An artificial learning biophysical model for the evolution of land use and greenhouse gases emissions

Bruno Pereira Costa
bruno.pereira.costa@tecnico.pt

Instituto Superior Técnico, Lisboa, Portugal

January 2021

Abstract

Quantitative models of greenhouse gases (GHG) emissions can help policy-makers gauge sustainable pathways towards limiting global warming to stay within 1.5°C of the Paris Agreement. The agricultural and energy sectors are particularly important, as they jointly represent more than two thirds of global GHG emissions. This thesis uses an ensemble of recurrent and convolutional neural networks and hybrid architectures to develop a biophysical model for the evolution of land use and associated GEE in Portugal. The models' hyperparameters tuning uses a state-of-the-art framework of Bayesian optimisation and a new error estimation algorithm based on an out-of-sample growing window approach is proposed. Land use shares were modelled as a function of final exergy in the agricultural sector and Gross Domestic Product (GDP), and emissions as a function of distribution of land use, total final exergy, and GDP. Models were trained for the period of 1961-2016 and applied from 2017 to 2030 under two plausible economic scenarios with and without COVID-19 influence. Results show that land use is correlated with GDP, and GHG emissions from agriculture and energy are correlated with total final exergy. Economic growth leads to a reduction in cropland area, increased intensity of energy consumption, and variations in sectoral GHG emissions. The novel coronavirus pandemic might decrease the cropland area reduction and mitigate the increase in emissions of CO_2e up to 2030. Despite the complexity of the model, estimation errors exceeded the variation range of the forecasted variables. Uncertainty is critical for scenario assessment, casting doubt over simpler models.
Keywords: Agriculture, Bayesian Optimisation, Exergy, Portugal, Recurrent Neural Networks, Convolutional Neural Networks

1. Introduction

1.1. The foundations of a Civilisation

The pillars that support the development and maintenance of a civilisation are access to a stable food supply, social structure, record keeping, technology and arts [1]. Projected impacts of climate instabilities and resources misusage particularly affect one of those pillars: **agriculture**.

Over the past century, there has been spatial segregation between food producers and consumers, leading to changes in terms of energy consumption patterns and land use [2]. Globally, there is a progression towards a set up of densely populated cities whose sustenance comes from intensively cultivated lands/raised livestock decoupled from the cities' site. Without agriculture, it is unmanageable to have modern institutions – it is unquestionably the foundation of our complex civilisation, which is pronouncedly city-based [3, 4]. Altogether, the fate of the civilisation follows the fate of agriculture - our survival is inherently dependent on agriculture's thriving [1, 5, 6].

1.2. Biophysical Modelling

In order to mitigate dangerous ambiguity of scenarios for the future, one should focus on the deceleration or even reversion of the current transition from the stable and naturally driven conditions of the Holocene to the human driven unstable conditions of the Anthropocene [7]. For this purpose, integrated biophysical and economic models capable of explicitly depicting the results of possible sustainable pathways are becoming a common approach. For instance, Automated Land Evaluation System (ALES) is a computerised framework that allows to estimate crop and consequent economic production; Decision Support System for Agrotechnology Transfer (DSSAT) is an explicit suite of models of crop plant production [8]. More recently, the data generated in agriculture operations, gathered largely through remote sensing, has been demonstrated to be suitable for new developed computational techniques. Those methods brought bold and powerful ways of examining particular agriculture problems, being applied essentially to image processing and data analysis [9].

Such models contemplate equilibria between every agent constituting society–nature interactions and processes with well established frontiers, corresponding to untransgressible planetary boundaries [10]. Furthermore, underlying these models is usually a time series approach that can be used to address questions of causality, trends, and the likelihood of future outcomes. Hence, policy makers will have unveiled solid ground for the development and implementation of sustainable management strategies and pathways for a sustainable future.

1.3. Artificial Intelligence

Attempts with modern techniques, such as Artificial Neural Networks (ANNs), being used as a method for modelling and forecasting in environmental studies for Portugal have already been verified. In a study, the useful exergy concept was used to predict the energy consumption based strictly on the economic evolution of the country [11].

Classical biophysical models are theory-driven, or, differently, implemented with a bottom-up approach. This way of description poses a difficulty of interpreting non-linearities between interplaying processes. On the other hand, there are alternative techniques that present themselves as data-driven or, differently, as top-down approaches. The idea is to scour data in order to find novel and useful relationships that might otherwise remain unknown with more traditionalist methods [12].

There is a strong need to deal with the already predictable high number of covariates in a way that enables the extraction of explicit answers on how materials, land and energy have been used to produce value for an economy over time. As mechanistic relationships between variables are unknown and conven-

tional statistics is limited by high covariance and indeterminate causality, the application of Machine Learning (ML) methods, a subcategory of Artificial Inteligence (AI), will attempt to produce clarity regarding the interdependencies between drivers of change and outcomes. Such methods are advantageous in situations where one does not posit an underlying process or any rules about that underlying process. The focus is rather on identifying patterns that describe the processes' behaviour in ways useful to predicting the outcome of interest [13].

It is possible to condense all applications (of where ML has been predominantly applied) to four first level categories. Those are crop management, livestock management, water management and soil management. In crop management some sub-categories are yield prediction, disease detection, weed detection, crop quality, species recognition, fruits counting, and crop type classification (that could be generalised to land cover classification). Regarding livestock management, there are the animal welfare and livestock production sub-categories. In terms of water management, the main object of study is usually the evapotranspiration. Finally, soil management consists of studying agricultural soil properties, such as the estimation of soil drying, condition, temperature and moisture content [14]. Those problems deal with either classification or prediction, with more prevalence of classification.

For all of the previous subjects, the following ML methods have been deployed: support vector machines, Bayesian models, deep learning, decisions trees, ensemble learning, instance based models, ANNs and Recurrent Neural Networks (RNNs). Each one of these algorithms were designed for different purposes and comprehensive reviews with correspondence between problems under study and suitable models already exist [14, 15].

This work is explicitly dives into the Portuguese agriculture sector's sustainability and target an initial analysis of the period comprised between 1961-2016, as substantial changes took place as a consequence of transition from organic fertilisers to chemical-based fertilisers, increased industrialisation and the entry into the European single market. The goal of the thesis is to contribute towards the development of a biophysical model framework. The model will be particularly applied to the evolution of Portuguese land use, with special focus on the agriculture, in order to explain how land and energy have been used to produce value for the Portuguese economy. Moreover, the model will be employed to forecast plausible pathways for the future of the sector, assessing land use emissions, up to 2030. More concretely, we are first going to try to explain the relationship between the distribution of land use in terms of final exergy usage in the agricultural sector [16] and economic growth explained in terms of Gross Domestic Product (GDP). As a second step, we are going to try to study how the former three covariates, land use distribution, economic growth, and final exergy, relate to Greenhouse Gases (GHG) emissions. By comprehensively studying individual sectors that constitute the global economic machine we expect to gain a more precise perspective on the reciprocal dynamics.

