Article

Improving Land Cover Classification Using Genetic Programming

João E. Batista [1,*], Ana I. R. Cabral [2], Maria J. P. Vasconcelos [2], Leonardo Vanneschi [3], Sara Silva [1]

[1] LASIGE, Faculty of Sciences, University of Lisbon, Campo Grande, 1749-016 Lisbon, Portugal
[2] Forest Research Centre, School of Agriculture, University of Lisbon, Tapada da Ajuda, 1349-017, Lisbon, Portugal
[3] NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312 Lisbon, Portugal
[*] Correspondence: jebatista@fc.ul.pt

Abstract: Genetic Programming (GP) is a powerful Machine Learning (ML) algorithm that can produce readable white-box models. Although successfully used for solving an array of problems in different scientific areas, GP is still not well known in Remote Sensing. The M3GP algorithm, a variant of the standard GP algorithm, performs Feature Construction by evolving hyper-features from the original ones. In this work, we use the M3GP algorithm on several satellite images over different countries to perform binary classification of burnt areas and multiclass classification of land cover types. We add the evolved hyper-features to the reference datasets and observe a significant improvement of the performance of three state-of-the-art ML algorithms (Decision Trees, Random Forests and XGBoost) on the multiclass classification datasets, with no significant effect on the binary classification ones. We show that adding the M3GP hyper-features to the reference datasets brings better results than adding the well-known spectral indices NDVI, NDWI and NBR. We also compare the performance of the M3GP hyper-features in the binary classification problems with those created by other Feature Construction methods like FFX and EFS.

Keywords: Genetic Programming; Evolutionary Computation; Machine Learning; Classification; Multiclass Classification; Feature Construction; Hyper-features; Spectral Indices

1. Introduction

The importance of forests, and the need to preserve them, is well established in the scientific community and in societies at large. Forests contribute to the regulation of biogeochemical cycles and play a crucial role in the mitigation of anthropogenic carbon emissions, while hosting essential biodiversity with species that are unable to adapt to other habitats. Additionally, many of the world's poorest rural communities depend on forests for survival.

The need to reduce carbon emissions, and the awareness that deforestation accounts for approximately 11% of global carbon emissions, led the international community to find mechanisms that create financial value for the carbon stored in forests, thus encouraging the maintenance of standing trees, as opposed to harvesting for economic return or other uses. One of the main mechanisms in place is the Reducing Emissions from Deforestation and forest Degradation (REDD+) developed by Parties to the United Nations Framework Convention on Climate Change (UNFCCC). Under this mechanism, developing countries can receive results-based payments from avoiding deforestation and forest degradation, forest conservation, sustainable management and enhancement of forest carbon stocks [1]. However, for these mechanisms to be effective, they require the ability to Measure, Report and Verify (MRV) performance.

Remote Sensing (RS) is recommended as an appropriate MRV technology since the establishment of the Warsaw Framework in 2013 [2]. However, many difficulties, from the spatial and temporal resolution of freely available satellite imagery to data processing power, have been hindering the full use of this technology. Now, with the evolution of Earth Observation systems (with provision of higher spatial and temporal resolution images) and with novel open-data distribution policies, there is an opportunity of applying Machine Learning (ML) to induce models that automatically identify land cover types in satellite images.

Previous ML work in land cover classification has been very successful. One simple practice that helps obtain good results is the inclusion of spectral indices as additional independent variables[3] in the reference dataset. Spectral indices are combinations of reflectance values from different wavelengths that represent the relative abundance of certain terrain elements. They have been used by the RS community for a long time to enhance the identification of vegetation (e.g., NDVI [1]), water (e.g., NDWI [2]), burnt areas (e.g., NBR [3]) and many other elements. Over the years, many indices were created and adapted to accommodate the particularities of different images. In the case of vegetation indices, this number is so vast that over one hundred of them were reviewed in [4].

Like indices, hyper-features are mathematical expressions that combine the original features of the data (the independent variables) with the goal of representing data properties that facilitate the learning of ML models. Spectral indices are, in fact, particular cases of hyper-features. Ideally, the hyper-features should be simple and meaningful, allowing the RS experts to easily understand the ML models that are based on them, or to directly use them in image analysis software to visualise what they represent.

Notwithstanding the success of ML methods when performing classification of satellite imagery, the reported results are often obtained by applying a model in the same images where it was trained (e.g., [5–7]), or in an image time series from the same location (e.g., [8–10]). Training models to be ready to be used outside their training images is not a trivial task due to the radiometric variations between different images. These variations can arise from multiple sources, such as the difference in the angle of the solar incidence on the ground; the weather; the conditions of the terrain; the type of terrain; or the growth stage of the vegetation. Spectral indices are also sensitive to these variations, despite the efforts to increase their robustness.

Our goal is to improve satellite imagery classification, by creating hyper-features that increase the performance of ML algorithms. In previous work [11], we used a Genetic Programming (GP) [12] classifier called M3GP [13] to evolve hyper-features that, when used instead of the original ones, were able to improve the accuracy of different ML algorithms in binary classification of images different from the ones used in training (although, for unseen data of the same images, there was no significant effect). GP is a powerful ML evolutionary algorithm that can produce readable white-box models. Successfully used for solving an array of problems in different scientific areas, GP is, however, still not well known in RS. The M3GP algorithm is a variant of standard GP that was originally developed as a multiclass classifier, but later used as a Feature Construction method for other algorithms, both for classification and for regression [11,14,15]. Creating hyper-features from one image and using them for classifying a different image falls under the area of Transfer Learning [15], which attempts to use knowledge from one problem to solve another similar problem.

In this work, we perform a thorough study of the effects of adding M3GP-evolved hyper-features to the reference datasets. We test our approach on several datasets from different images in two types of problems that have been tackled several times over the last decades, the binary classification of

[1] https://www.un-redd.org/
[2] https://unfccc.int/topics/land-use/resources/warsaw-framework-for-redd-plus
[3] also called attributes, or features, by the ML community

burnt areas [16–22] and the multiclass classification of land cover types [23–27]. The images used in our study cover several different regions over developing countries: Angola, Brazil, Democratic Republic of the Congo, Guinea-Bissau and Mozambique. We add the evolved hyper-features to the reference datasets and analyse the differences in the generalisation ability of different ML algorithms when tested on unseen data from the same images. Three common state-of-the-art algorithms are tested, namely Decision Trees [28], Random Forests [29] and XGBoost [30]. We also perform the same experiments when adding spectral indices instead of the hyper-features, comparing the results. The selected indices are the popular NDVI, NDWI and NBR. For the binary classification problems, we also compare our results with the ones obtained when adding hyper-features created by two different Feature Construction methods, EFS [31] and FFX [32].

It is important to emphasise the differences between the current work and the previous one [11]. On the previous work, a manual selection of evolved hyper-features completely replaced the original ones, while here all the hyper-features resulting from each run are automatically added to the reference datasets. The goal of the previous work was to explore feature spaces in order to explain the variable degrees of success of Transfer Learning to different images. Here, we concentrate on the performance inside each image, and compare our approach to alternative ones that use indices and other types of hyper-features. Finally, while the previous work only used binary classification datasets, this one greatly extends its reach by tackling also multiclass classification problems.

2. Related Work

Feature Engineering is an essential step in the knowledge discovery process and one of the keys to success in applied ML. The features used to induce a data model can directly influence the quality of the model itself and the results that it can achieve. Feature Engineering can be broadly partitioned into Feature Selection and Feature Construction. According to [33], Feature Selection is a process that chooses a subset of features from the original data variables, so that the feature space is optimally reduced according to a certain criterion, while Feature Construction/Extraction (also called Feature Generation, Feature Learning, or Constructive Induction) is a process that creates a new set of hyper-features from the original data. Feature Construction typically combines existing variables into more informative hyper-features. Both Feature Selection and Feature Construction attempt to improve model performance, and can be used in isolation or in combination.

Feature Construction, the focus of this work, has been widely studied in the last two decades. Recent surveys can be found in [34–36], while the book [37] gives an in-depth presentation of the area. In all these references, the importance of Evolutionary Computation (EC) as an effective method for Feature Construction is asserted, together with other Feature Construction methods such as the ones based on Decision Trees, Inductive Logic Programming and Clustering. A very recent survey of EC techniques for Feature Construction can be found in [38]. Among the different EC flavours, GP is probably the one that has been used more often and more successfully. Indeed, GP is particularly suited for Feature Construction because it naturally evolves functions of the original variables. The versatility offered by the user-defined fitness function of GP allows the user to choose among several possible criteria for evolving new hyper-features. Additionally, the fact that the evolved hyper-features are, in principle, readable and understandable, can play an important role in model interpretability. Several existing GP-based methods for Feature Construction are discussed in [39], and a deep analysis of previous work can be found in [15], where GP-constructed features are used for Transfer Learning.

Among the large set of Feature Construction methods available, in this paper we use M3GP [13] as our method of choice, and two others for comparison purposes: the non-EC method FFX [32] and the EC method EFS [31]. The remainder of this section will focus on GP-based Feature Construction, including applications, and on Feature Construction and GP in the context of RS.

2.1. Feature Construction with Genetic Programming

Among the several previous contributions in which GP was used for Feature Construction, Krawiec has shown that classifiers induced using the representation enriched by GP-constructed hyper-features provide better accuracy on a set of benchmark classification problems [40]. Krawiec and colleagues have also used GP in a co-evolutionary system for Feature Construction [41,42].

