Internet of Things 9 (2020) 100141

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Review article

Polly: A Tool for Rapid Data Integration and Analysis in Support of Agricultural Research and Education

Waqar Muhammad[a,*], Flavio Esposito[a], Maitiniyazi Maimaitijiang[b], Vasit Sagan[b], Enrico Bonaiuti[c]

[a] *Department of Computer Science Saint Louis University, Saint Louis, Missouri, USA*
[b] *Department of Earth and Atmospheric Sciences, Saint louis university, Saint Louis, Missouri, USA*
[c] *International Center for Agricultural Research in the Dry Areas (ICARDA), Beirut, Lebanon*

ARTICLE INFO

Article history:
Received 15 August 2019
Revised 16 November 2019
Accepted 16 November 2019
Available online 26 November 2019

Keywords:
smart farming
serverless computing
network virtualization
machine learning
UAV

ABSTRACT

Data analysis and modeling is a complex and demanding task. While a variety of software and tools exist to cope with this problem and tame big data operations, most of these tools are either not free, and when they are, they require large amount of configuration and steep learning curve. Moreover, they provide limited functionalities. In this paper we propose Polly, an online data analysis and modeling open-source tool that is intuitive to use and can be used with minimal or no configuration. Users can use Polly to rapidly integrate, analyze their data, prototype and test their novel methodologies. Polly can be used also as an educational tool. Users can use Polly to upload or connect to their structured data sources, load the required data into our system and perform various data processing tasks. Examples of such operations include data cleaning, data pre-processing, attribute encoding, regression and classification analysis. Aside from modeling, users can then download their results in the form of graphs in several standard visualization formats. While in this paper we focus on analyzing dataset for smart farming, our tool usage fits to a more general audience. To justify our backend design and implementation choices, we also present a performance analysis between backend virtualization technologies (containers or serverless computing), showing both expected and surprising results.

1. Introduction

The increasing demand for agricultural products in the context of climate change and population growth [1] is propelling the need for smart farming solutions. Smart farming has been shown to increase the quantity and quality of agricultural productivity. Big data and related web and IoT technologies such as machine learning are playing an essential role and enabling many opportunities for smart farming [2].

Recent advances in UAV (Unmanned Aerial Vehicle) and imaging sensors, as well as autopilots, and GPS systems, have enabled collection of large volumes of data with high spatial, spectral, and temporal resolution from agricultural fields, allowing fast and accurate estimation of biophysical and biochemical plant traits (e.g., chlorophyll content, height, biomass,

[*] Corresponding author. Tel.: +13144459273.
E-mail address: waqar.muhammad@slu.edu (W. Muhammad).

https://doi.org/10.1016/j.iot.2019.100141

and photosynthesis) that are important indicators of plant stress and health, and prediction of crop yield [3]. Additionally, combining UAV big data with other data sources such as environmental and meteorological variables sensed with IoT devices, as well as related communication network, automation and supporting platforms and systems, smart farming is able to provide significant predictive insights in farming management and sustainable agriculture, risk management and decision-making [2], and also benefit crop productivity and food security concerns.

To answer some of these questions related to smart farming, mining and analyzing data is the key. Complex and computationally intensive tasks require multiple steps to be performed successfully, from IoT based data gathering, to data cleaning and manipulation, for statistical analysis. Researchers often perform repetitive tasks that involves time consuming and tedious data cleaning operations.

The rapid growth of data sets in recent years exacerbated these problems. To quickly make decisions and obtain a basic understanding of datasets in hand, these tasks need to be performed in a timely fashion. Examples of questions are awareness of whether or not we need to collect more data from given Flying Ad-Hoc Network (FANET) sources, or the data that our UAV have collected is sufficient to answer our hypothesis.

In our experience, often plant or climate scientists rely on proprietary cloud platforms and data analysis tools such as RapidMiner [4], Weka [5] or SPSS [6] which either have very limited functionalities or require purchase of expensive licenses.