2. Background

2.1. Energy and Economy

Energy can be recognised as the only universal currency: one of its many forms need to be transformed to get anything done. Universal demonstrations of these transformations range from the rotations of galaxies to thermonuclear reactions in stars. Life on Earth would be impossible without the photosynthetic conversion of solar energy into phytomass, it underpins all higher life. Humans depend on this transformation for their survival, and many more energy flows to support their existence within increasingly complex societies [1].

2.1.1 Exergy: why and what is it?

A better way of tracing the influence of energy is to express it in terms of its potential usefulness, *i.e.*, the actually delivered heat, light, and motion. This subdivision of a given amount of energy is called exergy, and it is not a conserved property because it can be transformed into anergy by irreversibilities, as a consequence of the 2^{nd} law of thermodynamics.

The concept of exergy is defined as the maximum work obtainable when the system is brought to a reference state of thermodynamic equilibrium by means of completely reversible processes or, differently, by means of ideal energy conversion processes. Therefore, a reference state must be defined. In standard uses, it is defined to be a state of thermodynamic equilibrium characterised by the same temperature, pressure and chemical composition as the environment. Exergy is also a thermodynamic measure of energy quality, measuring the availability to perform work of a certain amount of energy, given reference environmental conditions.

2.1.2 Useful Exergy and Economic Growth

In an exergy framework, the useful stage of the energy flow accounts for satisfied energy needs. This means that an useful energy analysis with an exergy approach leads to a measure closer to the productive energy uses within an economy, providing better insights on the relation between economic growth and energy uses. It has been acknowledged as an appropriate stage for energy accounting, independent from efficiency improvements and technological progress at the different stages of the exergy flow [17].

Exergy is extensively presented in the literature as a good variable for economic and sustainability assessments of energy, as it accounts for the quality in use and conversion of energy vectors and materials [18]. Particularly, for Portugal, Serrenho et al. [18] concluded that there is an approximately linear relationship between useful exergy and the GDP throughout the period of 1960 to 2010. Practically, this means that 60 years ago we needed $1MJ$ of useful exergy per unit of GDP (1€ in 2010 prices), and the same relation still holds today.

2.2. Deep Learning

The most fundamental type of neural network is a perceptron (or a single layer of Threshold Logic Units (TLUs), figure 1). The reciprocal and most direct comparison is normally the neuron. One can simulate a set of neurons and their connections in an attempt to achieve the ability for a machine to "learn" and to "memorise" basic tasks.

Firstly, in the forward phase, a perceptron takes an array or list of numbers as inputs, computes a weighted sum of all inputs, and uses an activation function to compute the response.

The next step consists of the backward phase in which the output of the network is compared with a given target value (supervised learning). As this is a parametric model, the error between both output and target values can be minimised with an error backpropagation algorithm [20] by adjusting the weights, which are randomly initialised at the beginning. The amount by which each weight needs to be changed in order to minimise the error is unknown a priori. However, the direction in which each weight needs to be changed to reduce the error can be discovered via gradient descent method. The direction of a weight change is computed by using the sign of the gradient of the cost function with respect to each of the weights and by using a tunable learning rate parameter that quantifies the magnitude of the adjustment [21].

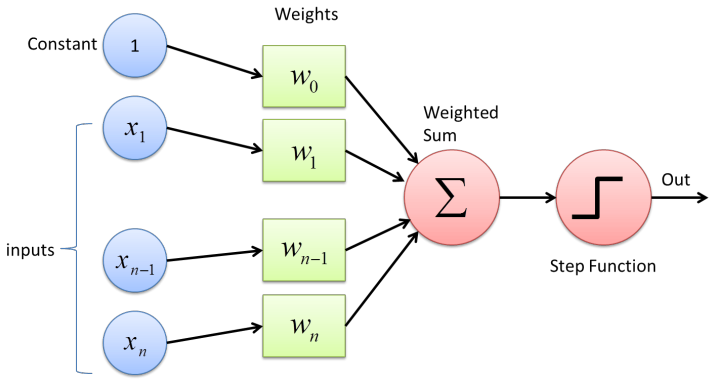

Figure 1: A TLU is an artificial neuron which computes a weighted sum of its inputs and then applies a function to that sum and outputs the result. Multiple TLUs compose a perceptron [19].

The forward phase and reverse phase are repeated for a number of iterations (epochs) until the network's error no longer decreases. A trained network is said to be a 'model' which 'encodes' the problem's domain, and the model can then be applied to unseen data and the network will produce an output in accordance with the input pattern [21].

In order to add complexity, it is possible to form layers composed of perceptrons, and when all the neurons in a layer are connected to every neuron in an immediately adjacent layer (*i.e.*, its input neurons), the layer is called a dense layer. Moreover, an extra bias feature is generally added and it is typically represented using a special type of neuron called a bias neuron, which outputs a constant value of 1, allowing a translation of the activation function. Without the presence of a bias neuron, each neuron would take the input and multiply it by a weight, with nothing else added to the equation. Their weights are estimated as part of the overall model [21].

The archetypal example of a Deep Learning (DL) model is the feed-forward deep network or Multi Layer Perceptron (MLP). Comparatively to a perceptron, a MLP is just a more complex mathematical function mapping a set of input values to output values. The function is constructed by composing simpler functions. Each application of a different mathematical function can be thought as providing a new representation of the input [22]. It is composed of one (pass-through) input layer, one or more layers of TLUs called hidden layers, and one final layer of TLUs called output layer. If an ANN contains a deep stack of hidden layers then it is called a Deep Neural Network (DNN) [21]. MLPs are widely used for classification and regression problems, and are proven to solve any continuously differentiable function to any precision making it possible to greatly benefit from them [21, 23].

The success of the model is closely dependent on the choice of good hyperparameters. The better they are, the smaller the network's error. This problem presents itself as an optimisation problem: the ultimate goal is finding the minimum of a 'black-box' function $f(\mathbf{x})$ on some bounded set $\boldsymbol{\chi}$, corresponding to diminishing the global network's error. As we do not have an analytical expression representative of the network, the analysis/evaluation is restricted to sampling. If the evaluation is cheap to perform we could sample at many combinations of hyperparameters randomly and extensively. However, if it turns out to be a computationally expensive task it is of major importance trying to minimise the number of samples.

In this work, the problem is considered through the framework of Bayesian optimisation, in which a learning algorithm's generalisation performance is modelled as a sample from a Gaussian Process (GP). This choice has been shown to outperform other state-of-the-art global optimisation algorithms on a number of challenging optimisation benchmark functions [24]. What makes Bayesian optimisation different from other procedures is that it constructs a probabilistic model for $f(\mathbf{x})$ and then exploits this model to make decisions about where in $\boldsymbol{\chi}$ to next evaluate the function, while integrating out uncertainty. The essential philosophy is to use all of the information available from previous evaluations of $f(\mathbf{x})$. This results in a procedure that can find the minimum of difficult non-convex functions with relatively few evaluations, at the cost of performing more computation to determine the next point to try. When evaluations of $f(\mathbf{x})$ are expensive to perform — as is the case when it requires training a ML algorithm — then it is easy to justify some extra computation to make better decisions [25].