The use of GP for Feature Construction was later deeply investigated by Zhang and colleagues. For instance, in [43], a GP approach was proposed that, instead of wrapping a particular classifier for single Feature Construction as in most of the existing methods, it used GP to construct multiple features from the original variables. The proposed method used a fitness function based on class dispersion and entropy, and thus was independent of any particular classification algorithm. The approach was tested using Decision Trees on the new obtained dataset and experimentally compared with the standard Decision Tree method, using the original features. The results showed that the proposed approach outperforms standard Decision Trees on the studied test problems in terms of the classification performance, dimension reduction and the learned Decision Tree size. Several years later, in [44], tree-based GP was used for both Feature Construction and implicit Feature Selection. The work presented a comprehensive study, investigating the use of GP for Feature Construction and Feature Selection on high-dimensional classification problems. Different combinations of the constructed and/or selected features were tested and compared on seven high-dimensional gene expression problems, and different classification algorithms were used to evaluate their performance. The results indicated that the constructed and/or selected feature sets can significantly reduce the dimensionality and maintain or even increase the classification accuracy in most cases. In [45], previous GP-based approaches for Feature Construction were extended to deal with incomplete data. The results indicated that the proposed approach can, at the same time, improve the accuracy and reduce the complexity of the learnt classifiers. While until a few years ago GP-based Feature Construction had been applied mainly to classification, in [46] it was applied with success to symbolic regression, thus giving a demonstration of the generality of the approach. In [47], different approaches based on GP to constructing multiple features were investigated. One of the most interesting reported results showed that multiple-feature construction achieves significantly better performance than single-feature construction. Consistently with that result, also the method presented in this paper uses GP to construct multiple features.

GP-based Feature Construction methods have been used with success in several real-life applications. For instance, [48] proposed a novel method for breast cancer diagnosis using the features generated by GP. A few years later, in [49], GP-based Feature Construction was used for improving the accuracy of several classification algorithms for biomarker identification. In [50], a method to find smaller solutions of equally high quality compared to other state-of-the-art GP approaches was coupled with a GP-based Feature Construction method and applied to cancer radiotherapy dose reconstruction. One year later, in [51], GP-based Feature Construction was successfully applied to the classification of ten different categories of skin cancer from lesion images. Interestingly, while the application tackled in [50] is a symbolic regression problem, the one in [51] is a multiclass classification problem, thus confirming that the GP-based Feature Construction approach can be successfully applied to both types of problems. Finally, in [52], GP-based Feature Construction was extended for the first time to experimental physics. In particular, to be applicable to physics, dimensional consistency was enforced using grammars. The presented results showed that the constructed hyper-features can both significantly improve classification accuracy and be easily interpretable.

2.2. Feature Construction and Genetic Programming in Remote Sensing

In the RS domain, many techniques have been used to extract features from satellite images. These features include statistical descriptors, obtained by the Gray Level Co-occurrence Matrix (GLCM) and other methods [53]; features of interest, such as known structures (e.g., buildings, roads), using deep learning [54]; sets of generic features, using the Principal Component Analysis (PCA) [55]; and even temporal features, using the Continuous Change Detection and Classification (CCDC) algorithm [56].

GP-based algorithms, mainly the standard GP algorithm, have been previously used in the area of RS in tasks such as the creation of vegetation indices [57], the detection of riparian zones [58] and the estimation of soil moisture [58,59], the estimation of canopy nitrogen content at the beginning of the tasselling stage [60], the estimation of chlorophyll levels [61], the prediction of soil salinity by estimating the electrical conductivity on the ground [62], the monitoring of water quality in reservoirs [63] and also in geoscience projects reviewed in [64].

The expressions obtained by the GP-based algorithms can be used in Transfer Learning by exporting them to datasets under the form of hyper-features, in the attempt to improve the performance of ML algorithms. Our work continues to develop this kind of application, which was already explored in the area of RS using EC-based algorithms [65,66] and specifically GP-based algorithms [11,58].

3. Materials and Methods

In this section, we begin by describing the datasets used and their geographic locations. Then we explain our methodology, and briefly describe the Feature Construction and classification algorithms used, as well as their implementations and parameters.

3.1. Datasets and Study Areas

The datasets used in this work are meant to train ML models to classify burnt areas and land cover types on a pixel-level. We use a total of nine datasets, obtained from Landsat-7, Landsat-8 and Sentinel-2A satellite images. The characteristics of these datasets are summarised in Table 1 and their associated geographic locations are highlighted in Figure 1.

From the Landsat-7 images, we have one binary classification dataset (Gw2) and two multiclass classification datasets (IM-10 and IM-3). The IM-3 dataset was built, in previous work, from IM-10 by extracting three forest land cover types that ML models failed to correctly discriminate. These images were both obtained over Guinea-Bissau.

From the Landsat-8 images, we have three binary classification datasets and two multiclass classification datasets. The binary classification datasets have the objective of training models to identify burnt areas, by classifying each pixel as "burnt" or "non-burnt". These three datasets were obtained from satellite images over Angola (Ao2), Brazil (Br2) and Democratic Republic of the Congo (Cd2). The multiclass classification datasets have the objective of training models to correctly classify each pixel as one of several different land cover types. These two datasets were extracted from satellite images over Angola (Ao8) and Guinea-Bissau (Gw10).

Lastly, from the Sentinel-2A satellite images, we have one multiclass classification dataset that was extracted from several satellite images from the entire country of Mozambique (Mz6). These images were obtained through approximately the 97^{th} and the 280^{th} day of 2016 [67].

3.2. Methodology

The core of this work is to expand the reference datasets with hyper-features that improve the performance of different ML methods. Figure 2 illustrates the process of obtaining and using such hyper-features. As usual, the reference dataset is split in two datasets, one for training the classifiers, called the training set, and one for testing the classifiers on unseen data, called the test set. Based only on the training set, the Feature Construction algorithm creates a set of hyper-features that are used to expand the reference dataset, in both training and test sets. The expanded training set is used by the classification algorithm to obtain a trained classifier, that is applied to both (expanded) training and test sets in order to report the performance in terms of learning and generalisation, respectively.

A small deviation to this process has been made for the datasets IM-3 and IM-10, where the hyper-features used to expand the IM-10 datasets where obtained in the training data of its subset IM-3, and not the training data of the complete IM-10. The goal of this deviation was to check whether

Table 1. Summary of the datasets used.

Dataset	Ref.	Country	Scene Identifier	Acq. Date	Satellite	Classes	No. Bands	No. Pixels
		ISO Code	Path / Row	DD/MM/YYYY			No. Features	
Ao2	[a]	AO	177 / 67	09/07/2013	LS-8	2	7	3882
Br2	[68]	BR	225 / 64	28/02/2015	LS-8	2	7	4872
Cd2	[68]	CD	175 / 62	08/06/2013	LS-8	2	7	2849
Gw2	[68]	GW	204 / 52	13/05/2002	LS-7	2	7	4531
IM-3	[b]	GW	203 / 51 52 204 / 51 52 205 / 51	From: 02/01/2010 To: 01/04/2010	LS-7	3	6	322
IM-10	[69]					10		6798
Ao8	[70]	AO	182 / 64 65	18/06/2016	LS-8	8	10	2183
Gw10	[c]	GW	204 / 51 52 205 / 51	01/03/2019 24/03/2019	LS-8	10	7	5080
Mz6	[67]	MZ	Entire Country	From: 06/04/2016 To: 06/10/2016	S-2A	6	10	190202

[a] This image is yet to be studied in a paper.
[b] This is a sub-dataset, obtained by extracting three forest classes from the IM-10 dataset.
[c] This paper is under the reviewing process.

a larger dataset could also benefit from hyper-features obtained in a much limited context (results reported in Subsection 4.3).

As Feature Construction algorithms, we use M3GP and compare it with FFX and EFS, all described below. As classification algorithms, we use Decision Trees (DT), Random Forests (RF) and XGBoost (XGB), also briefly described below. The number and complexity of the created hyper-features is not predefined, but automatically determined by the Feature Construction algorithm.

We also experiment with expanding the reference datasets with the NDVI, NDWI and NBR indices, instead of doing Feature Construction. These indices were selected from the RS literature as being helpful to the ML algorithms for separating vegetation, water and burnt classes, since these elements are present among the pixels used in the datasets.

Each experiment is performed 30 times, each time with a different random split of the reference dataset in training (70% of the pixels) and test sets (remaining 30%), stratified by class.

3.3. Feature Construction Algorithms

We use three different methods for Feature Construction. Our method of choice is the M3GP algorithm, because of the interpretability of the hyper-features it creates, and because it can evolve hyper-features for multiclass classification problems. For comparing our M3GP results with the results of other evolutionary and non-evolutionary methods, we selected the EFS and FFX algorithms, due to their running speed, availability of the authors' implementations and number of citations. However, EFS and FFX are focused on regression problems, rather than classification problems. They are easily adapted to binary classification, by defining a threshold separating the two classes, but there is no easy adaptation for multiclass problems, the reason why we test them only on the binary classification datasets.