To cope with the lack of versatile and open-source tools, we designed and implemented *Polly*, an online data analysis and modeling tool which eliminate the barrier for effective data mining. This system is designed with current technologies that allows data integration, processing and analysis using serverless architecture [7,8], coupled with a Cassandra No-SQL database. We built a set of API so that users can access a plethora of data analysis tools *without any coding knowledge* such as linear, support vector, elastic and logistic regression. We test our Polly tool on several datasets, including on a set of Missouri Transect EPSCoR Plant Team datasets [1]. Our aim is to demonstrate how to answer a few representative smart farming data science questions. To that end, we demonstrate the use our tool Polly to run several regression models with the aim of estimating different plant biochemical and biophysical traits *e.g.*, chlorophyll and nitrogen concentration that are indicative of plant growth, vigor, and yield.

2. Related Work

Various tools such as rapid miner [4], tableau [9], Weka [5], Power Bi [10] have been created to provide users a platform to analyze, clean, visualize, and model their data in an online or offline manner. All these tools have limitations, and they are often tailored to specific applications.

RapidMiner is suitable for users who have data science experience. It has a powerful visualization tableau: it provides users with various visualization options, although without the ability to download (at least the version that we have analyzed). It also allows fundamental data preprocessing; this includes joining data sets. RapidMiner is not extremely intuitive in our opinion, so its use in education would be challenging. The use cases covered by the tool are limited to the set of processors/modules that it contains as it runs on the host machine; as a consequence, it consumes large sets of memory, which in turn slows down primary user's systems. Moreover, aside from visualizing data, in most cases, analysts need to clean or reformat data. Such data cleaning step is often performed with external tools such as Altyrex, Power BI, Python, or even Excel.

Also, for data visualization, it is not an ideal platform to use, and their licensing fees are steep. To give a general idea of licensing fees associated with this or other tools, we mention Tableau, one of the most expensive options. If users have security and sharing constraints, the only option is Tableau Server that can cost $175,000 for an 8-core option, and $35 dollars per user. Alternatively, the company offers a Tableau Online version, which is limited but also costs $35 per user (quotes as of November 2019).

Another powerful and popular tool is Power BI which advertises itself as "a solution that places business intelligence creation into the hands of analysts who can extract source data, create a dataset, transform or manipulate the data, visualize the data and publish the resulting reports and dashboards". We found that their Dashboards and reports can only be shared with users who have the same email domains or email domains listed within an Office 365 tenant. While a dataset can include multiple data types, Power BI reports and dashboards can only source data from a single dataset. Another limitation that we found is the inability to import datasets larger than 100k records, at a reasonable price. Our tool is opensource, and does not have these hardware limitations.

3. Polly Architecture

Overview. Our architecture consists of several components. A user interface, where users can upload their data (in excel or CSV format) to apply machine learning algorithms, select a different number of parameters to be used as training variables and specify a target prediction variable. After the attribute selection phase, a user can perform a variety of statistical data analysis and cleaning operations. Once the pre-processing dataset phase is complete, the user is asked to select among a set of available machine learning algorithms. Once the algorithm is selected, a user can tune each hyperparameter to satisfy the application requirements. The front-end interface transfers each of these operation calls to the web-service written

[1] https://data.missouriepscor.org/.

in python using Flask [11], a lightweight web framework. Our architecture is designed with scalability in mind: each Polly processing call runs on a separate Docker container, allowing responsiveness and flexible modeling, data cleaning and data analysis.

3.1. Application Workflow

In this subsection, we present the details of the workflow of our tool. In particular, we describe all steps necessary to use our tool, Polly. We divide our workflow into four steps: (1) a data feeding step, (2) a statistical analysis and data processing step, (3) an algorithm selection and tuning step, and finally a (4) result view steps (or phases).

Data feeding. In this phase, users can either upload their data in excel format or can connect to the remote SQL database by providing an IP, a port, a database name, and the name of the table they wish to import. After importing their dataset, users will have the possibility to select a set of training variables and specify the target/estimation variable (if they wish to perform a machine learning operation on the dataset).

