of X by constructing a histogram from the $\hat{\theta}_{p_i}$. Considering the discretization of the interval $(0, 1)$ with step $\Delta_i = p_{i+1} - p_i$, we calculate the difference between consecutive approximate quantiles and fit a rectangle of area Δ_i.

3.2 Linear Quantile Regression

The idea presented in the previous section can be quickly generalized to the linear model

$$Y \sim X\beta + \varepsilon$$

Following the definition in Chapter 1, in this case, the error term ε is considered to have an $ALaplace(0, \sigma, p)$ distribution. It can be shown that

$$X\beta + \varepsilon \sim ALaplace(X\beta, \sigma, p).$$

Therefore, as mentioned, in this case, the quantile of order p of $Y|X$ is described as

$$Q_{Y|X}(p) = \beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m.$$

Note that instead of obtaining a single line describing the conditional mean, as occurs in the case of normal errors, we obtain a line for each p that leaves a proportion p of data below it.

To better illustrate this, we have simulated three samples of sizes $n = 2000$ for the first and $n = 1000$ for the other two, of the random variables

$$X \sim N(0, 1), \quad \varepsilon_1 \sim N(-5, 1), \quad \varepsilon_2 \sim N(5, 1).$$

The sample of the response variable has been defined as $Y = \beta_0 + \beta_1 X + \varepsilon$, where ε represents the bimodal random variable containing the same proportion of elements from ε_1 and ε_2. In practice, this translates to the union of the two samples from ε_1 and ε_2.

In Figure 3.1a, it can be seen that since the error is not normally distributed, Y is not normal either. In this case, the mean is not the best measure and the regression line using least squares is a poor model.

What is proposed with quantile regression is to estimate any quantile of order p of the variable Y. In the case of linear regression, for each quantile, we will obtain a line, as it can be seen in Figure 3.1b. The set of all quantiles allows us to estimate the conditional density. The result of this procedure can be seen in Figure 3.1c.
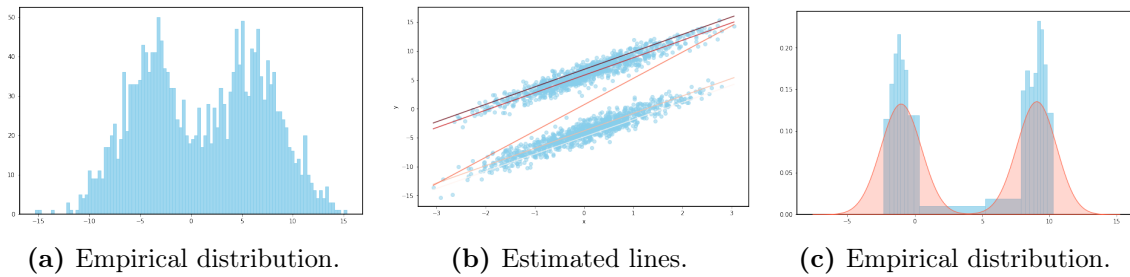

(a) Empirical distribution. (b) Estimated lines. (c) Empirical distribution.

Figure 3.1: Example of linear model using quantile regression, where different lines (for different p values) have been fitted and an approximation of the density of a sample with a bimodal distribution has been constructed.

3.3   Quantile Neural Network

Similarly to linear regression, we can generalize this to other models; in this case, to neural networks. As explained in Chapter 2, the training of a network depends on the loss function used. Alternatively, from a statistical perspective, it depends on the distribution of the error being assumed. Examples have shown that if we take the mean squared error as a risk function, the optimal solution is the conditional expectation, and consequently, the network predicts this measure. However, if we consider the quantile loss function (or equivalently, assume that the error is distributed with an $ALaplace(0, \sigma, p)$), we achieve modeling the quantile of $Y|X$.

Next, a sample is generated to illustrate this concept. We consider a sample of size $n = 2000$ of the random variables

$$X \sim U(-5, 5), \quad \varepsilon_1 \sim N(-5, 1), \quad \varepsilon_2 \sim N(5, 1).$$

We define $Y = X^2 + \varepsilon$, where ε is the bimodal random variable defined with ε_1 and ε_2. We train a neural network with 3 layers of 10 neurons each and a final output layer. In Figure 3.2, the different functions approximated by the network for the quantiles $0.25, 0.40, 0.60$ and 0.75 are observed. Moreover, it can be shown the approximate distribution of $Y|X$.
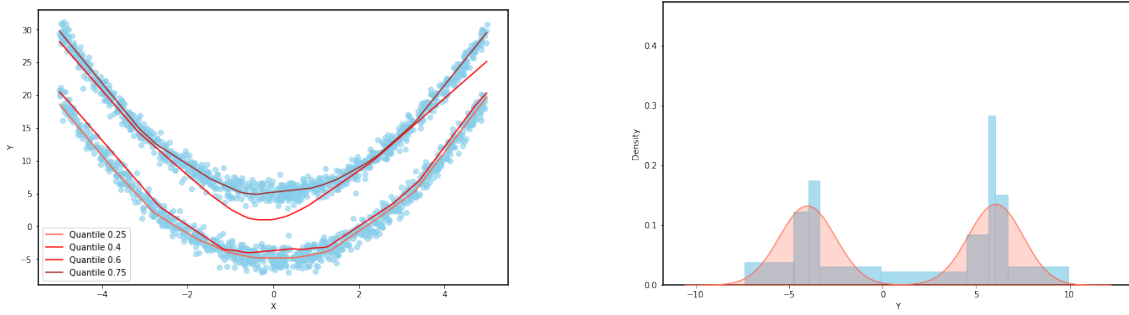

Figure 3.2: Functions approximated by a neural network trained with quantile estimation (left) and empirical distribution of $Y|X = 1$ constructed using the quantiles estimated with quantile regression (right).