M3GP algorithm: Multidimensional Multiclass GP with Multidimensional Populations (M3GP) is a GP-based algorithm that evolves a set of hyper-features that convert the original feature space into a new feature space, guided by a fitness function that measures the performance of a classifier in the new feature space. It is, in fact, an all-in-one algorithm that both creates the hyper-features and uses them for solving the problem. In the original implementation, the fitness function is the accuracy of the Mahalanobis Distance (MD) classifier (described below). In our implementation, we use the Weighted Average of F-measures (WAF) instead of the accuracy[4], for its robustness to class imbalance,

[4] although we still use the accuracy to assess performance

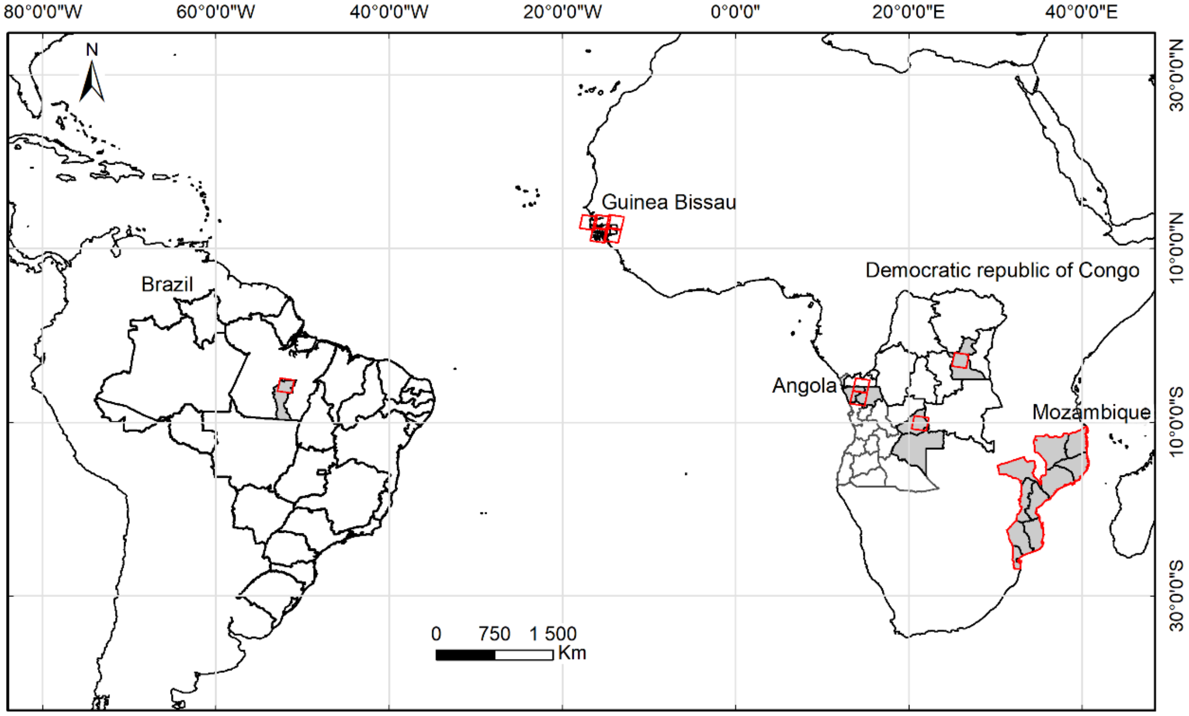


Figure 1. Location of the study areas, in red, in Brazil (Br2), Guinea-Bissau (Gw2, Gw10, IM-3, IM-10), Democratic Republic of Congo (Cd2), Angola (Ao2, Ao8) and Mozambique (Mz6) in South America and Africa continent.

especially in multiclass classification. Using the WAF as fitness for training M3GP models is only one of several possible options in this versatile algorithm. As an example, in order to solve a regression problem, M3GP can evolve hyper-features and use them in linear-in-parameter (or any other type of) regression, using an error measure as fitness. Figure 3 displays an example of a set of hyper-features that the M3GP algorithm could evolve. The double-slash denotes a division that is protected against division by zero, by returning 1 whenever the denominator is null.

EFS algorithm: Evolutionary Feature Synthesis (EFS) is an evolutionary algorithm that uses pathwise LASSO [71] regression to optimise multiple linear regression models that are extended for nonlinear relationships between features. This extension is made using functions such as cos, sin and log, as well as functions with several inputs, e.g., multiplication of variables. This regression tool can produce a set of interpretable hyper-features in seconds.

FFX algorithm: Fast Function Extraction (FFX) is a deterministic algorithm that applies Pathwise Regularised Learning [72] to a large set of generated nonlinear functions to search for a set of hyper-features with minimal error. Although the hyper-features observed are simple, this algorithm generates hundreds of hyper-features, which leads us to consider the final model non-interpretable.

3.4. Classification Algorithms

Four different classifiers are used in this work: MD, DT, RF and XGB. The MD classifier is used only as part of M3GP, but the other three are used to independently test the effectiveness of the indices and hyper-features added to the reference datasets.

Mahalanobis Distance classifier: The MD classifier is a non-parametric supervised cluster-based algorithm that classifies a data point by associating it with the closest cluster centroid using the Mahalanobis distance, where a cluster is defined as a set of pixels belonging to the same class.

Decision Tree classifier: The DT algorithm is a non-parametric supervised algorithm that infers simple decision rules from the training data. This algorithm can be used in both classification and regression problems.

doi:10.20944/preprints202010.0168.v1

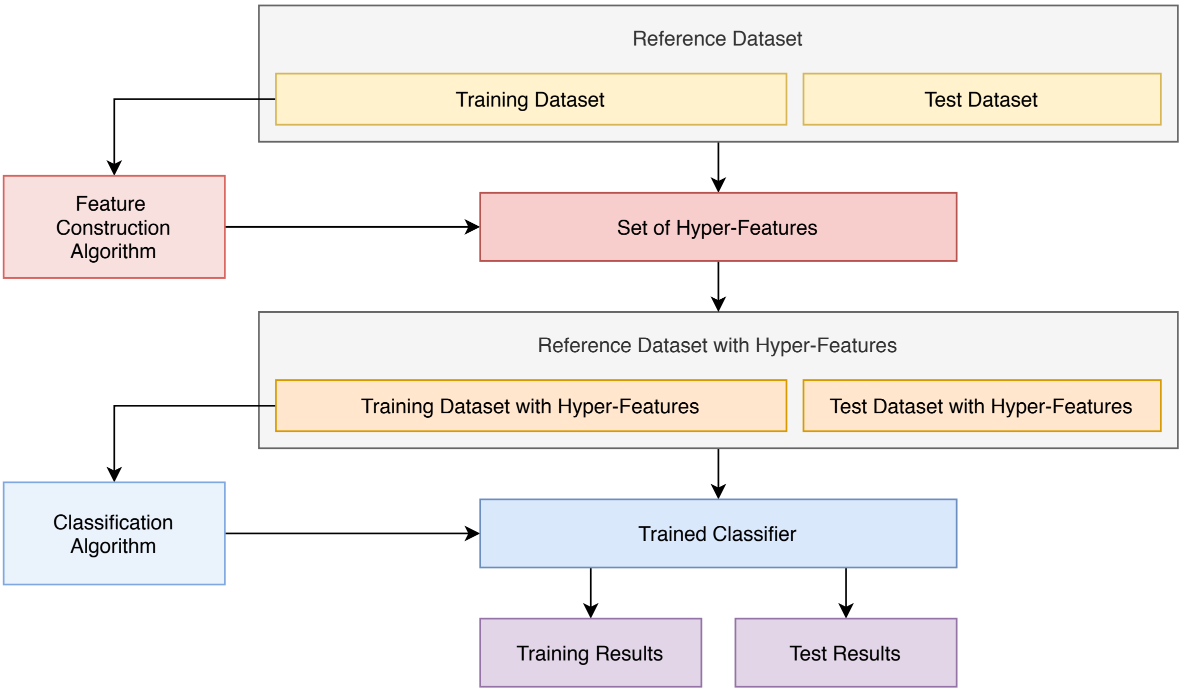


Figure 2. Representation of the methodology adopted to obtain and use hyper-features.

Random Forest classifier: The RF algorithm is an ensemble algorithm that uses a set of DTs to solve both classification and regression problems, by assigning each data point to the majority vote of all DTs in classification problems, or to the average of the prediction of all DTs in regression problems.

XGBoost classifier: The XGB algorithm is DT-based ensemble algorithm that uses an optimised gradient boosting to minimise errors. This algorithm can be used in both classification and regression problems.

3.5. Tools and Parameters

All the experiments involving M3GP are performed using our own implementation of the M3GP algorithm[5], which includes the implementation of the MD classifier. The DT and RF implementations belong to the sklearn python library [73] and the XGB implementation belongs to the xgboost python library [30]. The EFS implementation[6] is provided by the authors in their paper [31] and the FFX implementation[7] belongs to the ffx python library.

The parameter settings used in this work are the standard within the ML community, with the main parameters and variations specified in Table 2. The EFS, FFX and M3GP algorithms used the same parameters as those used by the authors in their respective papers. The variations in this work include our implementation of the M3GP using the WAF of the MD classifier (untied with the number of hyper-features and then with the total size of the model) as fitness, rather than the accuracy, and only pruning the final individual, for consistency with previous work that had this variation [11]. Every run using the DT, RF and XGB classifiers used the default parameters of their respective implementations, except for the XGB runs in the Mz6 dataset. In this dataset, the XGB was unable to obtain perfect training accuracy with the default maximum depth for its models. As such, the maximum depth was increased from 6 to 20.