Statistical Analysis and Data Processing. In this step, users can view basic statistics of their dataset, which is automatically performed by our system right after data upload. These analyses include computation of Mean, Mode, Median, Standard deviation, all percentile values and a count of null values of each data-column. Apart from these basic stats, we have equipped our Polly with the capability of solving the missing values problem. By merely selecting the attributes, users can fill the missing values using methods like mean, mode, median, standard deviation, most frequent, and can even drop missing columns and rows from the data-set.

This step also includes an attribute encoding option, that can help users convert string categorical attributes in their data to numbers using categorical or one-hot encoding, a notation that can be very useful during a data training model of some prediction algorithm.

Algorithm Selection and Tuning. This step allows users to select the machine learning algorithm to run two types of analysis. In Polly, we support both Regression and Classification Analysis. Within the regression analysis, we have embedded six widely used algorithms: A Linear, Lasso, Lars, LassoLars (LLR), Elastic Net Regression (ENR) and Support Vector Regression (SVR). For classification analysis, we have embedded six popular algorithms: Gaussian Naive Bayes (GNB), Bernoulli Naive Bayes, Ada Booster, Decision Tree, Random Forest Tree and Support Vector Classifier. Finally, users can tune the hyperparameter space of the selected algorithms.

Results and Data View. In this step, users can evaluate the results of their selected algorithm with the help of metrics defined by each analysis type, as well as compare the performance of each selected algorithm. Then Polly can be used to plot a series of graphs, automatically, to give insight about what data points used for training the algorithm, as well as to evaluate the performance of the algorithms and what are the results of those predictions, for each algorithm.

4. Use Case: UAV-based Phenotyping of Soybean

To demonstrate our tool, we have tested data integration and machine learning features of Polly on a set of soybean plant biophysical and biochemical data collected in an open agricultural field in Missouri, USA. Specifically, we demonstrated Polly's regression and machine learning capabilities in estimating leaf chlorophyll & nitrogen concentration, leaf area index, above-ground fresh biomass and dry biomass of soybean.

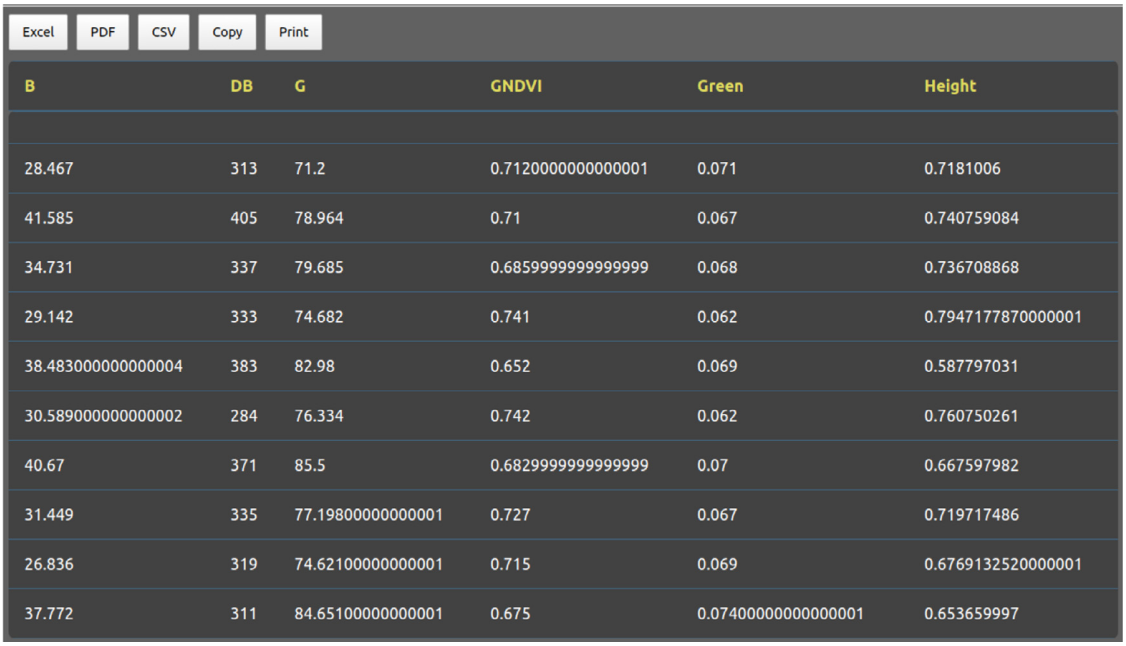
4.1. Multi-sensor aerial images and crop biochemical & biophysical traits dataset