2.3. Deep Learning for Time Series

Time series are collections of data points arranged in chronological order, representing temporal or spacial evolution of the dynamics of a given variable. Their analysis is intended to extract meaningful summary and statistical information, hence enabling the diagnosis of past behaviour as well as the forecasting of future behaviour [13].

Despite its predicting features, MLPs model manifests incompleteness drawn from the introduction of the dimension of time because it is no longer possible to have fixed-size inputs and produce fixed-size outputs, the causality of time is not encoded into the architecture of the model. RNNs are useful because they allow variable-length sequences as both inputs and outputs, encoding the causality of time. To circumnavigate the question arisen from the need of input of variable size data there is a simple and effective approach known as the sliding window. Basically, a data interval of a fixed size m is taken and fed into the RNN. From these m elements, one outputs the y_{m+1} element, which will be identified as the target. By continuing sliding the window across, the entire dataset will eventually be covered and input into the network. Backpropagation similarly occurs as in the perceptron, but in this case through time.

2.3.1 Recurrent Neural Networks

Essentially, a RNN cell does the exact same calculation as a perceptron, *i.e.*, a weighted sum with an activation function and it outputs a single number. In another words, it takes in a set of data over time and outputs a single number or a list of numbers over time. The major difference is that a RNN keeps a state that is passed on sequentially. In the following RNN (see figure 2), the described scheme is illustrated over all iterations.

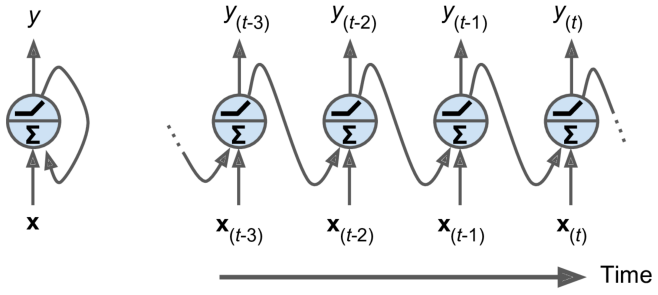

Figure 2: Recurrent neuron (left) unrolled through time (right). The notation is as follows: input vector - $\mathbf{x}_t$, output scalar - y_t, "hidden" state - y_{t-1} [21].

Stacking multiple recurrent layers of cells results in a deep RNN. Under that condition, a RNN can simultaneously take

a sequence of inputs and produce a sequence of outputs. This type of sequence-to-sequence network is useful for predicting/-forecasting time series: we feed it the data over the last n time-steps, and it outputs the data shifted by an arbitrary amount of days into the future. The advantage of this technique over sequence-to-vector or vector-to-sequence is that the loss will contain a term for the output of the RNN at every time-step, not just the output at the last time-step. Consequently, there will be many more gradients flowing through the model, and they will not have to flow only through time, they will also flow from the output of each time-step, resulting in a more stable and faster training [21].

2.3.2 Training RNNs

In order to train a RNN, we have to unroll it through time and then use regular backpropagation. This strategy is named Backpropagation Through Time (BPTT). As with regular backpropagation, there is a first forward pass through the unrolled network. In the next step, the output sequence is evaluated using a cost function. The gradients of that cost function are then propagated backward through the unrolled network. Lastly, the model parameters are updated using the gradients computed during BPTT. Note that in this case, sequence-to-sequence, the gradients will flow backward through all the outputs used by the cost function [21].

As the output of a recurrent neuron at time-step t is a function of all the inputs from previous time-steps, it is said to possess a form of memory. By definition, a part of a neural network that preserves a given state across time-steps is called a memory cell. A single recurrent neuron, or a layer of recurrent neurons, is a basic cell, or layer, capable of learning only short patterns. Because of the transformation the data goes through when passing over a vanilla RNN (see figure 2), some information is lost at each time-step. After some steps, the RNN's state contains practically no trace of the first inputs [21].

2.3.3 Tackling the Short-Term Memory Problem

However, there are different types of cells specially designed to learn longer patterns and carry a different source of information memory besides the previous output. What happens inside each type of those RNNs is different depending on the type used, but the underpinning concept is that they are mathematically designed to accumulate information (such as evidence for a particular feature or category) over a long duration, and once that information has been used, it might be useful for the neural network to dynamically forget the old state at each time-step, another programmed possibility [22]. This is what gated RNNs do. These include the Long Short-Term Memory (LSTM) and Gate Recurrent Unit (GRU) [26], which tend to be the most common. Further, the vanilla RNN already presented can too be considered a cell that sustains some memory, as already discussed.

In a LSTM cell, the state is split into two vectors, representing a short-term state and a long-term state. Instead, in a GRU cell both state vectors are merged into a single state vector. A GRU is one of the most widely used RNN cells. It is a simplified version of the LSTM cell, with fewer gate controllers and no output gate, meaning that the full state vector is the output at every time-step [13].

Both GRUs and LSTMs helped solving the exploding and vanishing gradients problem, that was present in RNNs. This problem was addressed with GRU and LSTM as a consequence of their tendency to keep inputs and outputs from the cell in tractable value ranges. This is due both to the form of the activation function they use and to the way that the update gate can learn to allow information in or not, leading to higher probability of having reasonable gradient values than in a vanilla RNN cell, which has no encoded notion of a gate [13].

2.3.4 Convolutional Neural Networks

Most modern time series analysis problems are undertaken with recurrent network structures, or, less commonly, with convolutional network structures. Convolutions are a way to capture information about the ordering of the entries on a matrix and it is described by applying a kernel (matrix) to a larger input matrix by sliding it across, forming a new, convoluted, matrix [13]. This kernel is applied repeatedly and it incorporates information about the entries' neighbours values into its own value. This is accomplished by pre-specifying a number of sets of kernels, so that different features can emerge.

Neuron's set of weights are called filters (or convolutional kernels), and a layer full of neurons using the same filter outputs a feature map, which highlights areas in a matrix that activates the filter the most. A convolutional layer can have an arbitrary number of filters, hence outputting a feature map per filter. This means that a convolutional layer simultaneously applies multiple trainable filters to its inputs, making it capable of detecting multiple features anywhere in its inputs. During training, the convolutional layer will automatically learn the most useful filters for its task, and the layers above will learn to combine them into more complex patterns. One advantage of this type of network is that they have few parameters since the same convolutional kernels are repeated over and over, meaning that there are not too many weights to train [21].

Traditional convolution is a poor match to time series because one of the main features consists on treating all spaces equally. This makes sense for images, the main area of application of Convolutional Neural Network (CNN), but it does not fit the philosophy of time series, where some points in time are necessarily closer than others. Convolutional networks are also structured to be scale invariant, however in time series we likely want to preserve scale and scaled features [13].