[5] Python implementation available at https://github.com/jespb/Python-M3GP
[6] EFS project website: http://flexgp.github.io/efs/
[7] Python implementation available at https://github.com/natekupp/ffx

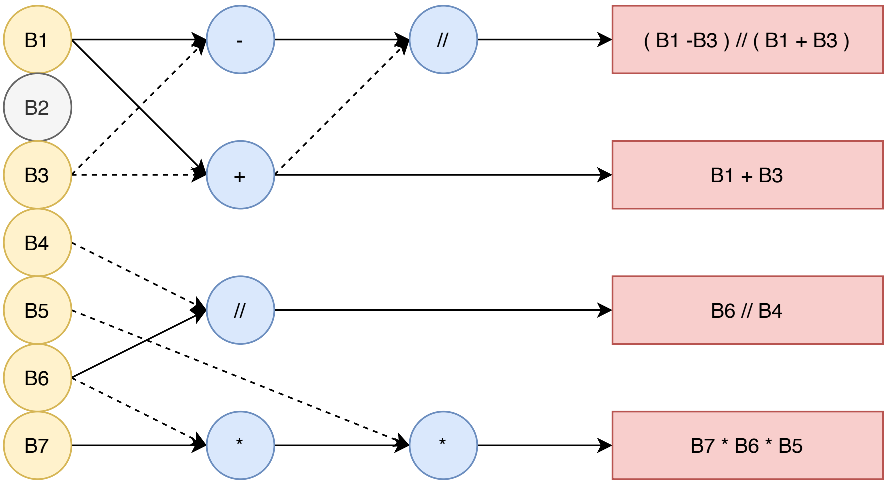


Figure 3. Example of an M3GP model that uses six of the seven available features to build four hyper-features. The solid and dashed lines indicate the first and second variables used by the operators. // is a division operator protected against division by zero.

Table 2. Main parameters and variations used in the experiments.

General:	
Runs	30
Training Set	70% of the samples of each class
Statistical Significance	p-value < 0.01 (Kruskal-Wallis H-test)
M3GP:	
Stopping Criteria	50 generations or 100% training accuracy
Fitness	WAF (Weighted Average of F-measures)
Pruning	Final individual
XGBoost:	
Maximum Depth	20 in the Mz6 dataset and 6 (default) in the other datasets

4. Results and Discussion

This section contains the results of our experiments and a small discussion in each of the four subsections. First, we present the results of running M3GP by itself on all the datasets, and discuss the interpretability of the hyper-features it evolved, also presenting examples of some of them. Then, we present several results of running DT, RF and XGB on the datasets expanded with the evolved hyper-features: on the four binary classification datasets (Ao2, Br2, Cd2, Gw2); on the two related multiclass classification datasets (IM-3 and IM-10), as a special case of evolving hyper-features to help discriminate only the most difficult classes; and on the remaining multiclass classification datasets (Ao8, Gw10, Mz6).

We present the results in tables and boxplots. On the tables, each accuracy value is the median obtained in the 30 runs. Statistical significance is determined with the non-parametric Kruskal-Wallis H-test (from the scipy Python library) at $p < 0.01$. The p-values of the pairwise comparisons are included in the tables, and coloured green/red when the obtained test accuracy is significantly better/worse than the test accuracy obtained by the other technique in the comparison. When describing and discussing these results, the terms "better" and "worse" always imply a statistically significant difference. On the boxplots, each box has lines at the lower quartile, median, and upper quartile values; the whiskers mark the furthest value within 1.5 of the quartile ranges, and outliers are represented as circles.

4.1. M3GP Performance and Hyper-feature Interpretability

Although we use M3GP as a Feature Construction method for other ML algorithms, M3GP is capable of performing binary and multiclass classification by itself, as described in Subsection 3.4.

While using M3GP to evolve the hyper-features, we have registered the accuracy values it achieved in each dataset, presented in Table 3. Although the accuracy is high, it is generally worse than the accuracy achieved by the other ML algorithms we used, and therefore we will not refer to the results of standalone M3GP again.

In terms of interpretability of the evolved hyper-features, in Table 3 we report the number of hyper-features and their median size (with minimum and maximum values, between parenthesis). While the number of evolved hyper-features seems to depend heavily on the number of classes of the problem, the median average size of each hyper-feature tends to be higher for the binary datasets, where a very large dispersion of values is observed.

To exemplify the variety of different sets of evolved hyper-features, we picked three examples. On the first two examples, a single hyper-feature was evolved, but with very different sizes. Both were evolved for the Gw2 dataset, and obtained perfect test accuracy on the respective runs. The third example is a set of 16 hyper-features that were evolved in a run for the Ao8 dataset, and obtained median test accuracy. This variety of hyper-features can be seen in Eqs. 1 through 7. Note that Bn refers to the n^{th} band of the satellite. As we can see, the M3GP algorithm can generate hyper-features that are as simple as (and perfectly equal to) the original features themselves (Eqs. 3), hyper-features that are simple enough to be interpreted (Eqs. 1, 4 and 5), and hyper-features which need to be decomposed for a proper analysis of the expression (Eqs. 2, 6 and 7).

Looking at Table 3 and the examples of hyper-features in 1 through 7, we can state that, although the M3GP sometimes produces complex hyper-features, the general case seems to be the production of interpretable hyper-features. While this work focuses exclusively on datasets from the RS domain, the same tendency regarding interpretability was already observed in the original M3GP paper [13], which used datasets from a much wider range of domains.

Table 3. The median training and test accuracy, size, number of hyper-features and average size of the hyper-features obtained by the M3GP models in 30 runs in each dataset.

	Ao2	Br2	Cg2	Gw2	IM-3	IM-10	Ao8	Gw10	Mz6
Accuracy									
Training	**1.000**	**0.992**	**0.993**	**1.000**	**0.996**	**0.932**	**1.000**	**0.988**	**0.620**
Test	**0.999**	**0.990**	**0.993**	**1.000**	**0.948**	**0.916**	**0.983**	**0.971**	**0.620**
Hyper-Features									
Number	5(3-8)	8(1-15)	4(3-8)	2(1-3)	8.5(5-13)	23(17-29)	18(14-21)	21(15-23)	14.5(12-17)
Avg. Size	14(9-28)	14(6-39)	23(6-40)	11(2-43)	11(5-22)	10(8-13)	10(5-14)	8(6-12)	11(5-17)

Gw2, Run#11, 1 Hyper-feature:

$$\frac{B5\,(B3+B5)}{B7+B4+1} \tag{1}$$

Gw2, Run#26, 1 Hyper-feature:

$$\frac{B2^2\,B3\,B4\,B5\,B6\;-\;B2^2\,B4^2\,B5\;+\;B2\,B3^2\,B5^2\,B6\;-\;B2\,B3\,B4\,B5^2\;-\;B3\,B4^3\,B7}{B4^2\,B7\,(B2\,B4\;+\;B3\,B5)} \tag{2}$$

Ao8, Run#18, 16 Hyper-features :

$$B3 \qquad B5 \qquad B6 \qquad B10 \qquad B11 \tag{3}$$

$$B9-B2 \qquad \frac{B3}{B5} \qquad \frac{B1}{B2\,B7^2} \qquad B5-B6+B9-2\,B10 \qquad \frac{B2\,B9}{B2\,B11-B10-B2\,B5} \tag{4}$$

$$\frac{(B1+B4-B10)\,(B3+B9)}{B6} \qquad B6\,B9-B1\,B2-B1+B3+B6-B9 \tag{5}$$

$$\frac{B9\,(B9-B11)}{B7\,(B3\,B6+B9-B11)} \qquad (B4+B10-\frac{B1^2}{B5-B9})\,(2\,B2+\frac{B3}{B4}-B4+B5+B10) \tag{6}$$

$$\frac{B7\,B9^2\,(B2+B7-B11)}{B5\,B6\,B11^2\,(B2+B3-B9)} \qquad \frac{B1\,B2\;+\;B1\,B3\,B6\;+\;B1\,B3\,B9\;+\;B3\,B4\,B5}{B11} \tag{7}$$

4.2. *Hyper-features in Binary Classification Datasets*

The results obtained on the binary classification datasets (Ao2, Br2, Cd2, Gw2) are reported in Table 4 and Figure 4. In terms of training accuracy, the three classification algorithms managed to obtain perfect results in nearly every run, and therefore those results are not included in the table. In terms of test accuracy, the induced models achieved very high values, nearly all above 99%, also on the original (non-expanded) datasets. The lowest results belong to DTs that, when applied to the Br2 dataset, achieved a median test accuracy of 98.9%. Without much room for improvement, still FFX was able to create hyper-features that improved the test accuracy in two cases (DT and XGB in the Gw2 dataset), surpassing also the M3GP hyper-features, while neither the indices nor the M3GP or EFS hyper-features caused any significant difference in the results. The boxplots show a very low dispersion of accuracy values (the ranges of the y-axes are very limited), which seems to be marginally larger for the EFS results.

These seemingly uninteresting results agree with the findings of our previous work [11]. There, using a different method of selecting and using the hyper-features also had no effect in the cases where the training and test datasets came from the same image. However, the hyper-features revealed to be beneficial when the induced models where applied to datasets that came from images not seen during training. This suggests that the current method of obtaining and using the hyper-features may also prove beneficial in a similar training and test setting.