Plant science dataset collection. UAV imagery and plant traits mentioned above were collected from the soybean (Glycine max) experimental field at the Bradford Research Center of the University of Missouri, Columbia, USA. Multispectral, visual, and infrared cameras mounted on a DJI S1000 + octocopter, including Parrot Sequoia multispectral, Sony Alpha ILCE-7R RGB, and the ICI 8640 P-series thermal cameras, were used to collect aerial images of the field. A set of plant biophysical and biochemical traits that are important indicators of plant growth was compiled by in-situ hand measurement or laboratory analysis of destructively collected samples. These traits include chlorophyll content (Chl a, Chl b and Chl a+b (g cm-2)) and nitrogen concentration (N), Leaf Area Index (LAI) and above ground fresh biomass (FB (g m-2)) and dry biomass (FD (g m-2)).

Spectral indices were computed from UAV images following rigorous image pre-processing involving orthorectification, mosaicking and radiometric calibration using the Pix4Dmapper software package (Pix4D SA, Lausanne, Switzerland). In addition, canopy structural features such as plant height (PH) was extracted using RGB imagery-based the photogrammetric point clouds; canopy temperature was computed from UAV thermal imagery. Average pixel values for each plot were calculated zonal statistics and exported to excel format data, which was then imported in the Polly tool.

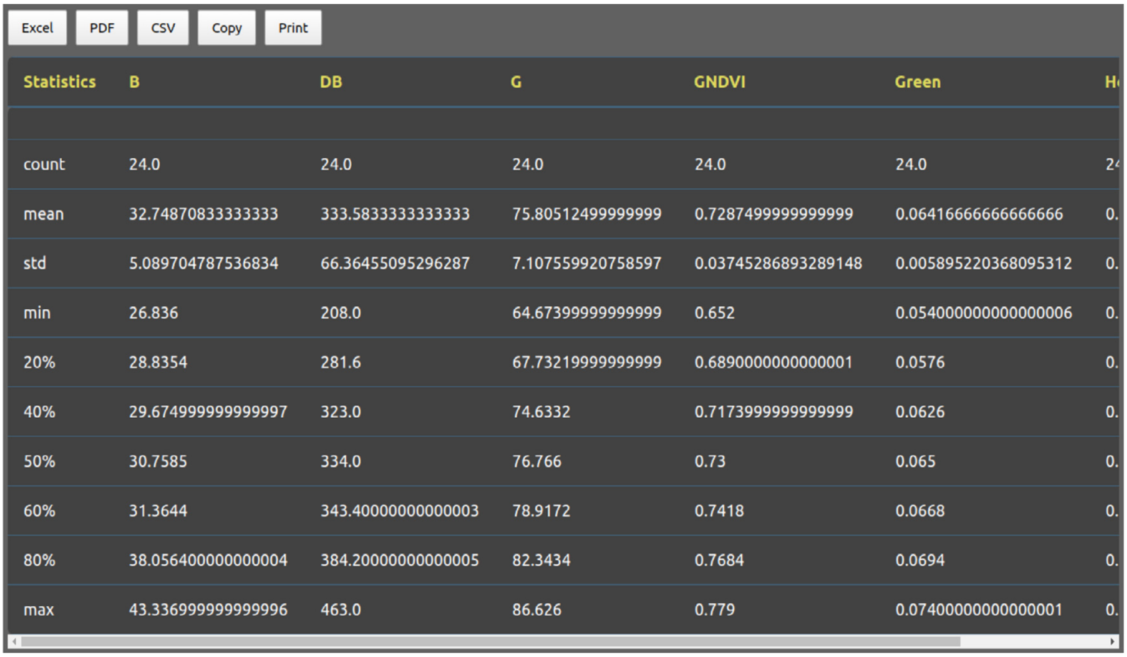
4.2. Process Overview

The Multi-sensor aerial imagery and plant traits data are analyzed by using scale-able options provided by the tool as per our requirement. This section explains how the data is processed and the results obtained for analysis.