Nonetheless, there is an architectural transformation that includes modifications to be time aware. In a dilated causal convolution, 1D convolutional layers are stacked and the dilation rate is doubled at every layer, *i.e.*, how spread apart each neuron's inputs are. With this configuration, the first layer gets access to two time-steps at a time, while the next one sees four time-steps, the next one sees eight time-steps, and so on [21]. Therefore, the lower layers learn short-term patterns, while the higher layers learn long-term patterns, being also capable of processing arbitrarily long sequences. This also promotes model sparsity and reduces redundant or overlapping convolutions, allowing the model to look further back in time, while keeping overall computations contained [13]. To conclude, this example of dilated causal convolution introduces the notion of temporal causality by permitting only data from prior time points.

2.3.5 Hybrid Networks

Continuing with the rationale of what was explained previously, we can have a 1D convolutional layer sliding several convolutional kernels across a sequence, producing a 1D feature map per convolutional kernel. Each convolutional kernel would learn to detect a single sequential pattern with a size no longer than the convolutional kernel's size. If we use k convolutional kernels, then the layer's output will be composed of k 1D sequences, all with the same length, or, equivalently, a single k-dimensional sequence. This means that we can build a neural network composed of a mix of recurrent layers, such as LSTM or GRU, and 1D convolutional layers. By using the zero padding method, the output sequence will have the same

dimensions as the input sequence. Alternatively, it is possible to use a stride greater than 1, downsampling the input sequences. By shortening the sequences, the convolutional layer may help the recurrent layers detect longer patterns [21].

3. Implementation

3.1. Framing the Problem

The first objective is to explain the relationship between the distribution of land use in terms of final exergy in the agricultural sector and economic growth explained in terms of GDP. Concretely, we fed a neural network with 6 features, *i.e.*, all the land use classes but one (5 were available in total) plus final exergy data for agricultural sector, and GDP. Then, the neural network was set to output the 4 classes of land use and the 5^{th} was then determined, as they summed up to 1. By excluding one feature from the training, we were promoting smaller optimisation times. To understand how the energy and economic factors affect land use distribution, data regarding energy and economy was lagged backwards one time period (1 year) in relation to the land use data. We assumed that the direction of influence is *past* (exergy + GDP) $\rightarrow$ *future* land use distribution with support on the argument that patterns of energy consumption and economic development in the past dictate how the farmer is going to utilise the land in the future. Money and energy are intuitively beforehand required to consummate significant modifications in land and infrastructures. Useful exergy usage was demonstrated to be positively correlated with GDP [18], therefore, as the GDP already explained the dynamics of useful exergy, final exergy was chosen to further help modelling the dynamics in the sector.

Secondly, with aid of the framework already established in the land use modelling, neural networks were fed 8 features and retrained. Those features were split into land use (4), total final exergy usage and GDP for Portugal (2), and data regarding emissions from the agricultural and energy sectors (2). As the goal is to explain those emissions, the neural networks were set to output 2 values. Here, the lagging of the data was differently done. The data regarding land use and GHG emissions was lagged backwards one time period (1 year), in relation to the energy and economic variables. The intuition behind lagging the land use distribution data is that decisions regarding land use management are expected to affect land use emissions in the following years. If, for instance, agriculture area is massively converted into forest area, we will observe reduced emissions from the forest for years after the conversion, but not necessarily in the year of the conversion. The effect is anticipated to happen starting only in the following year. The inclusion of land use data in this model was due to the fact we are trying to forecast GHG from agriculture, and land use data contains information concerning that matter. In terms of data about total final exergy, its use is justified because we are trying to forecast GHG from energy sectors. All models involved can be seen as autoregressive, because their outputs depends on observations from previous time-steps.

3.2. Data

The main data and respective known sources available for this work were organised into land use and land cover statistics, energy, economy, and policies. The first subject referred to 1) permanent pasture area, permanent culture and arable land area, forest area and urban/artificial area, and 2) land use emissions. The energy data consisted on 3) final exergy [27]. Economy was represented by 4) GDP growth rate [28]. Lastly, policies data [29, 30] concerned the following: 5) wheat campaigns and policy reform, 6) private forestation policy, 7) agrarian reform and 8) agricultural transitory measures and policy reform. As the goal of the thesis was to include a historical perspective from the 1960's onwards, but the data available was insufficient to depict long trends, data reconstruction techniques were applied when necessary, as explained in each sub-section below, and supported with historical policies.

3.3. Methods

For reproducibility, the software's versions of the programming language and main libraries used in the project were Python 3.8.4, Keras 2.3.1, Tensorflow 2.3.0, Sklearn 0.22.2.post1, Scipy 1.4.1, Kerastuner 1.0, Numpy 1.18.5, Pandas 1.0.4, and Matplotlib 3.2.1.

As neural network algorithms are stochastic, there is randomness associated, such as when initialising the network's random weights. Hence, for complete reproducibility, the seed of all the random generators involved in the dynamics of the multiple algorithms were set constant. By doing this, we guarantee stable and repeatable results. Neural networks use randomness by design to ensure they effectively learn the function being approximated for the problem, as this class of ML algorithm performs better with it.

To keep track of all the dynamics, the code was built modularly, having multiple scripts performing different tasks. This enables the reproduction of precise transformations on any dataset and it results in the construction of libraries of transformation functions that could be reused in future projects. The built modules are a data transformation pipeline that shapes the data into the appropriately expected format, an ensemble of neural networks for Bayesian optimisation, a hyperparameter tuner based on Bayesian optimisation, a score iterator for a rigorous exploration of all details within all trials, and utilitarian tools for data analysis and visualisation.

3.4. Performance Estimation

In a forecasting task it is extremely important to be capable of estimating an error that a predictive model will incur on unseen data. The time-agnostic solution for validation in non-serialised data needed to be adapted for time series, where dependency among observations was expected, instead of independent and identically distributed data.

Initially, a time series was split into two parts: the first part served as an initial fitting period in which a model is trained. This part was further split into training and validation sets. The last part of the time series was used for testing on unseen observations and then estimating the true loss of the model.

Within this method, it is possible to adopt different strategies regarding training/testing split point and growing or sliding window settings, for example. The best way of getting robust estimates of predictive performance is to employ these strategies in multiple test periods, generating more unbiased measures of the generalisation error. Figure 3 depicts the algorithm behind the expected error estimation.

First, by fold, and excluding the n^{th} fold, a model m was built on the training set and the loss estimate that a predictive model m would incur on new observations was computed on the validation set (green extension on Figure 3). Reiterating the process, a set of models was generated and sorted by their loss estimate. Second, in order to unbiasedly evaluate the estimations produced by the best model's configurations, the top models were re-tested using the testing set (red extension on Figure 3). Effectively, we obtained a measure of the error $\boldsymbol{\delta}$, the ground true loss that a model m incurred on new data.