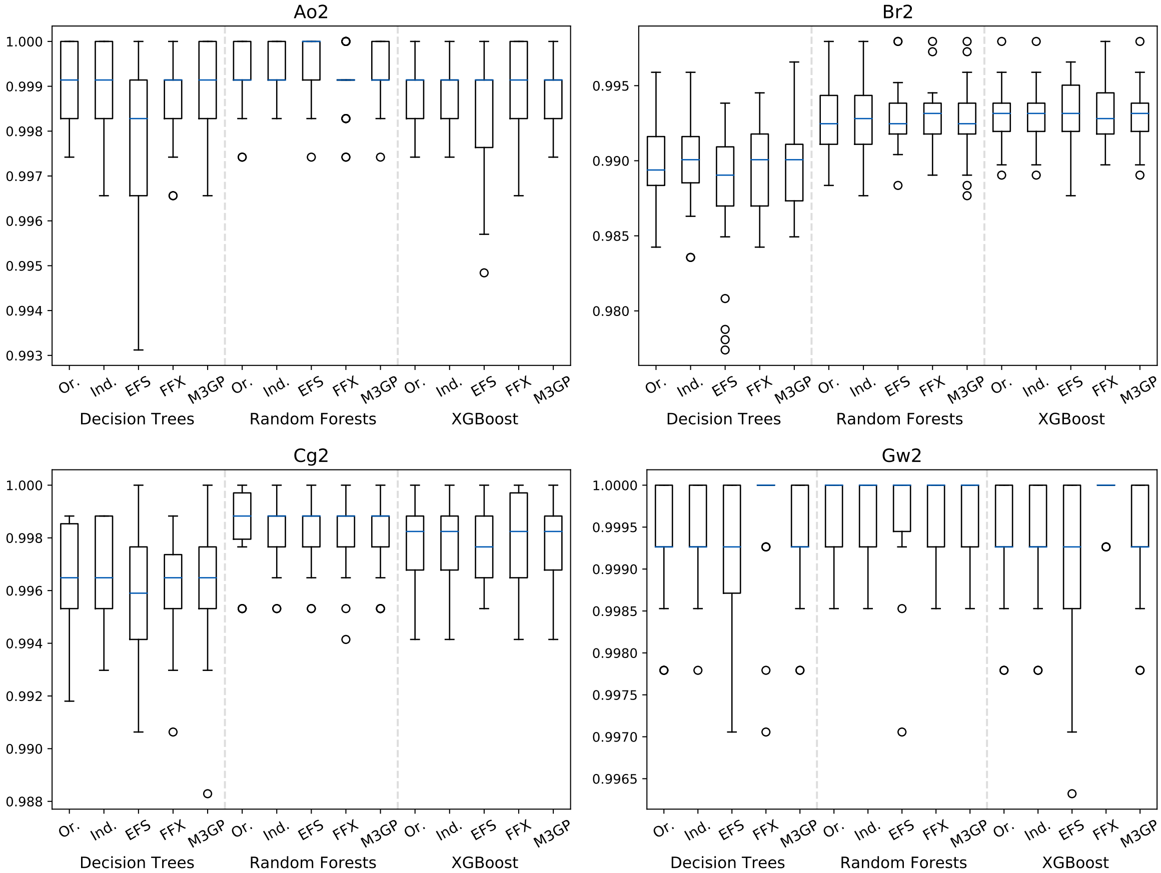
Table 4. Comparison of the median test accuracy obtained by the three ML algorithms in the original datasets, when adding indices, and when using hyper-features evolved by EFS, FFX and M3GP. The coloured p-values indicate significantly better/worse results.

	Decision Trees				Random Forests				XGBoost			
Dataset	Ao2	Br2	Cd2	Gw2	Ao2	Br2	Cd2	Gw2	Ao2	Br2	Cd2	Gw2
Orig. Dataset												
Test Accuracy	**0.999**	**0.989**	**0.996**	**0.999**	**0.999**	**0.992**	**0.999**	**1.000**	**0.999**	**0.993**	**0.998**	**0.999**
Indices												
Test Accuracy	**0.999**	**0.990**	**0.996**	**0.999**	**0.999**	**0.993**	**0.999**	**1.000**	**0.999**	**0.993**	**0.998**	**0.999**
p-value vs Orig.	0.694	0.678	0.909	0.669	0.675	0.917	0.436	0.871	1.000	1.000	1.000	1.000
EFS												
Test Accuracy	**0.998**	**0.989**	**0.996**	**0.999**	**1.000**	**0.992**	**0.999**	**1.000**	**0.999**	**0.993**	**0.998**	**0.999**
p-value vs Orig.	0.012	0.226	0.143	0.735	0.091	0.777	0.137	0.619	0.256	0.682	0.489	0.127
FFX												
Test Accuracy	**0.999**	**0.990**	**0.996**	**1.000**	**0.999**	**0.993**	**0.999**	**1.000**	**0.999**	**0.993**	**0.998**	**1.000**
p-value vs Orig.	0.224	0.941	0.294	0.000	0.363	0.988	0.148	0.730	0.739	0.794	0.886	0.000
M3GP												
Test Accuracy	**0.999**	**0.990**	**0.996**	**0.999**	**0.999**	**0.992**	**0.999**	**1.000**	**0.999**	**0.993**	**0.998**	**0.999**
p-value vs Orig.	0.908	0.947	0.672	0.813	0.500	0.846	0.688	0.871	1.000	1.000	1.000	1.000
p-value vs Ind.	0.782	0.761	0.598	0.849	0.780	0.982	0.738	1.000	1.000	1.000	1.000	1.000
p-value vs FFX	0.276	0.830	0.525	0.001	0.095	0.905	0.319	0.868	0.739	0.794	0.886	0.000
p-value vs EFS	0.017	0.272	0.291	0.572	0.286	0.682	0.309	0.757	0.256	0.682	0.489	0.127

4.3. *Hyper-features to Discriminate Similar Classes in a Multiclass Classification Dataset*

Before looking at these results, it is worth recalling that the IM-3 dataset was built from three similar classes within the IM-10 dataset. As such, even though it has a reduced number of classes, it is not unexpected to see a lower accuracy in this dataset. It is also worth specifying that the hyper-features

Figure 4. Boxplots of the test accuracy obtained in the binary classification datasets in each test case.



used in the IM-10 dataset were obtained only in the IM-3 dataset, in an attempt to help discriminate these similar classes. Finally, we also recall that EFS and FFX are not used in the multiclass datasets.

The results for IM-10 and IM-3 are reported in Table 5 and Figure 5. Once again, the training results are omitted from the table because all three algorithms achieved perfect results in nearly every run. In terms of test accuracy, we observe that, although the values are high, they have a larger margin for improvement when compared to the binary classification results reported above. When adding indices to the original dataset, the test accuracy on the IM-10 dataset increased with two algorithms (RF and XGB). When adding the hyper-features evolved by M3GP, the test accuracy in the IM-10 dataset increased with all three algorithms, and in the IM-3 dataset it increased with the XGB algorithm. Neither the indices nor the M3GP hyper-features degraded the test accuracy. When comparing the performance of indices versus M3GP hyper-features, M3GP is better with two algorithms (DT and XGB). In the boxplots, we observe that IM-3 has a larger dispersion of values than IM-10 (notice the different y-axes ranges). On IM-10, the DT algorithm visibly falls behind RF and XGB.

Although the M3GP hyper-features performed better than the indices, the indices were also clearly beneficial when added to the original datasets. It is worth noticing that the IM-3 and IM-10 datasets were extracted from a set of satellite images with different acquisition dates. Next, we will observe additional evidence that indices and hyper-features seem to be more useful in datasets coming from sets of images with different acquisition dates.

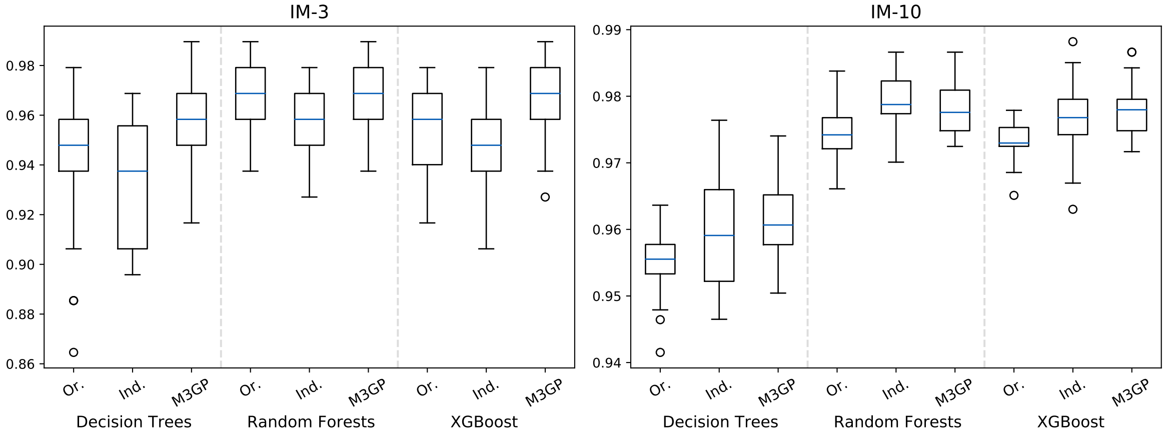
4.4. *Hyper-features to Discriminate All Classes in Multiclass Classification Datasets*

The results obtained on the three unrelated multiclass classification datasets (Ao8, Gw10, Mz6) are reported in Table 6 and Figure 6. Once again, the training results were omitted from the table, as

Table 5. Comparison of the median test accuracy obtained by the three ML algorithms in the original datasets, when adding indices, and when adding hyper-features evolved by the M3GP algorithm. The coloured *p*-values indicate significantly better results.

	Decision Trees		Random Forests		XGBoost	
Dataset	IM-3	IM-10	IM-3	IM-10	IM-3	IM-10
Original Dataset						
Test Accuracy	**0.948**	**0.956**	**0.969**	**0.974**	**0.958**	**0.973**
Indices						
Test Accuracy	**0.938**	**0.959**	**0.958**	**0.979**	**0.948**	**0.977**
p-value vs Original	0.151	0.051	0.062	0.000	0.178	0.001
M3GP						
Test Accuracy	**0.958**	**0.961**	**0.969**	**0.978**	**0.969**	**0.978**
p-value vs Original	0.020	0.000	0.844	0.000	0.009	0.000
p-value vs Indices	0.000	0.407	0.055	0.218	0.000	0.711

Figure 5. Boxplots of the test accuracy obtained in the IM-3 and IM-10 datasets in each test case.



perfect accuracy was achieved in almost every run. However, for the Mz6 dataset, XGB required a maximum tree depth larger than the implementation default in order to achieve it (see Subsection 3.5).