B	DB	G	GNDVI	Green	Height
28.467	313	71.2	0.7120000000000001	0.071	0.7181006
41.585	405	78.964	0.71	0.067	0.740759084
34.731	337	79.685	0.6859999999999999	0.068	0.736708868
29.142	333	74.682	0.741	0.062	0.7947177870000001
38.483000000000004	383	82.98	0.652	0.069	0.587797031
30.589000000000002	284	76.334	0.742	0.062	0.760750261
40.67	371	85.5	0.6829999999999999	0.07	0.667597982
31.449	335	77.19800000000001	0.727	0.067	0.719717486
26.836	319	74.62100000000001	0.715	0.069	0.6769132520000001
37.772	311	84.65100000000001	0.675	0.07400000000000001	0.653659997

Fig. 1. Uploaded data and selected attributes.



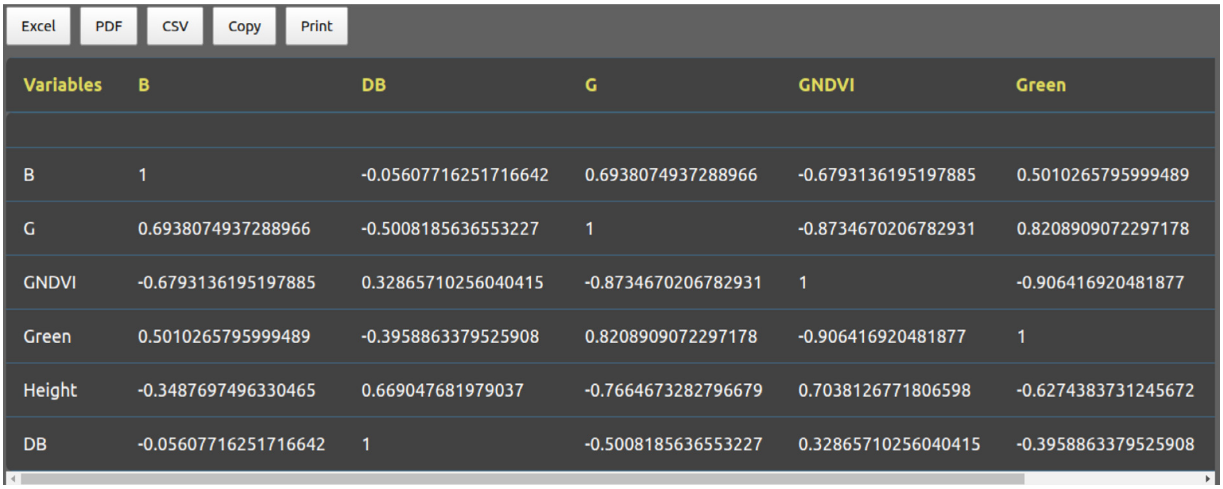
Statistics	B	DB	G	GNDVI	Green	H
count	24.0	24.0	24.0	24.0	24.0	2
mean	32.74870833333333	333.5833333333333	75.80512499999999	0.7287499999999999	0.06416666666666666	0.
std	5.089704787536834	66.36455095296287	7.107559920758597	0.03745286893289148	0.005895220368095312	0.
min	26.836	208.0	64.67399999999999	0.652	0.054000000000000006	0.
20%	28.8354	281.6	67.73219999999999	0.6890000000000001	0.0576	0.
40%	29.674999999999997	323.0	74.6332	0.7173999999999999	0.0626	0.
50%	30.7585	334.0	76.766	0.73	0.065	0.
60%	31.3644	343.40000000000003	78.9172	0.7418	0.0668	0.
80%	38.056400000000004	384.20000000000005	82.3434	0.7684	0.0694	0.
max	43.336999999999996	463.0	86.626	0.779	0.07400000000000001	0.