3.4   Quantile Crossing Problem

Since the distribution function is monotonically increasing, by definition, it satisfies the order relation

$$p_1 \leq p_2 \implies Q(p_1) \leq Q(p_2).$$

Given $0 < p < 1$, we are approximating $\hat{Q}(p)$, that is a random variable with a well-known distribution. If we are estimating two quantiles very close to each other, the confident interval of each approximation crosses. For this reason, it is common for this order relation not to be satisfied due to this uncertainty. To correct this error, an improvement in the method is proposed.

Instead of training a model for each quantile, the structure is modified so that the output has m variables corresponding to the m quantiles to be approximated. In this way, we can modify the loss function to preserve the order relation.

Definition 10 (Regularized Quantile Loss Function)**.**

$$\bar{L}(x_j, x_{p_j}) = \sum_{j=1}^{m} L(x_j, x_{p_j}) + \lambda \mathbb{1}_{\{p_{i-1} \geq x_{p_i}\}}(x_{p_i}),$$

where L is the Quantile Loss Function and $\lambda \in \mathbb{R}$ *is the regularization coefficient.*

The idea is based on adding a term that penalizes the non-compliance of the order relation by subtracting two consecutive predictions.

Figure 3.3 shows the same example as in section 3.3, but in this case using this improved method. The main difference lies in the permissibility of calculating closer quantiles with greater certainty of not crossing. Consequently, we obtain a more accurate result.
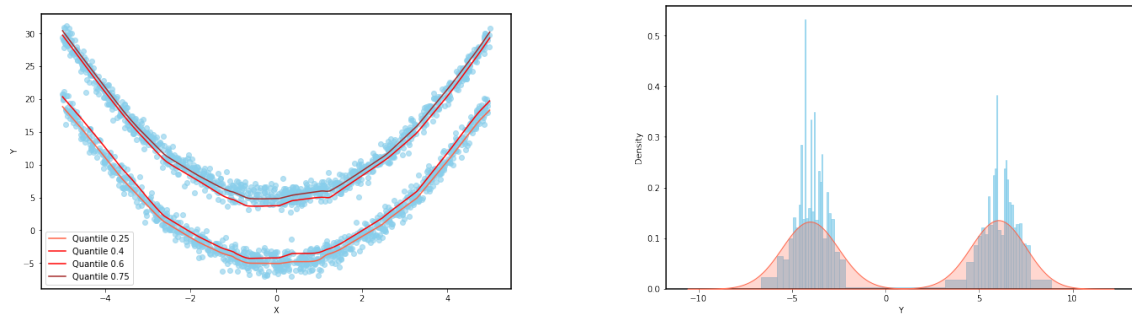
Figure 3.3: Approximated functions (left) and empirical distribution of $Y|X = 1$ (right) calculated by the neural network trained with the improved quantile estimation method.

Chapter 4

Disturbance Storm Time Model
Forecasting its conditional distribution

Geomagnetic storms, also known as solar storms, are disturbances of the Earth's magnetic field. They are mainly caused by solar wind waves and coronal mass ejections [13]. In recent years, several storms have been recorded that have caused severe damage to electrical and electronic systems, satellites, communication devices and other devices that use the Earth's magnetic field as a reference [1]. In an era where all these technologies are of great importance in many aspects of daily life, from individuals to large companies, precise control of these extreme phenomena is crucial.

The severity of these solar storms is measured by the *Disturbance Storm-time*, *Dst*, index [3]. Although significant advances have been made in recent decades in predicting the *Dst* through physical studies and machine learning models [14], there are still gaps in understanding how this data behaves [15].

Therefore, it is essential to develop more sophisticated models that not only predict the *Dst* index but also estimate its conditional density. This approach would allow a more detailed and accurate understanding of geomagnetic fluctuations, thereby improving the response capacity to potential damage caused by solar storms.

4.1 Problem Statement

The main objective is to create a model for the *Dst* using various geomagnetic data to understand these events and prevent technological damage that could result from the unforeseen occurrence of one of these storms. More specifically, it is proposed to:

- Design and implement a model capable of explaining the *Dst* variable at the current time using known variables. Specifically, approximate the distribution and density of the *Dst* conditioned on the other observations. The model must be robust for any value, especially extreme ones.

- Accurately describe the tail of this distribution to make predictions about a threshold of the *Dst* value and estimate the probability of occurrence of an extreme event.

- Propose a family of distributions that coherently fits the real law of the *Dst* conditioned on the other variables.

In order to achieve the previous goals, the following geomagnetic real data has been chosen. NASA's Advanced Composition Explorer (ACE) and NOAA's Deep Space Climate

Observatory (DSCOVR) satellites provide various solar wind data every minute. This data has been directly extracted from the official NASA and NOAA websites [16, 17] over three different time periods. The following details each of these:

- The components (x, y, z) of the interplanetary magnetic field (IMF) in geocentric solar ecliptic (GSE) coordinates, measured in nanoteslas.

- The components (x, y, z) of the IMF in geocentric solar magnetospheric (GSM) coordinates, measured in nanoteslas.

- The magnitude of the IMF vector, in nanoteslas.

- The latitude of the IMF in GSE and GSM coordinates, defined as the angle between the magnetic vector and the ecliptic plane, measured in degrees.

- The longitude of the IMF in GSE and GSM coordinates, defined as the angle between the projection of the magnetic vector on the ecliptic and the Earth-Sun direction, measured in degrees.

- The proton density of the solar wind, in N/cm^3.

- The mass speed of the solar wind, in km/s.

- The ion temperature of the solar wind, in Kelvin.