In terms of test accuracy, the indices improved the accuracy in two test cases (DT on Gw10, XGB on Mz6) and reduced the accuracy in one test case (RF on Mz6). On the other hand, the hyper-features evolved by M3GP improved the test accuracy in five test cases (Mz6 with all algorithms, and Gw10 with RF and XGB), when comparing the results with those on the original dataset, and in four test cases, when comparing with the results obtained with the indices (Mz6 with all algorithms, Gw10 with XGB). Once again, the hyper-features evolved by M3GP did not lead to a degradation of the test accuracy in any of the cases.

Both the indices and the M3GP-evolved hyper-features had an impact on the Gw10 and Mz6 datasets, which were obtained from a set of satellite images with different acquisition dates. Neither the indices nor the hyper-features had an impact on the Ao8 dataset, which was obtained from two images with the same acquisition date. These results, together with those displayed in the previous subsection, seem to indicate that both the indices and the hyper-features are particularly useful in datasets obtained by mixing satellite images with different acquisition dates.

On the boxplots, once again we observe that DT falls behind RF and XGB, and completely struggles on the Mz6 problem.

5. Conclusions

We performed Feature Construction using M3GP, a variant of the standard Genetic Programming algorithm, with the goal of improving the performance of several Machine Learning algorithms by

Table 6. Comparison of the test accuracy obtained by the three ML algorithms in the original datasets, when adding indices, and when adding hyper-features evolved by the M3GP algorithm. The coloured *p*-values indicate significantly better/worse results.

	Decision Trees			Random Forests			XGBoost		
Dataset	Ao8	Gw10	Mz6	Ao8	Gw10	Mz6	Ao8	Gw10	Mz6
Original Dataset									
Test Accuracy	**0.977**	**0.964**	**0.662**	**0.988**	**0.981**	**0.773**	**0.985**	**0.979**	**0.780**
Indices									
Test Accuracy	**0.978**	**0.968**	**0.662**	**0.989**	**0.980**	**0.769**	**0.986**	**0.980**	**0.781**
p-value vs Original	0.291	0.000	0.371	0.213	0.824	0.000	0.645	0.335	0.003
M3GP									
Test Accuracy	**0.980**	**0.970**	**0.665**	**0.988**	**0.982**	**0.775**	**0.987**	**0.983**	**0.786**
p-value vs Original	0.125	0.000	0.000	0.923	0.054	0.006	0.389	0.000	0.000
p-value vs Indices	0.693	0.847	0.000	0.228	0.038	0.000	0.650	0.002	0.000

adding the new hyper-features to the reference datasets. We tested the approach in the tasks of binary classification of burnt areas and multiclass classification of land cover types. The datasets used were obtained from Landsat-7, Landsat-8 and Sentinel-2A satellite images over the countries of Angola, Brazil, Democratic Republic of Congo, Guinea-Bissau, and Mozambique.

The hyper-features produced by the M3GP algorithm, although variable in number and size, were generally not very complex, and considered to be quite interpretable. While a larger number of hyper-features were created on the multiclass classification problems, a higher dispersion of sizes was observed on the binary problems.

The performance of Decision Trees, Random Forests and XGBoost was assessed on the original datasets and on the datasets expanded with the evolved hyper-features, and the results compared for statistical significance. For comparison purposes, we also assessed the performance of the same algorithms on all datasets expanded with the well-known spectral indices NDVI, NDWI and NBR, and on the binary datasets expanded with hyper-features created by the FFX and EFS Feature Construction algorithms.

On the binary classification problems, we conclude that neither of the four alternatives (M3GP, indices, FFX, EFS) leads to substantial improvements. Only FFX was able to improve the results in 2 out of 12 test cases (both on the same dataset). On the multiclass classification problems, the hyper-features evolved by the M3GP caused significant improvements in 9 out of 15 test cases, with no degradation of results in any test case, while the indices caused significant improvements in 4 out of 15 test cases and significant degradation of results in one test case. The approach appears to be equally beneficial to all three Machine Learning algorithms.

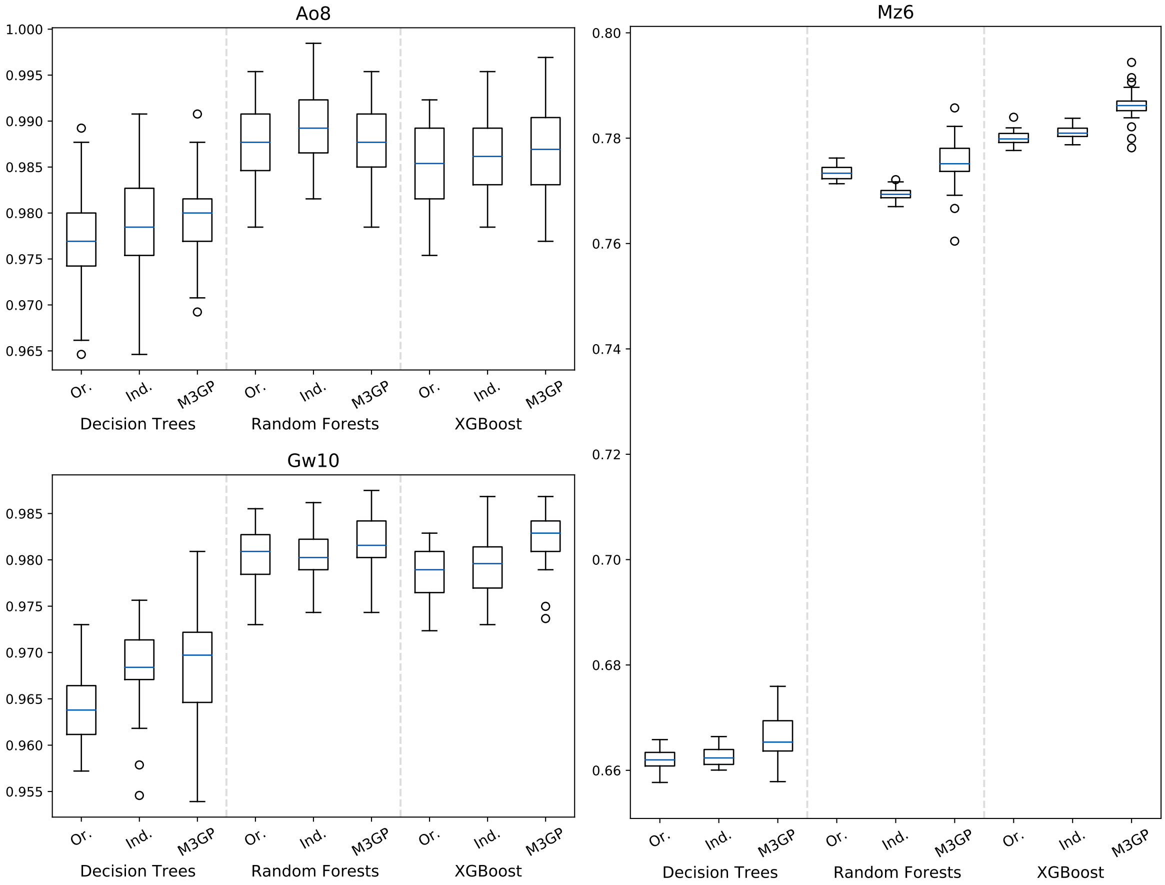
Overall, the indices and hyper-features were only useful in multiclass classification problems whose datasets were built from multiple satellite images with different acquisition dates, regardless of the Machine Learning algorithm used. This suggests that expanding the datasets with these additional variables improves the robustness of different Machine Learning models to the radiometric variations that naturally occur between images acquired at different times.

As future work, we want to continue exploring Feature Construction algorithms whose hyper-features can mitigate the effects of the radiometric variations across satellite images. This would expand this work by exploring Transfer Learning scenarios in either multiclass classification or regression problems, such as the prediction of biomass in satellite images.

Author Contributions: Conceptualization, J.B. and S.S.; methodology, J.B.; software, J.B.; validation, S.S.; formal analysis, J.B.; investigation, J.B.; resources, S.S.; data curation, A.C.; writing–original draft preparation, J.B and L.V.; writing–review and editing, A.C., M.V and S.S.; visualization, A.C. and J.B.; supervision, S.S.; project administration, S.S.; funding acquisition, L.V, M.V and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by FCT through funding of LASIGE Research Unit (UIDB/00408/2020 and UIDP/00408/2020) and CEF (UIDB/00239/2020); projects BINDER

Figure 6. Boxplots of the test accuracy obtained in the Ao8, Gw10, and Mz6 datasets in each test case.