Across each model's type, the performance estimation method was conceived with evaluations in two dimensions: 1)

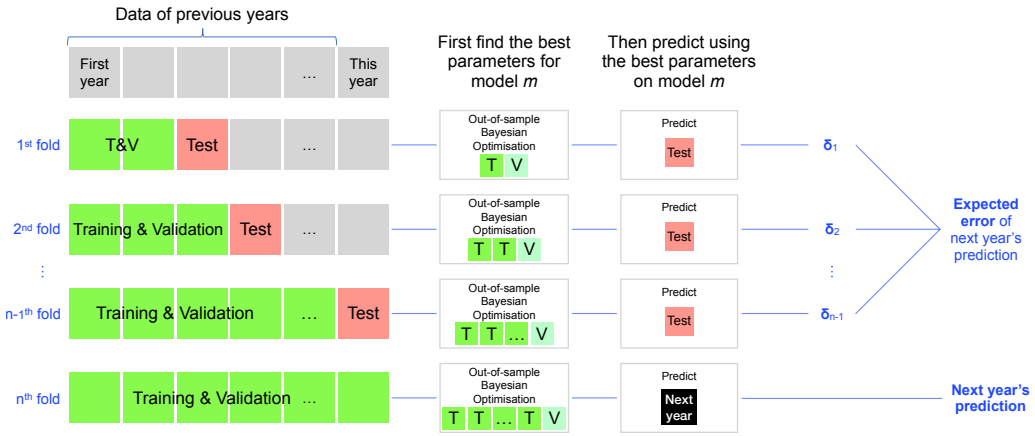

Figure 3: Error estimation algorithm based on an out-of-sample growing window approach.

preliminary loss estimate, by converging to the ideal architecture's set up – this measured the magnitude of the difference between the estimated and the actual error on all models; and 2) final loss estimate, by repeating the previous measurement on new data and on the best models. One could think that the generalisation error (out-of-sample error) from the evaluation of the model on the validation data would suffice to infer an error for the prediction, however it is extremely important to note that by selecting the best configuration based on the results from the validation data we were incurring in an implicit overfitting of the model to that data. Hence, if we want to avoid being biased it was mandatory to compute a second error by evaluating the model on the testing data set.

Ultimately, the process was repeated on the n^{th} fold, the difference is that no final loss was estimated because we were trying to forecast unregistered data, so the past final loss estimates would be the expected error incurred on this data. It is arguable that we were not using all the available data for training and that the predictions were made with a chronological gap, non-consecutively. Nonetheless, as the main interest was to forecast future scenarios, the model should be able to predict several time-steps ahead. Thus, this algorithmic proposal made it more resilient to fit the aim.

For each model m and fold n_{fold}, three prediction matrices for the testing data were generated, one per sub-model s. Next, the Root Mean Squared Error (RMSE) was calculated column-wise for all prediction matrices. These calculations resulted in an error vector for all sub-models.

These exploratory results served the purpose of diagnosing which models or sub-models should proceed in the analysis. There were two acceptability criteria: 1) the Mean Squared Error (MSE) from each sub-model on validation data shall not surpass the MSE obtained from the baseline model; 2) the total RMSE for each sub-model on testing data shall be inferior to 15%, which represented an average tolerance of 3% of error per land use category (there are 5 categories), and 7.5% per emission category. For each of the remaining candidate models, an average of the prediction matrices from the respective sub-models was computed.

The aggregate results represent the expected response for each model to this type of data. Reducing a model to a sample sub-model would be insufficiently representative of how well it predicts, hence the ensemble of sub-models for intermediate predictions with respective errors and posteriorly the ensemble of models utilised for the final prediction. A richer response is expected to arise with the contributions of multiple sources of models.

Now, a vector composed of the expected errors for each category was estimated. Lastly, all errors were combined across all folds and models, resulting in global error vector. These values would be the uncertainty associated with all models' averaged out prediction for each feature. In the end, this committee of models was expected to produce a robust estimation of the average prediction matrix optimised in the last existing fold. To this forecast, the expected error $\boldsymbol{\delta}$ would be associated.

3.5. Training Procedure

The data was sub-divided into 5 folds with the growing window method, and the last fold was constituted only by training and validation data. Oppositely to what happens in the training set, both validation and testing set preserved their absolute size for all folds. For each fold, the flatten data was transformed into batches and normalised taking into account just the training set for the scaling factor.

All 6 models hyperparameters' boundaries were specified, the models were then ready for instantiation. Some worth mentioning are the maximum number of intermediate layers and the maximum number of neurons for each of those layers, which were set to 3 and 100 (with steps of 5), respectively, for all models where applicable. The convolutional layers had a maximum of 15 filters (with steps of 2) and a kernel with maximum size of 6. The learning rate was universally set to start at 0.2.

For every model and for every fold, Out-Of-Sample (OOS) Bayesian optimisation was deployed, enabling the search of the hyperparameters that best generalise for both training and validation data. A maximum of 1000 epochs was set as well as 1500 trials (maximum) and 300 random generated samples for the initial training. Within every model and fold's search, the best 3 configurations achieved were saved, as there could be innumerable sets of hyperparameters (local minima) that could conduct to good results. Because of this choice of picking the best models, there was an implicit overfitting, so it was favourable to have more models to generate more statistics for the expected error of a forecast. This generated 3 measures of the error for every combination of model m and fold n_{fold} (excluding $n_{fold} = 5$), 72 in total. How these measures of error were treated is explained in section 3.4;

Lastly, the same optimisation was redone on the last fold, $n_{fold} = 5$, which included all data for training and validation. 3 forecasts per model were made and averaged out if previously all models presented an acceptable generalisation power. Otherwise, only the best architectures for our type of data were included.

3.6. Forecasting Procedure

As at the time of writing we were approaching the end of the year of 2020, solid estimations of the GDP for 2020 already existed [31], hence they were included in the data considered for forecasting. From this point onwards, new data was

adopted from two scenarios for Portugal in 2030 developed in the MEET project report [32], in order to extend the final exergy data, available up to 2016, and to extend the GDP data that stopped at 2020. These scenarios are product of a collaborative partnership between industry leaders from around 30 Portuguese companies, from 13 different sectors, representing approximately 20% of the nation's GDP, and government agents. Plausible trajectories for the Portuguese economy in the context of the fourth industrial revolution (stronger linking between technologies and the physical world) were disputed.

A more pessimistic future marked the first scenario, from now one denoted by scenario P, which was characterised by evolutionary stagnation. As a consequence, both exergy and GDP steadily (but not drastically) decreased until 2030 [32].

Antagonistically, a more optimistic scenario, here denoted by scenario O, described a brighter future where Portugal excelled in growth and evolution, resulting in a continuous increase of both final exergy consumption and GDP [32]. Note that the GDP time series included a severe economic contraction in 2020, consequence of the 2019/2020 beginning of a worldwide pandemic, caused by a new coronavirus. Supplementary, two sets without the new coronavirus influence were created, with the properties stated for scenarios P and O.

In order to deepen the analysis on the dependency of the model on the input parameters, 4 additional sets of data were built. The first set consisted on constant GDP values starting at 2017, and yearly decreases of 5% in the exergy values. For the second set, the modifications were performed in the opposite direction, with an annual increase of 5% for the exergy. The third and forth sets are identical to the previous ones, the difference relies on the fact that the changes are mode on swapped time series, *i.e.*, the exergy time series was kept constant and the GDP was varied accordingly.