Due to the high correlation between solar cycles and geomagnetic storms [18], sunspot number data [19] during the same time period has also been included; in this case, the records are monthly. Additionally, since the ACE and DSCOVR satellites are not stationary but orbit around the Earth-Sun Lagrange point *L1* in a constant relative position to the Earth [20], the daily position of these two satellites has also been considered. Finally, the hourly *Dst* values calculated by the four ground observatories have been taken as the response variable of the model [21].

4.2   Design and Implementation

This section describes the implemented model to approximate the conditional distribution of the *Dst* given the mentioned features. As shown in Figure 4.1, it is based on three stages: preprocessing, the neural network and the distribution calculation. The process used in each of these is detailed below.
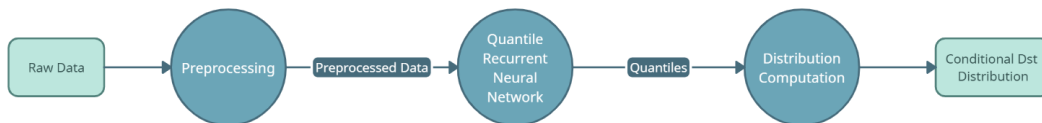

Figure 4.1: Architecture of the implemented model.

4.2.1 Preprocessing

The two main problems that arise when working with real data like the ones we are dealing with are diversity in the format of the different data sources and missing values.

First, the data is divided into two subsets: training and test. A portion of 20% is separated as the test set to evaluate the model's performance at the end of the process. The remaining data is used to train the model. From this point on, we work with the training set to avoid problems like *data leakage*, which can lead to an incorrect and overly optimistic evaluation of the model.

We start by processing the solar wind data set. First, missing data is imputed using interpolation. This method has been chosen due to the nature of the data: being a time series, it is necessary to preserve the continuity of the data and maintain their patterns and trends. Next, to transform the data into hourly, the median of each hour is calculated. As seen in Figure 4.2, the appearance of heavy tails for large values in the variables presented as examples is noted. The data is not symmetrical and tends to present outliers, concluding that the mean is not a representative measure of the data. Instead, the median helps reduce the effect of these extreme values that may not be representative and allows us to handle the asymmetry of the data [22]. Additionally, an extra variable with the standard deviation of each hour is added to understand the variability of each feature.
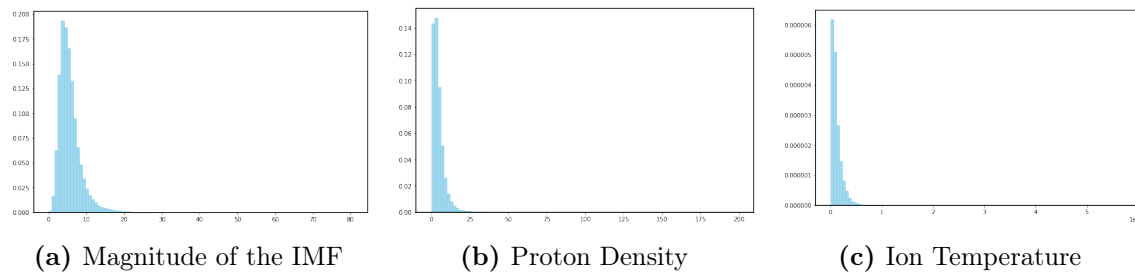

(a) Magnitude of the IMF   **(b)** Proton Density   **(c)** Ion Temperature

Figure 4.2: Distribution of the magnitude of the IMF, proton density, and ion temperature of the solar wind.

Second, we merge the sunspot data. Since we only have monthly information, missing data is imputed with the most recent value. This technique is used because the primary objective of including this data is to capture the influence of the eleven-year solar cycles; the scale of a cycle is too large to notice effects on an hourly basis.

Third, we include the daily coordinates of the two satellites. Since there are observations from both ACE and DSCOVR, depending on the period, the corresponding coordinates must be merged. In cases where we do not have a reference from which satellite the information comes from, it is imputed with the most recent value.

Finally, we prepare the *Dst* data set by creating a second variable corresponding to the *Dst* value one hour earlier. This will help us predict the *Dst* using previous values. It should be noted the distribution of this variable, observed in Figure 4.3. We note the presence of non-symmetrical data with a tendency towards extreme negative values, creating a heavy tail. This is reflected in the QQ-plot, also in Figure 4.3, allowing us to conclude that we are dealing with non-normal data. Recognizing this non-normality is important for choosing the appropriate statistical methods for analysis.

Finally, we normalize the data so that our model interprets each variable's contribution fairly and accurately, without being biased by scale differences [23].
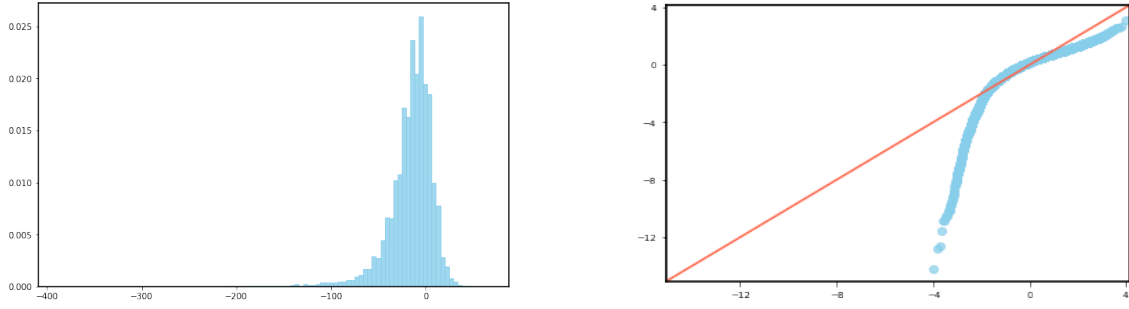
Figure 4.3: Distribution and QQ-plot of the response variable, the *Dst* index.