(PTDC/CCI-INF/29168/2017), OPTOX (PTDC/CTA-AMB/30056/2017), PREDICT (PTDC/CCI-CIF/29877/2017), INTERPHENO (PTDC/ASP-PLA/28726/2017), GADgET (DSAIPA/DS/0022/2018), AICE (DSAIPA/DS/0113/2019); PhD Grant (SFRH/BD/143972/2019).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

Ao	Angola
B*x*	Band *x*
Br	Brazil
CCDC	Continuous Change Detection and Classification
Cd	Democratic Republic of the Congo
DT	Decision Tree
EC	Evolutionary Computation
EFS	Evolutionary Feature Synthesis (algorithm)
FFX	Fast Function Extraction (algorithm)
GLCM	Gray Level Co-occurrence Matrix
Gw	Guinea-Bissau
GP	Genetic Programming
LS-7	Landsat 7
LS-8	Landsat 8
M3GP	Multidimensional Multiclass GP with Multidimensional Populations (algorithm or classifier)
MD	Mahalanobis Distance (classifier)
ML	Machine Learning
MRV	Measure, Report and Verify
Mz	Mozambique
NBR	Normalized Burn Ratio
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
PCA	Principal Component Analysis
REDD+	Reducing Emissions from Deforestation and forest Degradation
RF	Random Forest
RS	Remote Sensing
S-2A	Sentinel-2A
UNFCCC	United Nations Framework Convention on Climate Change
XGB	XGBoost
WAF	Weighted Average of F-measures

References

1. Herring, J.A. Measuring Vegetation (NDVI EVI) : Feature Articles. 2000.
2. McFEETERS, S.K. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *International Journal of Remote Sensing* **1996**, *17*, 1425–1432. doi:10.1080/01431169608948714.
3. Key, C.; Benson, N., Landscape Assessment: Ground measure of severity, the Composite Burn Index; and Remote sensing of severity, the Normalized Burn Ratio.; 2006; pp. LA 1–51.
4. Jinru, X.; Su, B. Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications. *Journal of Sensors* **2017**, *2017*, 1–17.
5. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional Neural Networks for Large-Scale Remote Sensing Image Classification. *IEEE T. Geosci. Remote Sens.* **2016**, *55*. doi:10.1109/TGRS.2016.2612821.
6. Ribeiro, F.; Roberts, D.; Hess, L.; Davis, F.; Caylor, K.; Daldegan, G. Geographic Object-Based Image Analysis Framework for Mapping Vegetation Physiognomic Types at Fine Scales in Neotropical Savannas. *Remote Sensing* **2020**, *12*, 1721. doi:10.3390/rs12111721.
7. Dragozi, E.; Gitas, I.; Stavrakoudis, D.; Theocharis, J. Burned area mapping using support vector machines and the FuzCoC feature selection method on VHR IKONOS imagery. *Remote Sensing MDPI* **2014**, *Remote Sens. 2014*, 12005–12036. doi:10.3390/rs61212005.
8. Solano Correa, Y.; Bovolo, F.; Bruzzone, L. A Semi-Supervised Crop-Type Classification Based on Sentinel-2 NDVI Satellite Image Time Series And Phenological Parameters. 2019. doi:10.1109/IGARSS.2019.8897922.
9. Orynbaikyzy, A.; Gessner, U.; Mack, B.; Conrad, C. Crop Type Classification Using Fusion of Sentinel-1 and Sentinel-2 Data: Assessing the Impact of Feature Selection, Optical Data Availability, and Parcel Sizes on the Accuracies. *Remote Sensing* **2020**, *12*, 2779. doi:10.3390/rs12172779.

10. Carrao, H.; Gonçalves, P.; Caetano, M. Contribution of multispectral and multitemporal information from MODIS images to land cover classification. *Remote Sensing of Environment* **2008**, *112*, 986–997. doi:10.1016/j.rse.2007.07.002.
11. Batista, J.E.; Silva, S. Improving the Detection of Burnt Areas in Remote Sensing using Hyper-features Evolved by M3GP. 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2020. doi:10.1109/cec48606.2020.9185630.
12. Poli, R.; B. Langdon, W.; Mcphee, N. *A Field Guide to Genetic Programming*; 2008.
13. Muñoz, L.; Silva, S.; Trujillo, L. M3GP – Multiclass Classification with GP. 2015, Vol. 9025, pp. 78–91.
14. Muñoz, L.; Trujillo, L.; Silva, S.; Castelli, M.; Vanneschi, L. Evolving multidimensional transformations for symbolic regression with M3GP. *Memetic Comput.* **2019**, *11*, 111–126. doi:10.1007/s12293-018-0274-5.
15. Muñoz, L.; Trujillo, L.; Silva, S. Transfer learning in constructive induction with Genetic Programming. *Genetic Programming and Evolvable Machines* **2019**, pp. 1–41.
16. Bastarrika, A.; Chuvieco, E.; Martín, M.P. Mapping burned areas from Landsat TM/ETM+ data with a two-phase algorithm: Balancing omission and commission errors. *Remote Sensing of Environment* **2011**, *115*, 1003 – 1012. doi:https://doi.org/10.1016/j.rse.2010.12.005.
17. Chen, W.; Moriya, K.; Sakai, T.; Koyama, L.; Cao, C. Mapping a burned forest area from Landsat TM data by multiple methods. *Geomatics, Natural Hazards and Risk* **2016**, *7*, 384–402. doi:10.1080/19475705.2014.925982.
18. Daldegan, G.; de Carvalho, O.; Guimarães, R.; Gomes, R.; Ribeiro, F.; McManus, C. Spatial Patterns of Fire Recurrence Using Remote Sensing and GIS in the Brazilian Savanna: Serra do Tombador Nature Reserve, Brazil. *Remote Sensing* **2014**, *6*, 9873–9894. doi:10.3390/rs6109873.
19. Liu, J.; Heiskanen, J.; Maeda, E.E.; Pellikka, P.K. Burned area detection based on Landsat time series in savannas of southern Burkina Faso. *International Journal of Applied Earth Observation and Geoinformation* **2018**, *64*, 210 – 220. doi:https://doi.org/10.1016/j.jag.2017.09.011.
20. Silva, J.M.N.; Pereira, J.M.C.; Cabral, A.I.; Sá, A.C.L.; Vasconcelos, M.J.P.; Mota, B.; Grégoire, J.M. An estimate of the area burned in southern Africa during the 2000 dry season using SPOT-VEGETATION satellite data. *Journal of Geophysical Research: Atmospheres* **2003**, *108*, n/a–n/a. doi:10.1029/2002jd002320.
21. Stroppiana, D.; Bordogna, G.; Carrara, P.; Boschetti, M.; Boschetti, L.; Brivio, P. A method for extracting burned areas from Landsat TM/ETM images by soft aggregation of multiple Spectral Indices and a region growing algorithm. *ISPRS Journal of Photogrammetry and Remote Sensing* **2012**, *69*, 88–102. doi:10.1016/j.isprsjprs.2012.03.001.
22. Trisakti, B.; Nugroho, U.C.; Zubaidah, A. TECHNIQUE FOR IDENTIFYING BURNED VEGETATION AREA USING LANDSAT 8 DATA. *International Journal of Remote Sensing and Earth Sciences (IJReSES)* **2017**, *13*, 121. doi:10.30536/j.ijreses.2016.v13.a2447.
23. Cabral, A.I.R.; vasconcelos, M.J.P.; Pereira, J.M.C.; Martins, E.; Bartholomé, É. A land cover map of southern hemisphere Africa based on SPOT-4 Vegetation data. *International Journal of Remote Sensing* **2006**, *27*, 1053–1074. doi:10.1080/01431160500307409.
24. Cabral, A.; Vasconcelos, M.; Oom, D.; Sardinha, R. Spatial dynamics and quantification of deforestation in the central-plateau woodlands of Angola (1990–2009). *Applied Geography* **2011**, *31*, 1185 – 1193. doi:https://doi.org/10.1016/j.apgeog.2010.09.003.
25. Ceccarelli, T.; Smiraglia, D.; Bajocco, S.; Rinaldo, S.; Angelis, A.D.; Salvati, L.; Perini, L. Land cover data from Landsat single-date imagery: an approach integrating pixel-based and object-based classifiers. *European Journal of Remote Sensing* **2013**, *46*, 699–717. doi:10.5721/eujrs20134641.
26. Midekisa, A.; Holl, F.; Savory, D.J.; Andrade-Pacheco, R.; Gething, P.W.; Bennett, A.; Sturrock, H.J.W. Mapping land cover change over continental Africa using Landsat and Google Earth Engine cloud computing. *PLOS ONE* **2017**, *12*, e0184926. doi:10.1371/journal.pone.0184926.
27. Phiri, D.; Morgenroth, J. Developments in Landsat Land Cover Classification Methods: A Review. *Remote Sensing* **2017**, *9*, 967. doi:10.3390/rs9090967.
28. Quinlan, J.R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106. doi:10.1023/A:1022643204877.
29. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. doi:10.1023/A:1010933404324.
30. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. *ArXiv* **2016**, *abs/1603.02754*.
31. Arnaldo, I.; O'Reilly, U.M.; Veeramachaneni, K. Building Predictive Models via Feature Synthesis. 2015, pp. 983–990. doi:10.1145/2739480.2754693.