Fig. 2. Statistical description of data obtained from Polly.

4.2.1. Statistical Analysis

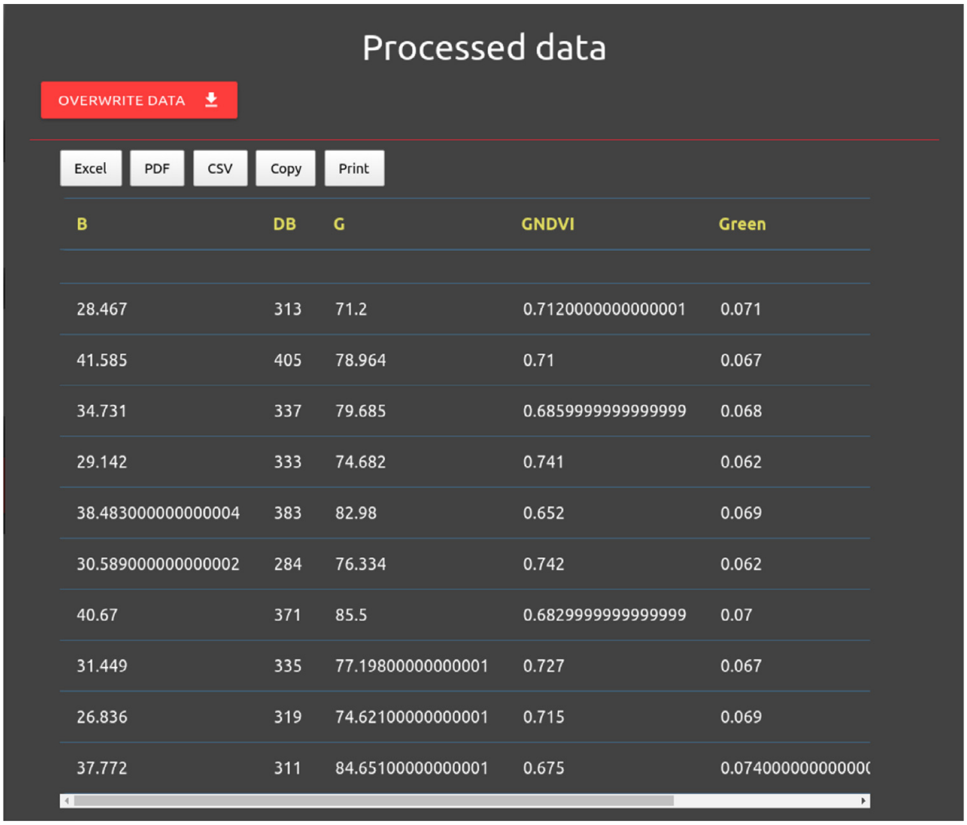
First, the data in excel format is uploaded and can be viewed in the tool (Fig. 1). DB is selected as the testing variable whereas B, G, GNDVI, Green, and Height are chosen as variables for training. For better understanding the variability and distribution of multi-sensor aerial imagery and plant traits data, our tool shows a set of collective descriptive statistics. These stats include mean, standard deviation, 25th to 75th percentile values and more (Fig. 2).