4. Results and Discussion

4.1. Model Performance

4.1.1 Land Use Distribution

In all the folds, the causal convolutional model had the worst training MSE when compared to its peers, and in all folds, except for one, it was outperformed by the remaining models. This suggests that either this model does not fit well to our data or stronger regularisation techniques such dropout, to mention one, or a more varied layer initialisation, with different number of filters and/or kernel size within intermediate layers, needs to be implemented to enhance the training/learning process.

Apart from the first fold, we obtained a maximum error per category of 1.3%. One critical component of the error vector was the category Urban/Artificial, because of its low absolute value ($\sim 5\%$) in the data it was important that the error would be the lowest of them all. This fact is substantiated across all folds with an error below 0.4% except, again, on the first fold. Nonetheless, its value on first fold is quite close to being the lowest and to the ones obtained in the other folds: 0.5%.

Ultimately, the attained error vector was set at

$$\boldsymbol{\delta}_{LULC} = [1.4, 1.7, 0.5, 0.4, 1.2]\ (\%)\ . \tag{1}$$

which is expected to prevail up to a 7-year long prediction if the evolution of statistical dynamics of the input variables do not drastically change in the future. This corresponds to the temporal length established during the batching process. In other words, the model was explicitly trained with a 7 time-step horizon to generate sequences with that length in which the last instance (seventh time-step) represents a forecast. Also, let us remember that the forecasting exercises and respective errors estimation is performed on data non-consecutive to the data used for training. Therefore, this value of 7 years would in principle represent the minimum admissible validity for the error of equation 1 to be considered. Adding the number of batches from the testing set, 5, the maximum admissible validity is arguably 12 years. The achieved magnitude of the errors is at par with the measurements' errors reported in [33].

The results from the first fold and from the causal convolutional model were included, so the error is expected to be majorated. The prediction for earlier steps will usually be more accurate than the predictions for later time-steps, further than that errors might accumulate.

4.1.2 Greenhouse Gases

For the application of the previous framework to the data regarding emissions, the error/predictive performance analysis was repeated. Here, the acceptance rate of sub-models, within the criteria set in section 3.4, decreased appreciably.

Earlier, the acceptance rate was set at 93%, with 67 measures of the error that culminated in the error vector presented in equation 1. Here, overall the Bayesian search denoted much more difficulties in finding appropriate sets of hyperparameters that would result in well fitted models to the data, with just 50% of the sub-models being considered as valid. The most notorious case is the one respecting the causal convolutional architecture. For that type of network, only two sub-models (out of a universe of possible 12) fell under the 15% threshold for all data folds. Low performance issues relating to the convolutional architecture were found as well in section 4.1.1. Consequently, that architecture's results were rejected and no forecast was attempted with architecture #4. The best performant models with this data across all data folds were the LSTM, GRU, and hybrid convolutional 1D-LSTM networks.

These results might be an indicator of either insufficient covariates to fully explain GHG emissions, inadequate lagging, or that in more recent periods the statistical properties of data regarding GHG emissions have greatly changed, with those new dynamics suggesting a transition to different emitting patterns than those of the past, and consequently complexing the learning process. The last argument could be also supported on the fact that the training scores achieved on the validation data do not differ appreciably from the scores previously displayed in section 4.1.1. Nonetheless, 34 measures of the error contributed to the final error estimation from equation 2, divided into emissions from agriculture and energy sectors, respectively, legitimating the joined forecasting capability of all models, excluding the left-out convolutional architecture. The error vector,

$$\boldsymbol{\delta}_{GHG} = [0.2, 4.2]\ (\text{Mt } CO_2\text{e})\ , \tag{2}$$

relatively to the values registered in the year of 2016, represents a deviation of 3.2% and 8.3%, respectively.

4.2. Model Forecasting

4.2.1 Land Use Distribution

A generalised stagnation of the evolution of land use distribution is evident. In spite of the good fitting, the error from 1 exceeds largely the variation scale of the variables in this scenario. Hence, error bars were not added to slowly varying trends, focusing the analysis on the qualitative side.

It is remarkable that even though we have a model very well fitted to the data, its estimated (already low) uncertainty is greater than the expected variations in the variables for an extreme plausible scenario. This fact confirms the inherent complexity and uncertainty in this forecasting exercises, despite

the rigorous treatment. This also hints that simpler methods may be subject to stronger scrutiny.

In our simulated data, the GDP decreases up to lower levels than those of the MEET project report [32] for the pessimistic scenario. The effect is due to the fact that in 2020 the pandemic introduced some severe volatility in the Portuguese economy, rendering an abrupt divergence from historical values in terms of GDP. As the utilised GDP value for the year of 2020 got closer to the worst economic foresaw case for 2030, it may be understandable that no major dynamics were reflected in the land use, partially validating the argument used in the problem framing (see section 3.1) that economic resources were required to consummate significant modifications in land and infrastructures.

In turn, the GDP values from the optimistic scenario did not achieve levels as high as those from the MEET project report [32], due to the lower state of the starting point. In our results, towards the year of 2030 some trends are already notorious. Nonetheless, this economic lower starting point may explain the slow departure from stability, hindering land use changes.

Firstly, there is a slight increase in the forest area. Secondly, there are non negligible changes in both permanent pasture area, and permanent culture and arable land. While the former yielded a higher area, the later steadily started decreasing. Thus, there is an evidence pointing towards the reduction of cropland (permanent culture and arable land) area with economic acceleration.

In regards to the effects the pandemic introduced in the final results, the most striking difference was that in the economic optimistic scenario the reduction in cropland land area was much more pronounced ($\approx$ 1% less area than in the original scenario). That difference was mainly transferred into gains for grassland (permanent pasture) area.

The data sets that had the GDP varied were the major contributors to new dynamics. It is clear that for our model, the coupling between the GDP and the output variables is stronger than with the final exergy. In the case where the GDP growth was accelerated, cropland area sharply decreased, approximately 4%, and forest area, grassland area, and urban/artificial area steadily increased by 3.1%, 1.1%, and 0.4%, respectively, when compared to 2017. The opposite trends were observed when the GDP growth was inversely accelerated (economic contraction).

4.2.2 Greenhouse Gases

In the GHG emissions results, two interesting patterns arose in both scenarios. For the economic pessimistic scenario, the emissions from energy tended to stall, whereas the emissions from agriculture rose, though negligibly. In the optimistic case, emissions from agriculture decreased slightly, while the emissions from energy sharply increased - roughly 10% more GHG emissions than that of scenario P in the forecast of 2030. One might argue that there is a trade-off where an increasingly richer country starts to abandon agriculture production, and it uses activities that consume more energy, oppositely to the case of an increasingly poorer country that operates less high-energy activities, and where the agriculture production gains some momentum.