4.2.2 Quantile Recurrent Neural Network

Since we are dealing with time series, the chosen model is a Recurrent Neural Network. This type of neural network is particularly well-suited for time series analysis because it can consider past information when predicting the current *Dst* value. By maintaining an internal state that captures information from previous time steps, the RNN can effectively model temporal dependencies within the data. This way, the model can incorporate historical context to make more accurate predictions.

Additionally, to approximate the distribution of the *Dst* conditioned on the other features, the quantile loss function is chosen (with a restriction parameter $\lambda = 10^{-4}$).

Finally, following the notation in Chapter 2, the defined hyperparameters are: sequences of length $T = 34$, number of neurons $K = 256$ and $m = 100$ output values, corresponding to each of the quantiles. In this case, a uniform partition of the interval $(0, 1)$ has been considered.

It should be noted that, to ensure better generalization of the model, the use of regularization techniques could be considered. Additionally, optimizing additional hyperparameters such as the learning rate and the type of optimizer (for example, AdaGrad or RMSprop) could lead to improvements in model performance [24]. However, both for computational reasons and because it deviates from the ultimate goal of this work, they have been fixed in advance.

The model is trained with 80% of the data over 500 epochs. An epoch refers to one complete pass through the entire training dataset, during which the model updates its weights based on the loss function. Training over 500 epochs ensures that the model has sufficient opportunities to learn the patterns within the data [25].

4.2.3 Distribution computation

Once the model has been trained, predicting a new observation results in a set of m p_i-quantiles, for $i = 1, \ldots, m$. That is,

$$\hat{F}(X) = (\hat{Q}(p_1), \ldots, \hat{Q}(p_m)).$$

On the one hand, the points $(\hat{Q}(p_i), p_i)$ plot the distribution function. On the other hand, if we process the results following the method mentioned in Chapter 3, the density function is plotted.

From this point, we have the necessary tools to conduct a detailed analysis and answer the questions posed at the beginning.

4.3   Results

Finally, all the results extracted from the implemented model are detailed in the following section. This model has allowed estimating the entire distribution and, because of this, some other studies has been done.

4.3.1   Estimated distribution and density function of $Y|X = x$

As it has been explained, the proposed model not only allows predicting a single value but approximates the entire distribution. The approximation of the distribution and density functions of a new observation, where the Dst value at the current time is -153, is shown in Figure 4.4.
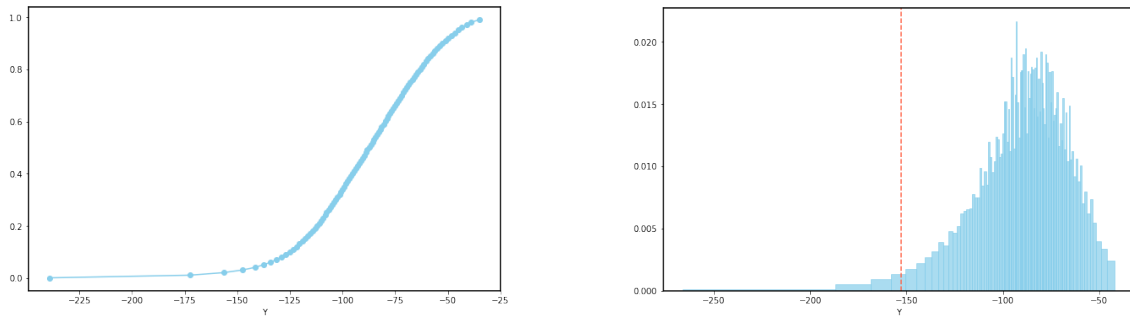
Figure 4.4: Distribution (left) and density (right) function approximated by the model.

4.3.2   Testing the model

To verify the robustness of the model, we use the fact that any random variable applied to its distribution function results in a uniform random variable [26]. Considering that each real value y_i is a realization of the true conditional distribution and considering the previous result, the set $\{p_i \,|\, y_i = \hat{Q}(p_i), i = 1, \ldots, m\}$ should be a uniform sample. In Figure 4.5, this behavior is observed for both the training and test sets. This allows us to give some credibility to the results obtained below.
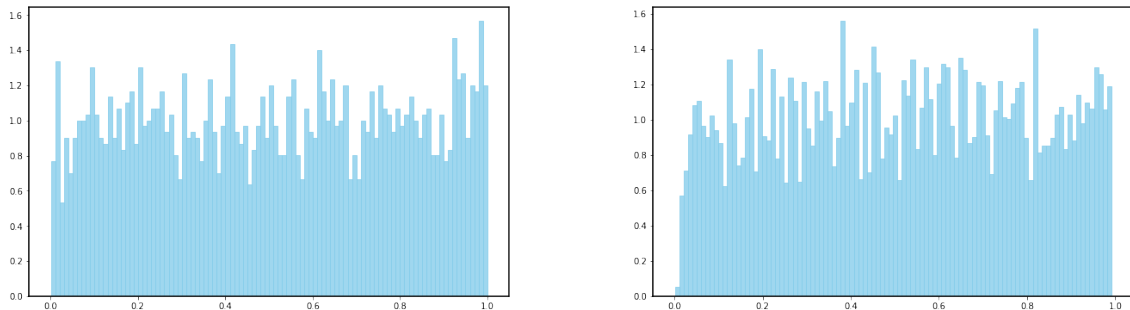

Figure 4.5: Histogram of the sample of probabilities for each real Dst value according to the approximate distribution, for some training set observations (left) and all test set (right).