Apart from descriptive statistics, our tool provides a correlation matrix that can be used to identify the relationship and strength of association among different variables in the dataset. In our data, it is clearly visible that the majority of



Variables	B	DB	G	GNDVI	Green
B	1	-0.05607716251716642	0.6938074937288966	-0.6793136195197885	0.5010265795999489
G	0.6938074937288966	-0.5008185636553227	1	-0.8734670206782931	0.8208909072297178
GNDVI	-0.6793136195197885	0.32865710256040415	-0.8734670206782931	1	-0.906416920481877
Green	0.5010265795999489	-0.3958863379525908	0.8208909072297178	-0.906416920481877	1
Height	-0.3487697496330465	0.669047681979037	-0.7664673282796679	0.7038126771806598	-0.6274383731245672
DB	-0.05607716251716642	1	-0.5008185636553227	0.32865710256040415	-0.3958863379525908

Fig. 3. Correlation Matrix.


Processed data

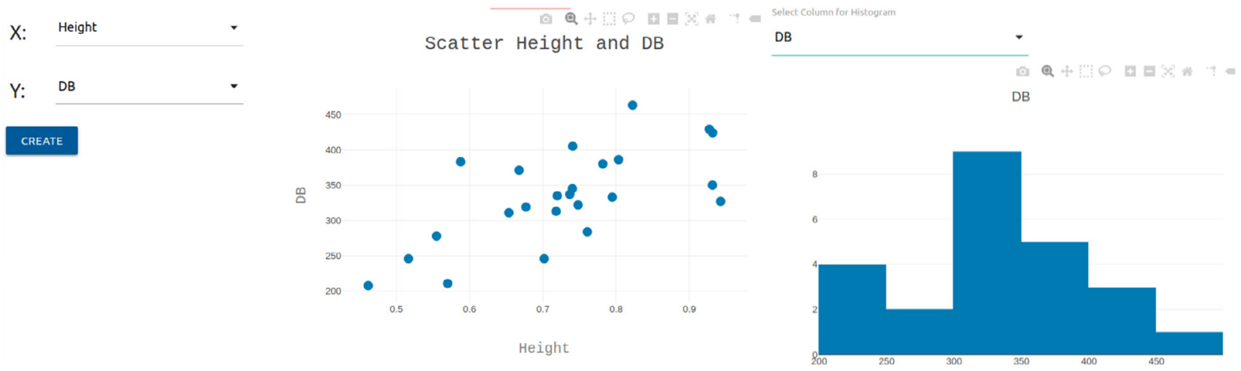
B	DB	G	GNDVI	Green
28.467	313	71.2	0.7120000000000001	0.071
41.585	405	78.964	0.71	0.067
34.731	337	79.685	0.6859999999999999	0.068
29.142	333	74.682	0.741	0.062
38.483000000000004	383	82.98	0.652	0.069
30.589000000000002	284	76.334	0.742	0.062
40.67	371	85.5	0.6829999999999999	0.07
31.449	335	77.19800000000001	0.727	0.067
26.836	319	74.62100000000001	0.715	0.069
37.772	311	84.65100000000001	0.675	0.07400000000000(

Fig. 4. Data decision after applying data processing steps such as handling missing values.

the variables hold both strong positive and negative correlation which gives us confirmation that the selected variables are significantly important to carry out this study (Fig. 3).