Here, it was concluded that in the coronavirus-free built scenarios, emissions from the energy sector increased, with final greater values in 2030 than those of the original scenarios. The biggest fluctuation was observed for the emissions from energy in the optimistic scenario. Overall, it was evidenced that COVID-19 might mitigate the increase in emissions of CO_2e to the atmosphere up to 2030, under scenarios P and O.

In terms of coupling, it was observed that total final exergy data was responsible for introducing major variations, having 2017 as reference. On the case where final exergy grew year-over-year 5%, emissions from agriculture deeply decreased, close to 4%, and emissions from energy sharply increased, 30%. For the decreasing final exergy time series, the new exergy dynamics induced greater emissions from agriculture ($\approx$ 4%) and a steep decrease in emissions from energy ($\approx$ 12%). The GDP time series did not contributed to significant modifications, when compared to the original forecast. The results suggest that, operating at the business as usual mode, without additional politics, generating more wealth and consuming more energy comes at the expense of greater GHG emissions [17, 32].

5. Conclusions

5.1. Findings and Achievements

The major findings were that: 1) there is a strong coupling between GDP and the dynamics of land use distribution, as well as between total final exergy and GHG emissions from both agriculture and energy sectors, 2) when a country gets richer (higher GDP) it tends to decrease its cropland area while increasing the intensity of activities that consume more energy, emitting GHG at higher levels, 3) in ever increasing impoverished countries cropland area grows and energy intense activities are reduced, leading to overall lower levels of GHG emissions, 4) the coronavirus pandemic might decrease the cropland area reduction in a forecasted positive economic scenario, and 5) the coronavirus pandemic might have induced a slight deceleration for emissions related to agriculture and energy sectors up to 2030.

It was also evident that the pre-COVID-19 scenarios, from collaboration between companies' representatives and government agents, were not sufficiently aggressive (within a plausible pathway) when combined with data that took into account the economic downturn. These new economic conditions hindered the evolution of land use in both studied scenarios, being required a sensitivity analysis to further explore the interplay between all factors.

In the conditions of the contemplated scenarios, the forecasted evolution for 2030 of land use distribution and GHG emissions was tenuous when compared with the estimated uncertainty. Despite the very well fitted models and rigorous uncertainty estimation, the compound evolution of the plausible scenarios stayed always within the uncertainty, complexing the analysis. This lead to casting doubts over simpler methods.

Methodologically, a new performance estimation method for time-series was developed within an ensemble of models environment. This method involved measures in two dimensions: 1) preliminary loss estimate on validation data, by converging to the ideal architecture's set up; and 2) final loss estimate, by repeating the previous measurement on new data and on the best models. Hence, we got control over two potential overfitting situations, being them the overfitting to the training data, and the implicit overfitting of the model to the validation data when selecting the best model's configurations. As we are forcing the model to predict several time-steps ahead and estimating the error on that prediction, 7 years would in principle represent the minimum admissible validity for the uncertainty to be considered. Adding the number of batches from the testing set, 5, the maximum admissible validity is arguably 12 years.

The insufficiency of data implied the creation of a new batching method. It is usually advisable to use hard frontiers between the training, validation, and testing data sets. However,

in time series analysis, when using a sequence-to-sequence configured network for predicting m time-steps ahead, the error is only accounted for that last prediction. Therefore, overlapping data sets were constructed in a way that no lookahead was introduced, maximising the batches cardinality.

The originality of our work came from the fact that in the literature this is the first study where there is a compounded approach to the problems of forecasting land use change and associated emissions, by coupling both modelling exercises into an integrated framework, hence strengthening the predictive power. Commonly, land use changes and GHG emissions have been individually forecasted with ML, by directly inputting either land use distribution data and related data or GHG emissions data and related data to the models, respectively [34, 35].

5.2. Limitations and Hypotheses

The foremost limiting aspect came from the scarcity of historical data regarding land use distribution. Even for existing data, there was considerable uncertainty. The second most influential limitation resulted from the fact the time scale of this study contributed to shorter time series. Given the nature of the models employed, longer series are preferable in order to increase the likelihood of revealing patterns.

On the technical side, due to training data constraints, we have only used stateless RNNs, where at each training iteration the model starts with a hidden state full of zeros, then it updates the state at each time-step, and after the last time-step it throws it away. It is possible to preserve this final state after processing one training batch and use it as the initial state for the next training batch. By doing so, the model could learn longer long-term patterns even though it only backpropagates through short sequences. This defines what it is called as stateful RNNs and they make sense in a context of abundant data, because the condition for this model to be used is that each input sequence in a batch starts exactly where the corresponding sequence in the previous batch left off, therefore it is required to use sequential and non-overlapping input sequences.

Another major constraint was that the training times were quite high, with an estimation of 30h per thread per fold of data per model. With 24 threads available, the running trials had to be capped, with roughly 1-5% of the search space studied during the Bayesian optimisation. As a consequence, we had to compromise in terms of which covariates would be included.

In terms of hypotheses, with support on studies in the area, the two hypotheses conjectured were that: 1) land use distribution is, besides to its own lagged values, tightly related to both energy usage and economic development, and 2) GHG emissions can be defined in terms of its own lagged values, land use and land cover, energy usage, and economic development. Also, in autoregressive models there is the assumption that the observations at previous time-steps are useful to predict the value at the next time-step are made, *i.e.*, there is a correlation between variables. It was also hypothetised that GDP as an economic indicator would reliably help reflecting land use changes as well as GHG emissions for the agriculture and energy sectors. This indicator has been criticised for ignoring the depreciation of assets, non-market economy, damages to the environment, and for being a poor proxy for societal well-being [36, 37].

5.3. Future Work

Due to the intricate nature of the dynamics of models in the field of nature-related processes, this work was carried out based on empirical evidences in regards the relationship between the input features and the targets, meaning that it was assumed that the input variables to our model contained useful information that would allow us to predict the target variables based on those features. In the future, a thorough study on statistical relationships, such as correlation, autocorrelation, spurious correlations, among other statistics, shall be done for a broader set of covariates. Specific data-analysis techniques on multivariate time series, such as Principal Component Analysis (PCA), similarity searches, feature-subset-selection, and clustering might be a good start [38]. Even though correlation does not imply a causal relationship, and non-linear causal relationships are difficult to demonstrate, such tests might lead to new directions on which features are the most relevant and how they are temporally coupled.

Besides, there is a huge performance dependency on how the splitting and batching processes are done, as well as on how the time series are lagged. By plotting the model's errors on the validation set across time it would be assessable if the model performs better on the first part of the validation set than on the last part, if so, then it would be advisable to shorten the window size, training the model on a shorter time span. However, to automate the process the ideal would be to apply a second Bayesian optimisation for the the window size, splitting point, and lagging period hyperparameters. Additionally, a new, non-default, set of exploitation/exploration parameters for both Bayesian optimisations is recommended.