4.3.3   Study of the distribution shape

Returning to the previous example, the first notable characteristic is the unimodality and improvement in the symmetry of the conditional distribution $Y|X$ compared to that of Y. In other words, the model provides a simplification of the *Dst* law, making it easier to study. However, the appearance of long tails and the information from the QQ-plot suggest that the distribution is not normal, as it can be seen in Figure 4.6.
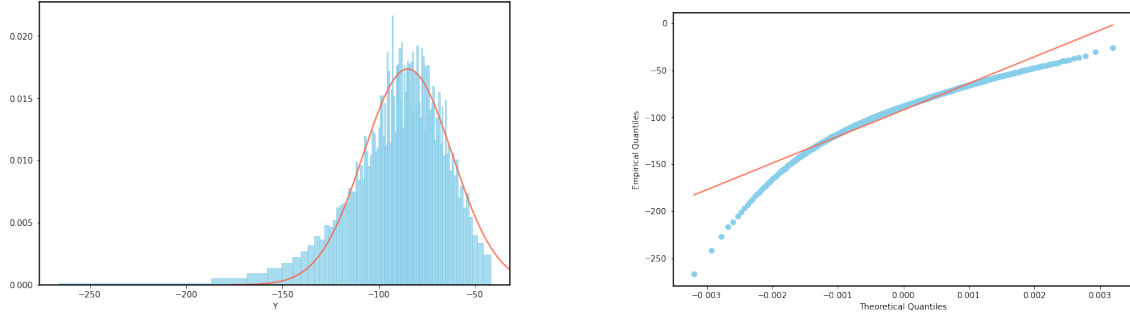

Figure 4.6: Comparison between the approximate distribution and a normal distribution, and a QQ-plot of the empirical distribution relative to a normal distribution. Notice that, for both graphics, the appearance of heavy tails is observed.

4.3.4   Estimated left tail of the $Y|X = x$ distribution

The previous observation, along with the fact that the model approximates the whole rang of the left tail, leads us to fit a function f that coherently approximates this tail. Given its monotonous decline, it is proposed to fit it with an exponential function. Therefore, for $y \in \mathbb{R}$ small enough, the structure is

$$f(y) = e^{a+by}.$$

One way to fit this function f is through a logarithmic transformation and simple linear regression. Following the explanation in Chapter 1, the quantiles calculated by the model through the fixed probabilities, that can be thought of as realizations of a random variable, along with least squares give us the approximation shown in Figure 4.7.
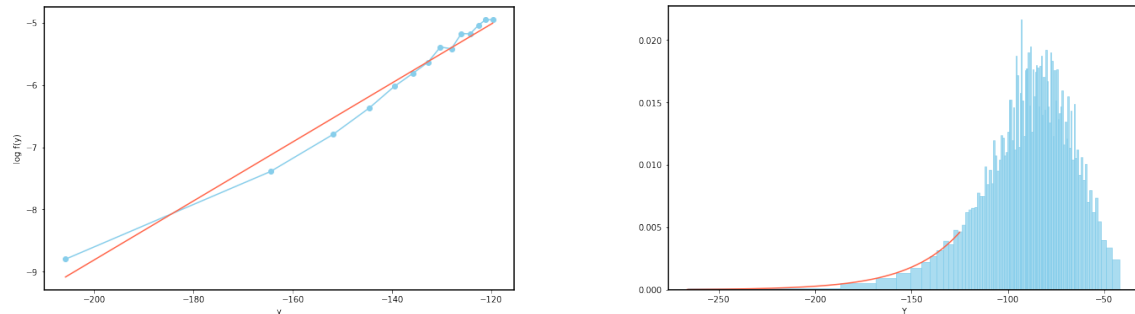

Figure 4.7: On the left, the good fit of the data after the log-transformation is observed. On the right, the proposed semi-parametric model (PoT) using an exponential function for the left tail, formed by the lower 15% of the data (threshold $u = -143$), is shown.

4.3.5 Inference in extreme values

Once the conditional tail distribution function f is fitted, an extreme event can be accurately predicted. Concretely, the following statements can be analyzed:

- Determine a lower bound given a confidence level p.

- Determine a probability given a lower bound.

Each of these problems is formulated using the expression

$$\mathbb{P}(Y \le y \,|\, X = x) = 1 - p \iff \int_{-\infty}^{y} f(s)ds = \frac{1 - p}{0.15},$$

where $\mathbb{P}(Y \le u \,|\, X = x) = 0.15$ and $p > 0.85$. Thanks to this result, the model allow us to make inference in the hourly Dst value. Concretely, given the geomagnetic observations from the NOAA and DSCOVR satellites and the number of sunspots over 35 hours, along with the Dst values measured by the 4 ground observatories over the previous 34 hours, the model can predict the current minimum Dst value with a 90% probability (or the chosen probability $p > 0.85$). Additionally, it provides the probability of a solar storm occurrence, considering the effects to be catastrophic starting from a value of -200 (or the chosen threshold).

Considering October 5, 2000, at 8:00 AM. The value of Dst at that hour was -153 (an observation to which the implemented model has been applied and the tail has been adjusted as described). Using the 35-hour sequence of the mentioned data, as shown in Figures 4.8b and 4.8c, along with the 34-hour Dst data shown in Figure 4.8a, the model can provide a prediction for the current Dst value. The maximum IMF magnitude during this period was $28.22nT$, the proton density reached $16.86N/cm^3$ and the Dst peaked the previous hour with a value of -175. With this data, along with the other necessary sequences, the model predicts the following values:

- The value of the Dst was greater than -153.49, with a 90% of certainty.