32. Mcconaghy, T., FFX: Fast, Scalable, Deterministic Symbolic Regression Technology; 2011; pp. 235–260. doi:10.1007/978-1-4614-1770-5_13.
33. Liu, H.; Motoda, H. *Feature Extraction, Construction and Selection: A Data Mining Perspective*; Kluwer Academic Publishers: USA, 1998.
34. Sondhi, P. Feature construction methods: a survey. *Sifaka. cs. uiuc. edu* **2009**, *69*, 70–71.
35. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. 2014 Science and Information Conference, 2014, pp. 372–378.
36. Rasan, N.; Mani, D. A Survey on Feature Extraction Techniques. *International Journal of Innovative Research in Computer and Communication Engineering* **2015**, *03*, 52–55. doi:10.15680/ijircce.2015.0301009.
37. Dong, G.; Liu, H. *Feature Engineering for Machine Learning and Data Analytics*, 1st ed.; CRC Press, Inc.: USA, 2018.
38. Swesi, I.M.A.O.; Bakar, A.A., Recent Developments on Evolutionary Computation Techniques to Feature Construction. In *Intelligent Information and Database Systems: Recent Developments*; Huk, M.; Maleszka, M.; Szczerbicki, E., Eds.; Springer International Publishing: Cham, 2020; pp. 109–122. doi:10.1007/978-3-030-14132-5_9.
39. Xue, B.; Zhang, M. Evolutionary computation for feature manipulation: key challenges and future directions. 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 3061–3067.
40. Krawiec, K. Genetic Programming-based Construction of Features for Machine Learning and Knowledge Discovery Tasks. *Genetic Programming and Evolvable Machines* **2002**, *3*, 329–343.
41. Krawiec, K.; Bhanu, B. Coevolutionary Feature Learning for Object Recognition. Machine Learning and Data Mining in Pattern Recognition; Perner, P.; Rosenfeld, A., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2003; pp. 224–238.
42. Krawiec, K.; Włodarski, L. Coevolutionary feature construction for transformation of representation of machine learners. Intelligent Information Processing and Web Mining; Kłopotek, M.A.; Wierzchoń, S.T.; Trojanowski, K., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2004; pp. 139–150.
43. Neshatian, K.; Zhang, M.; Johnston, M. Feature Construction and Dimension Reduction Using Genetic Programming. Proceedings of the 20th Australian Joint Conference on Advances in Artificial Intelligence; Springer-Verlag: Berlin, Heidelberg, 2007; AI'07, p. 160–170.
44. Tran, B.; Xue, B.; Zhang, M. Genetic programming for feature construction and selection in classification on high-dimensional data. *Memetic Computing* **2016**, *8*, 3–15.
45. Tran, C.T.; Zhang, M.; Andreae, P.; Xue, B. Genetic Programming Based Feature Construction for Classification with Incomplete Data. Proceedings of the Genetic and Evolutionary Computation Conference; Association for Computing Machinery: New York, NY, USA, 2017; GECCO '17, p. 1033–1040. doi:10.1145/3071178.3071183.
46. Chen, Q.; Zhang, M.; Xue, B. Genetic Programming with Embedded Feature Construction for High-Dimensional Symbolic Regression. Intelligent and Evolutionary Systems; Leu, G.; Singh, H.K.; Elsayed, S., Eds.; Springer International Publishing: Cham, 2017; pp. 87–102.
47. Tran, B.; Xue, B.; Zhang, M. Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognition* **2019**, *93*, 404 – 417. doi:https://doi.org/10.1016/j.patcog.2019.05.006.
48. Guo, H.; Nandi, A.K. Breast cancer diagnosis using genetic programming generated feature. *Pattern Recognition* **2006**, *39*, 980 – 987. doi:https://doi.org/10.1016/j.patcog.2005.10.001.
49. Ahmed, S.; Zhang, M.; Peng, L.; Xue, B. Multiple Feature Construction for Effective Biomarker Identification and Classification Using Genetic Programming. Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation; Association for Computing Machinery: New York, NY, USA, 2014; GECCO '14, p. 249–256. doi:10.1145/2576768.2598292.
50. Virgolin, M.; Alderliesten, T.; Bel, A.; Witteveen, C.; Bosman, P.A.N. Symbolic Regression and Feature Construction with GP-GOMEA Applied to Radiotherapy Dose Reconstruction of Childhood Cancer Survivors. Proceedings of the Genetic and Evolutionary Computation Conference; Association for Computing Machinery: New York, NY, USA, 2018; GECCO '18, p. 1395–1402. doi:10.1145/3205455.3205604.
51. Ain, Q.U.; Xue, B.; Al-Sahaf, H.; Zhang, M. Genetic Programming for Multiple Feature Construction in Skin Cancer Image Classification. 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ), 2019, pp. 1–6.

52. Cherrier, N.; Poli, J.; Defurne, M.; Sabatié, F. Consistent Feature Construction with Constrained Genetic Programming for Experimental Physics. IEEE Congress on Evolutionary Computation, CEC 2019, Wellington, New Zealand, June 10-13, 2019. IEEE, 2019, pp. 1650–1658. doi:10.1109/CEC.2019.8789937.
53. Gong, P.; Marceau, D.J.; Howarth, P.J. A comparison of spatial feature extraction algorithms for land-use classification with SPOT HRV data. *Remote Sensing of Environment* **1992**, *40*, 137 – 151. doi:https://doi.org/10.1016/0034-4257(92)90011-8.
54. Romero, A.; Gatta, C.; Camps-Valls, G. Unsupervised Deep Feature Extraction for Remote Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing* **2016**, *54*, 1349–1362. doi:10.1109/TGRS.2015.2478379.
55. Ren, J.; Zabalza, J.; Marshall, S.; Zheng, J. Effective Feature Extraction and Data Reduction in Remote Sensing Using Hyperspectral Imaging [Applications Corner]. *IEEE Signal Processing Magazine* **2014**, *31*, 149–154.
56. Pasquarella, V.J.; Holden, C.E.; Woodcock, C.E. Improved mapping of forest type using spectral-temporal Landsat features. *Remote Sensing of Environment* **2018**, *210*, 193 – 207. doi:https://doi.org/10.1016/j.rse.2018.02.064.
57. Puente, C.; Olague, G.; Smith, S.; Bullock, S.; González-Botello, M.; Hinojosa, A. A Genetic Programming Approach to Estimate Vegetation Cover in the Context of Soil Erosion Assessment. *Photogrammetric Engineering and Remote Sensing* **2011**, *77*, 363–375. doi:10.14358/PERS.77.4.363.
58. Makkeasorn, A.; Chang, N.B.; Li, J. Seasonal change detection of riparian zones with remote sensing images and genetic programming in a semi-arid watershed. *Journal of Environmental Management* **2009**, *90*, 1069 – 1080. doi:https://doi.org/10.1016/j.jenvman.2008.04.004.
59. Makkeasorn, A.; Chang, N.B.; Beaman, M.; Wyatt, C.; Slater, C. Soil moisture estimation in a semiarid watershed using RADARSAT-1 satellite imagery and genetic programming. *Water Resources Research* **2006**, *42*.
60. Chion, C.; Landry, J.A.; Costa, L. A Genetic-Programming-Based Method for Hyperspectral Data Information Extraction: Agricultural Applications. *Geoscience and Remote Sensing, IEEE Transactions on* **2008**, *46*, 2446 – 2457. doi:10.1109/TGRS.2008.922061.
61. Chen, L. A study of applying genetic programming to reservoir trophic state evaluation using remote sensor data. *International Journal of Remote Sensing* **2003**, *24*, 2265–2275.
62. Taghizadeh-Mehrjardi, R.; Ayoubi, S.; Namazi, Z.; Malone, B.; Zolfaghari, A.; Roustaiee-Sadrabadi, F. Prediction of soil surface salinity in arid region of central Iran using auxiliary variables and genetic programming. *Arid Land Research and Management* **2016**, *30*. doi:10.1080/15324982.2015.1046092.
63. Chen, L. A study of applying genetic programming to reservoir trophic state evaluation using remote sensor data. *International Journal of Remote Sensing* **2003**, *24*, 2265–2275. doi:10.1080/01431160210154966.
64. Lary, D.J.; Alavi, A.H.; Gandomi, A.H.; Walker, A.L. Machine learning in geosciences and remote sensing. *Geoscience Frontiers* **2016**, *7*, 3 – 10. Special Issue: Progress of Machine Learning in Geosciences, doi:https://doi.org/10.1016/j.gsf.2015.07.003.
65. Costa, L.; Nunes, L.; Ampatzidis, Y. A new visible band index (vNDVI) for estimating NDVI values on RGB images utilizing genetic algorithms. *Computers and Electronics in Agriculture* **2020**, *172*, 105334. doi:https://doi.org/10.1016/j.compag.2020.105334.
66. Kabiri, P.; Pandi, M.H.; Nejat, S.K.; Ghaderi, H. NDVI Optimization Using Genetic Algorithm. 2011 7th Iranian Conference on Machine Vision and Image Processing, 2011, pp. 1–5.
67. Lopes, C.; Leite, A.; Vasconcelos, M.J. Open-access cloud resources contribute to mainstream REDD+: The case of Mozambique. *Land Use Policy* **2019**, *82*, 48 – 60. doi:https://doi.org/10.1016/j.landusepol.2018.11.049.
68. Cabral, A.I.; Silva, S.; Silva, P.C.; Vanneschi, L.; Vasconcelos, M.J. Burned area estimations derived from Landsat ETM+ and OLI data: Comparing Genetic Programming with Maximum Likelihood and Classification and Regression Trees. *ISPRS Journal of Photogrammetry and Remote Sensing* **2018**, *142*, 94 – 105. doi:https://doi.org/10.1016/j.isprsjprs.2018.05.007.
69. Vasconcelos, M.; Cabral, A.; B. Melo, J.; Pearson, T.; Pereira, H.; Cassamá, V.; Yudelman, T. Can blue carbon contribute to clean development in West Africa? The case of Guinea-Bissau. *Mitigation and Adaptation Strategies for Global Change* **2014**, *20*. doi:10.1007/s11027-014-9551-x.

70. Temudo, M.; Cabral, A.; Talhinhas, P. Petro-Landscapes: Urban Expansion and Energy Consumption in Mbanza Kongo City, Northern Angola. *Human Ecology* **2019**, *47*, 565–575. doi:10.1007/s10745-019-00088-6.
71. Tibshirani, R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **1996**, *58*, 267–288.
72. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* **2010**, *33*. doi:10.18637/jss.v033.i01.
73. Pedregosa et al, F. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.