In this work, we evaluated models' loss according to two distinct metrics: RMSE and Mean Absolut Error (MAE). In the future, the inclusion of the Huber loss function could be considered. Likewise, more sophisticated overfitting-preventing/regularisation techniques present a great potential. In the list, one could mention the L1 and L2 regularisation, dropout, recurrent dropout, and Monte Carlo dropout. Architecture-wise, alterations that could render better results are the introduction of customised output layers, different activation functions in different layers, the use of different number of filters and kernel size within convolutional layers, and trying to use of a vanilla 1D-convolutional model. Additionally, computational time could be purchased and Graphics Processing Units (GPUs) could be used to increase the number of trialled models in the case of larger batches. Finally, having a method to estimate forecast errors for longer periods would come handy.

References

[1] Vaclav Smil. *Energy and Civilization: A History*. The MIT Press, rev - revised, 2 edition, 2017. ISBN 9780262035774.

[2] Karen Seto et al. Hidden linkages between urbanization and food systems. *Science*, 352(6288):943–945, 2016. ISSN 0036-8075. doi: 10.1126/science.aaf7439.

[3] Amy K Styring et al. Isotope evidence for agricultural extensification reveals how the world's first cities were fed. *Nature Plants*, 3(6):17076, 2017.

[4] Guillermo Algaze et al. Initial social complexity in southwestern asia: the mesopotamian advantage. *Current Anthropology*, 42(2):199–233, 2001.

[5] David R Montgomery. Is agriculture eroding civilization's foundation? *GSA TODAY*, 17(10):4, 2007.

[6] Dorian Q Fuller and Chris J Stevens. Between domestication and civilization: the role of agriculture and arboriculture in the emergence of the first urban societies. *Vegetation history and archaeobotany*, 28(3):263–282, 2019.

[7] O. Bertolami and F. Francisco. A physical framework for the earth system, anthropocene equation and the great acceleration. *Global and Planetary Change*, 169:66 – 69, 2018. ISSN 0921-8181. doi: https://doi.org/10.1016/j.gloplacha.2018.07.006.

[8] Gerrit Hoogenboom et al. Chapter 13 - biophysical agricultural assessment and management models for developing countries. In Charles A.S. Hall, Carlos Leon Perez, and Gregoire Leclerc, editors, *Quantifying Sustainable Development*, pages 403 – 422. Academic Press, San Diego, 2000. ISBN 978-0-12-318860-1. doi: https://doi.org/10.1016/B978-012318860-1/50020-4.

[9] Nanyang Zhu et al. Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *International Journal of Agricultural and Biological Engineering*, 11, 07 2018. doi: 10.25165/j.ijabe.20181104.4475.

[10] M. Ejaz Qureshi et al. A biophysical and economic model of agriculture and water in the murray-darling basin, australia. *Environmental Modelling & Software*, 41:98 – 106, 2013. ISSN 1364-8152. doi: https://doi.org/10.1016/j.envsoft.2012.11.007.

[11] Rita Alves. Aplicação de modelos de redes neuronais para previsão de consumos de energia [bachelor's thesis]. *Lisbon, Portugal: Instituto Superior Técnico, University of Lisbon*, 2016.

[12] Pang-Ning Tan et al. *Introduction to data mining*. Pearson Education India, 2016.

[13] Aileen Nielsen. *Practical Time Series Analysis*. O'Reilly Media, Inc., 1 edition.

[14] Konstantinos G. Liakos et al. Machine learning in agriculture: A review. *Sensors*, 18(8), 2018. ISSN 1424-8220. doi: 10.3390/s18082674.

[15] Andreas Kamilaris and Francesc X. Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70 – 90, 2018. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2018.02.016.

[16] Ricardo Manso et al. The way forward in quantifying extended exergy efficiency. *Energies*, 11(10), 2018. ISSN 1996-1073. doi: 10.3390/en11102522.

[17] A Serrenho. Useful work as an energy end-use accounting method: historical and economic transitions and european patterns [phd dissertation]. *Lisbon, Portugal: Instituto Superior Técnico, University of Lisbon*, 2013.

[18] André Cabrera Serrenho et al. Structure and dynamics of useful work along the agriculture-industry-services transition: Portugal from 1856 to 2009. *Structural Change and Economic Dynamics*, 36:1 – 21, 2016. ISSN 0954-349X. doi: https://doi.org/10.1016/j.strueco.2015.10.004.

[19] Sagar Sharma. What the hell is perceptron? https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53, last accessed on 31/12/19.

[20] David E Rumelhart et al. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[21] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*. O'Reilly Media, Inc., 2 edition.

[22] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[23] Roberto Lopez and Eugenio Oñate. A variational formulation for the multilayer perceptron. In *International Conference on Artificial Neural Networks*, pages 159–168. Springer, 2006.

[24] Donald Jones. A taxonomy of global optimization methods based on response surfaces. *J. of Global Optimization*, 21: 345–383, 12 2001. doi: 10.1023/A:1012771025575.

[25] Jasper Snoek et al. Practical bayesian optimization of machine learning algorithms. 2012.

[26] Junyoung Chung et al. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[27] André Cabrera Serrenho et al. Decomposition of useful work intensity: The eu (european union) - 15 countries from 1960 to 2009. *Energy*, 76:704 – 715, 2014. ISSN 0360-5442. doi: https://doi.org/10.1016/j.energy.2014.08.068.

[28] Organisation for Economic Co-operation and Development (OECD). OECD-FAO Agricultural Outlook. Last accessed on 11/01/20.

[29] Carlos A Guerra et al. Policy impacts on regulating ecosystem services: looking at the implications of 60 years of landscape change on soil erosion prevention in a mediterranean silvo-pastoral system. *Landscape ecology*, 31(2): 271–290, 2016. doi: 10.1007/s10980-015-0241-1.

[30] Nádia Jones et al. Historical review of land use changes in portugal (before and after eu integration in 1986) and their implications for land degradation and conservation, with a focus on centro and alentejo regions. *Applied Geography*, 31(3):1036 – 1048, 2011. ISSN 0143-6228. doi: https://doi.org/10.1016/j.apgeog.2011.01.024.

[31] European Comission. Macro-economic database AMECO. Last accessed on 8/12/20.

[32] António Alvarenga. Towards a carbon neutral economy how is portugal going to create employment and grow? – technical report, 2017.

[33] ICNF. 6º inventário florestal nacional. page 1, 2019.

[34] Muhammad Hadi Saputra and Han Soo Lee. Prediction of land use and land cover changes for north sumatra, indonesia, using an artificial-neural-network-based cellular automaton. *Sustainability*, 11(11):3024, 2019.

[35] Abderrachid Hamrani et al. Machine learning for predicting greenhouse gas emissions from agricultural soils. *Science of The Total Environment*, 741:140338, 2020.

[36] Saša Stjepanović et al. A new approach to measuring green gdp: a cross-country analysis. *Entrepreneurship and sustainability issues*, 4(4):574–590, 2017.

[37] James D Ward et al. Is decoupling gdp growth from environmental impact possible? *PloS one*, 11(10):e0164733, 2016.

[38] Kiyoung Yang and Cyrus Shahabi. On the stationarity of multivariate time series for correlation-based data analysis. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.