- If a value below -200 is considered alarming enough to trigger a significant solar storm, the probability of occurrence is 0.31%.
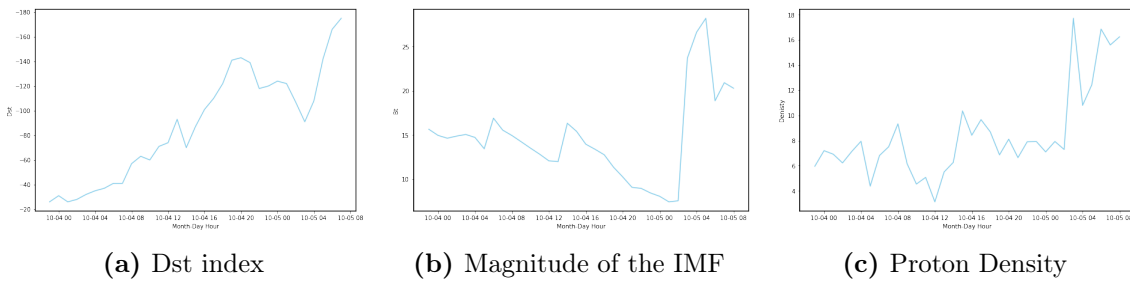

(a) Dst index   (b) Magnitude of the IMF   (c) Proton Density

Figure 4.8: Time series of three necessary components to make inference about the actual Dst value.

4.3.6 Model for the conditioned distribution of Dst

Finally, the detailed study of a specific observation can be extrapolated to the rest of the test set observations. This is due to the similarity in the shape of the distributions of the different observations (after they have been normalized) and the appearance of long tails in each of them, shown in Figures 4.9a and 4.9b. Therefore, it is reasonable to think that the distribution of Dst conditioned on each of the features can be modeled by the same family of distributions. Given the resemblance of the shape of each of the approximate distribution functions to the *sigmoid* function, the general law of $Y|X$ can be modeled by

$$F(y) = \frac{1}{1 + e^{a+by}},$$

where F denotes the distribution function. As it has been seen, the scale of $Y|X$ can be adjusted with two parameters, which makes the sigmoid a good choice because is one of the simplest functions that allow this, as well as the good properties that it has [27]. However, other functions could have been chosen. Similarly to what was done to model the tail of the density, a linear model can be fitted with all values at once. Due to the good fit obtained, as it can be shown in Figure 4.9b and Figure 4.9c, it is concluded that the conditional distribution can be modeled by the logistic distribution.
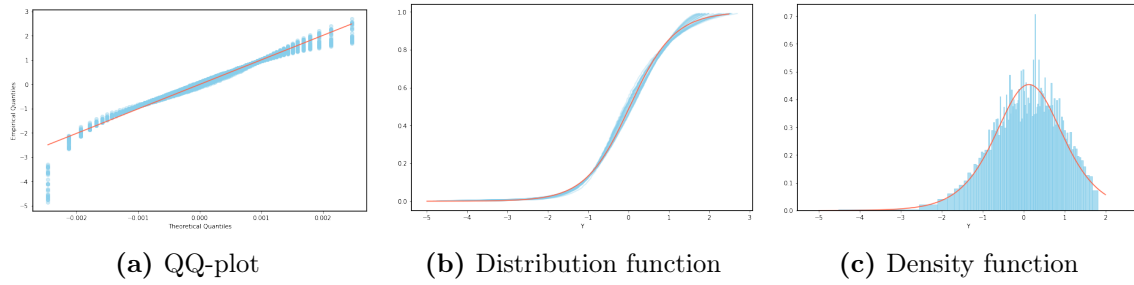

(a) QQ-plot   (b) Distribution function   (c) Density function

Figure 4.9: The QQ-plot performed on different sequences and the similarity of the approximate distribution functions show indications of the general appearance of heavy tails. This leads to considering the logistic distribution as a model for the law. It is observed that both the distribution function and the density function fit the data correctly.

Finally, the results section closes by mentioning the previous result obtained thanks to the model implemented in this project.

The distribution of Dst conditioned on the mentioned variables follows a logistic distribution. That is, $Y|X = x \sim Logistic(\mu(x), s(x))$ and has a density function

$$f_{Y|X=x}(y) = \frac{e^{-(y-\mu(x))/s(x)}}{s(x)(1 + e^{-(y-\mu(x))/s(x)})^2}$$

where $\mu(x) = \mathbb{E}[Y|X = x]$ and $s(x)^2 = 3/\pi^2 \, Var(Y|X = x)$, and therefore depends on the sequence $X = x$ being conditioned.

This result not only allows us to give a model to the entire conditioned Dst distribution, but also concludes that it can be modeled with two parameters. That is, thanks to the chosen features, the complexity of the Dst density function has been reduced to such an extent that the tails of the distribution could have been modeled and the density can be described by a model with two parameters, mean and variance, that only depends on these features.

Chapter 5

Conclusions

The importance of being able to describe and predict an extreme event, such as solar storms, in a accurate way is fundamental. Nowadays, society depends strongly on electronic devices that could be damaged due to a lack of anticipation to these occurrences. However, there are still limitations to describe how these events behave.

For that reason, the main goal of this project has been to design and implement a model that allows us to describe the *Disturbance Storm Time* index. In order to achieve this final goal, a different perspective on facing the problem has been given. A combination of the linear model, a Recurrent Neural Network and the idea of the quantile regression has been used not only to predict a single value, but to provide a description of the Dst distribution and density function conditioned on some geomagnetic data.

In the first place, a mathematical definition of the linear model has been provided, as well as examples when the data follows a normal distribution. In the second place, neural networks have been presented; each element of this model has been defined rigorously. Moreover, a probabilistic perspective has been provided that allows us to understand the relationship between the linear model and how the assumptions that are made affect the measure that is being described. This insight has allowed us to subsequently introduce the concept of quantile regression. This allows us to describe the quantile of order p instead of the mean, which is usually the most common due to the normality assumption. This fact gives us several advantages that have been exemplified with synthetic data: among others, it allows us to describe and predict the conditional density function. Once all the ingredients have been presented, the model's objectives have been defined in a detailed way, the employed methodology has been explained and, finally, the results have been exposed.

The implemented model, together with the geomagnetic data from NOAA and DSCOVR satellites, has led to the following results. Firstly, the distribution and density function approximated by the model for a new observation have been shown. Thanks to this, an analysis of the shape of this distribution has been made: the conclusion drawn is that the Dst conditioned on the other variables doesn't follow a normal distribution. In spite of having the particular form of this distribution, the long tails are the main characteristic to refute this hypothesis. Because of this, it has been sought to precisely model the left tail to make inferences about extreme values. Through a logarithmic transformation and a linear model, this tail has been adjusted. Thanks to this, the model is capable of calculating the probability of occurrence of an extreme event and a bottom level with a fixed confidence level (conditioned on the variables of the particular observation). Finally, it has been shown that conditioning the Dst on the geomagnetic data reduce the complexity of the Dst.

Additionally, the fact that each of the distribution functions approximated by the model could be rescaled has provided evidence that they all come from the same family. Therefore, the sigmoid function has been chosen to fit this distribution and give a model because of its simplicity. Due to the good fit, the conclusion is that the *Dst* distribution conditioned on the chosen geomagnetic variables is a *Logistic*. More specifically, the following result has been presented.

The distribution of Dst conditioned on the mentioned variables follows a logistic distribution. That is, $Y|X = x \sim Logistic(\mu(x), s(x))$ and has a density function

$$f_{Y|X=x}(y) = \frac{e^{-(y-\mu(x))/s(x)}}{s(x)(1 + e^{-(y-\mu(x))/s(x)})^2}$$

where $\mu(x) = \mathbb{E}[Y|X = x]$ and $s(x)^2 = 3/\pi^2 \ Var(Y|X = x)$, and therefore depends on the sequence $X = x$ being conditioned.

This result not only allows us to create a model for the entire conditioned *Dst* distribution, but it also concludes that it can be modeled with two parameters. That is, due to the selected features, the complexity of the *Dst* density function has been reduced to such an extent that the tails of the distribution could be modeled, and the density can be described by a two-parameter model, mean and variance, which only depends on these features.

Note that, this conclusion provides us with sufficient evidence to use it as a starting point for future projects dedicated to improving these extreme models. Because of this, several ideas are proposed to continue this study and improve it:

- It would be interesting to complement the study with various statistical tests. Thanks to this, more support could be given to the proposed model or some details could be modified to improve it.

- The model could improve if some other architectures were used.

- Taking advantage of the implemented model, it could be analyzed in depth how each of the variables affects the distribution. Recently, studies on neural network interpretability techniques have emerged [28], which, if adapted to this model, would help to complement these changes more rigorously.

In summary, this work has served to understand the relationship between two of the most used models in data science in order to present quantile regression and provide a different solution to a problem with real data. Therefore, the study can serve as an inspiration for future proposals, different from conventional machine learning models.

References

[1] NASA. *Space Weather*. 2024. URL: https://www.nasa.gov/mission_pages/sunearth/spaceweather/index.html.

[2] Gabor Toth et al. *Predicting solar storms to protect Earth*. https://earthsky.org/space/predicting-solar-storms-to-protect-earth/. 2023.

[3] NOAA National Centers for Environmental Information. *Dst Index - Geomagnetic Disturbance Storm-Time Index*. https://www.ngdc.noaa.gov/stp/geomag/dst.html.

[4] Peter McCullagh. "What is a statistical model?" In: *Annals of Statistics* 30.5 (2002), pp. 1225–1310. DOI: 10.1214/aos/1035844977. URL: https://projecteuclid.org/euclid.aos/1035844977.

[5] E. John Brunner and Gregory N. White. *Linear Models in Statistics*. 2018. URL: https://www.utstat.toronto.edu/~brunner/books/LinearModelsInStatistics.pdf.

[6] Bin He et al. "Machine learning in biomedicine and oncology: Towards precision medicine". In: *Frontiers in Genetics* 8 (2017), p. 32. DOI: 10.3389/fgene.2017.00032. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5370305/.

[7] Murat Murat Arat. *Proof of Unbiased Estimator*. https://mmuratarat.github.io/2019-09-27/unbiased-estimator-proof. 2019.

[8] S. S. Wilks. "The large-sample distribution of the likelihood ratio for testing composite hypotheses". In: *The Annals of Mathematical Statistics* 9.1 (1938), pp. 60–62.

[9] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. 1st. New York: Springer, 2013. ISBN: 978-1-4614-7138-7. URL: https://www.statlearning.com/.

[10] Aisha Lichtner-Bajjaoui. "A Mathematical Introduction to Neural Networks". MA thesis. Barcelona: Universitat de Barcelona, 2020.

[11] Viren Rehal. "Heteroscedasticity: Causes and Consequences". In: *SPUR Economics* (2023). URL: https://spureconomics.com/heteroscedasticity-causes-and-consequences.

[12] Roger Koenker and Gilbert Bassett Jr. *Regression Quantiles*. Research Report NSF-RANN-781. Champaign, IL: University of Illinois, 1978. URL: http://www.econ.uiuc.edu/~roger/NAKE/rqs78.pdf.

[13] NOAA Space Weather Prediction Center. *Geomagnetic Storms*. 2024. URL: https://www.swpc.noaa.gov/phenomena/geomagnetic-storms.

[14] E.-Y. Ji et al. "Comparison of Dst forecast models for intense geomagnetic storms". In: *Journal of Geophysical Research: Space Physics* 117.A3 (2012). DOI: 10.1029/2011JA016872. URL: https://doi.org/10.1029/2011JA016872.

[15] University of Michigan News. *Solar flares: U-M experts highlight gaps preventing accurate predictions of impacts around Earth*. https://news.umich.edu/solar-flares-u-m-experts-highlight-gaps-preventing-accurate-predictions-of-impacts-around-earth/. 2023.

[16] NOAA National Centers for Environmental Information. *Geomagnetic Data*. 2024. URL: https://www.ngdc.noaa.gov/geomag/data/.

[17] NASA Goddard Space Flight Center. *ACE Solar Wind Data*. 2024. URL: https://omniweb.gsfc.nasa.gov/ftpbrowser/pla_cross_wind_ace.html.

[18] T. Sundstorm. "Correlations between Solar Cycles and Geomagnetic Storms". In: *Monthly Notices of the Royal Astronomical Society* 530.3 (2021), pp. 3171–3182. URL: https://academic.oup.com/mnras/article/530/3/3171/7649392.

[19]   Royal Observatory of Belgium, Solar Influences Data Analysis Center (SILSO). *Sunspot Data*. 2024. URL: `https://www.sidc.be/SILSO/datafiles`.

[20]   NOAA / NWS Space Weather Prediction Center. *Real Time Solar Wind and NOAA's DSCOVR Satellite*. 2018. URL: `https://www.swpc.noaa.gov/news/real-time-solar-wind-and-noaas-dscovr-satellite`.

[21]   Kyoto World Data Center for Geomagnetism. *Dst Index Data*. 2024. URL: `https://wdc.kugi.kyoto-u.ac.jp/dstdir/`.

[22]   Research Collective. *Means and Medians: When to Use Which*. `https://research-collective.com/means-and-medians-when-to-use-which/`. Accessed: 2024-06-23. 2023.

[23]   Samit Bhanja and Abhishek Das. "Impact of Data Normalization on Deep Neural Network for Time Series Forecasting". In: *arXiv preprint arXiv:1812.05519* (2018). URL: `https://arxiv.org/abs/1812.05519`.

[24]   A. Oppersmann. "Optimization in Deep Learning: AdaGrad, RMSProp, ADAM". In: *KI Tutorials* (2023). URL: `https://artemoppermann.com/optimization-in-deep-learning`.

[25]   Liping Yang. *What is the Difference Between a Batch and an Epoch in a Neural Network?* `https://deeplearning.lipingyang.org/wp-content/uploads/2018/07/What-is-the-Difference-Between-a-Batch-and-an-Epoch-in-a-Neural-Network_.pdf`. 2018.

[26]   Massachusetts Institute of Technology. *Probability and Random Variables*. 2014. URL: `https://ocw.mit.edu/courses/18-440-probability-and-random-variables-spring-2014/d06be636fc6ff9365caf6ef166eb2b23_MIT18_440S14_Lecture18.pdf`.

[27]   Ian C. Clendinning. *The Logistic Function*. 2023. URL: `https://www.mygeodesy.id.au/documents/The%20Logistic%20Function.pdf`.

[28]   Aleix Sancho. *Explainable Machine Learning and Credit Risk Models*. `https://github.com/Aleix-Sancho/Explainable-Machine-Learning/blob/main/Explainable_Machine_Learning_and_Credit_Risk_Models.pdf`. 2024.

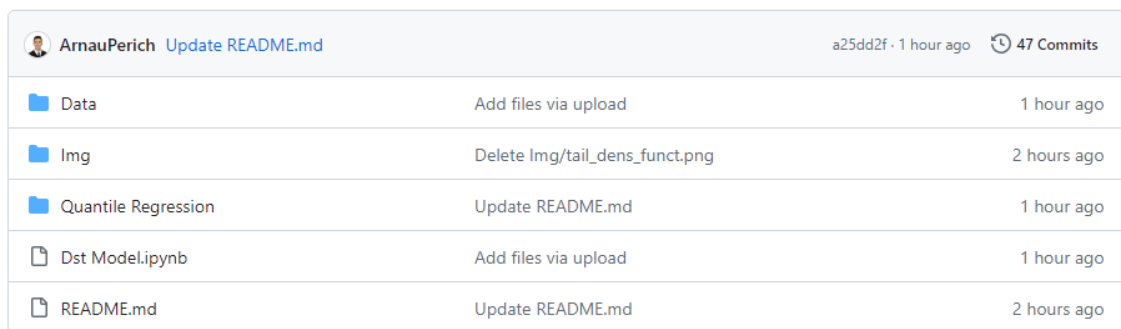# Appendix A

Implemented Code
Model and Figures

The implemented model and all the images shown in this project are our own. And because of this, they have been implemented in Python scripts. Manily, this project has two Python notebooks files:

- **Quantile Regression:** all the examples shown in Chapter 3 are implemented in this file.

- **Disturbance Storm Time Model:** the model described in Chapter 4 is implemented in this file, as well as the figures used to extract the conclusions.

All the examples have been explained before and both Python files are commented so as that the reader has a better comprehension. Moreover, these Python files, as well as the generated images and the data files used in this project, have been uploaded to GitHub. The link is provided as follows:

https://github.com/ArnauPerich/DisturbanceStormTimeModel

Figure A.1: GitHub repository

As it can be seen in the Figure A.1, the GitHub repository is organized with three folders and the main Python file, that contains the implemented model. The folders contains the data files, the generated images and the quantile regression code file, respectively.