4.2.2. Data Processing

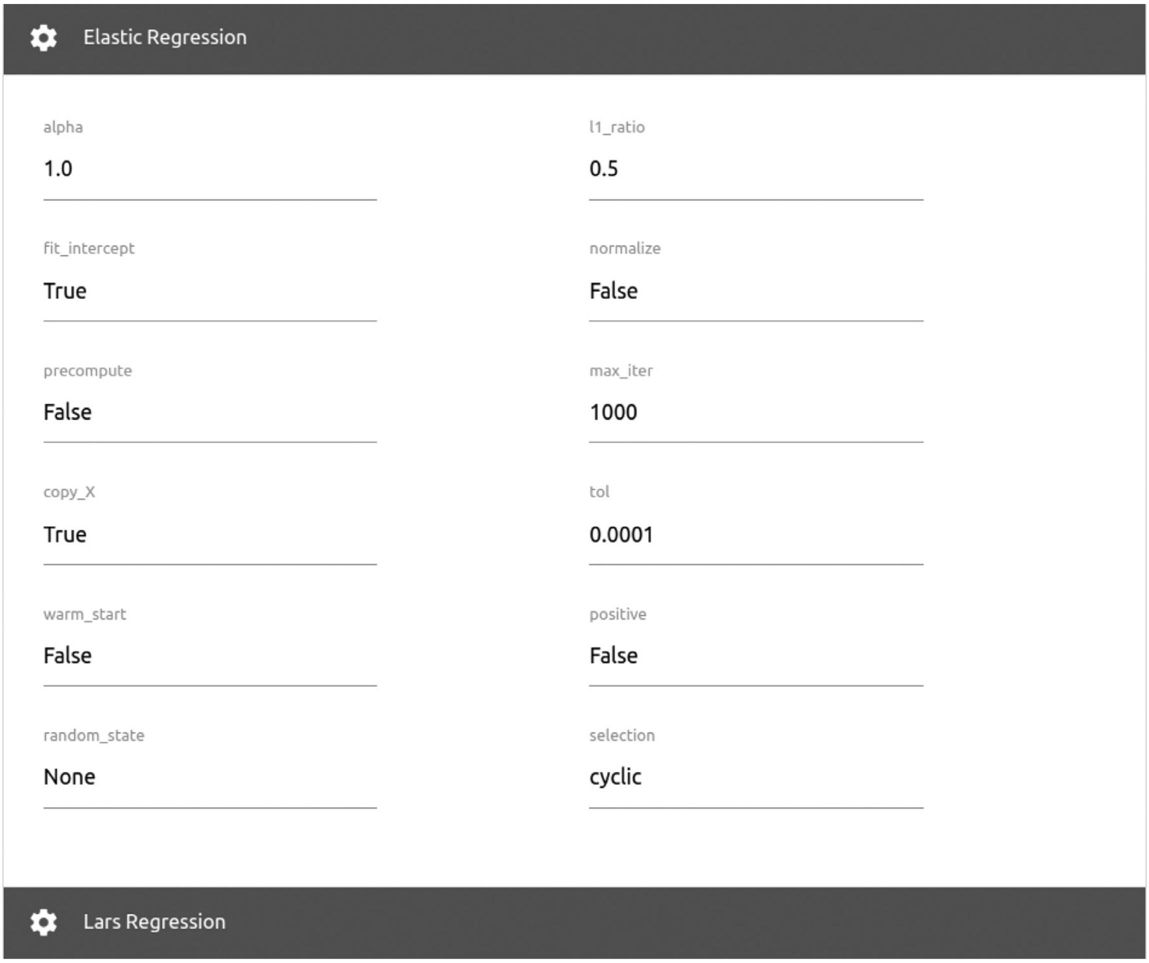
As missing observations have a significant impact on the training of the model, we pre-processed our data in order to handle the missing values through methods provided by the tool that includes replacing with mean, median, mode, most frequent or drop missing values (Fig. 4). Each method was applied and used before selection to see which yielded better results The processed data can then be viewed, downloaded or over-written to be utilized for training of our model.



X: Height

Y: DB

Fig. 5. Data distribution and association between selected variables.



Elastic Regression

alpha: 1.0	l1_ratio: 0.5
fit_intercept: True	normalize: False
precompute: False	max_iter: 1000
copy_X: True	tol: 0.0001
warm_start: False	positive: False
random_state: None	selection: cyclic

Lars Regression

Fig. 6. Polly allows tuning of parameters of several machine learning algorithms.

After the data has been pre-processed, we generated a few histograms to see the deviation and distribution of our variable values critical for identifying trends. Moreover, scatter plots were also generated through which we were able to spot the potential association among different variables in our data. These graphical representations serve as an efficient and quick way to describe data (Fig. 5).

4.2.3. *Algorithm Selection and Parameter Tuning*

Our tool provides two modes of analysis, classification, and regression. We use regression in our use case. Our tool supports six widely use algorithms for each analysis mode listed below. In terms of regression, it supports, linear, lasso, lars, lasso-lars regression (LLR), Elastic net regression (ENR) and Support vector regression (SVR).

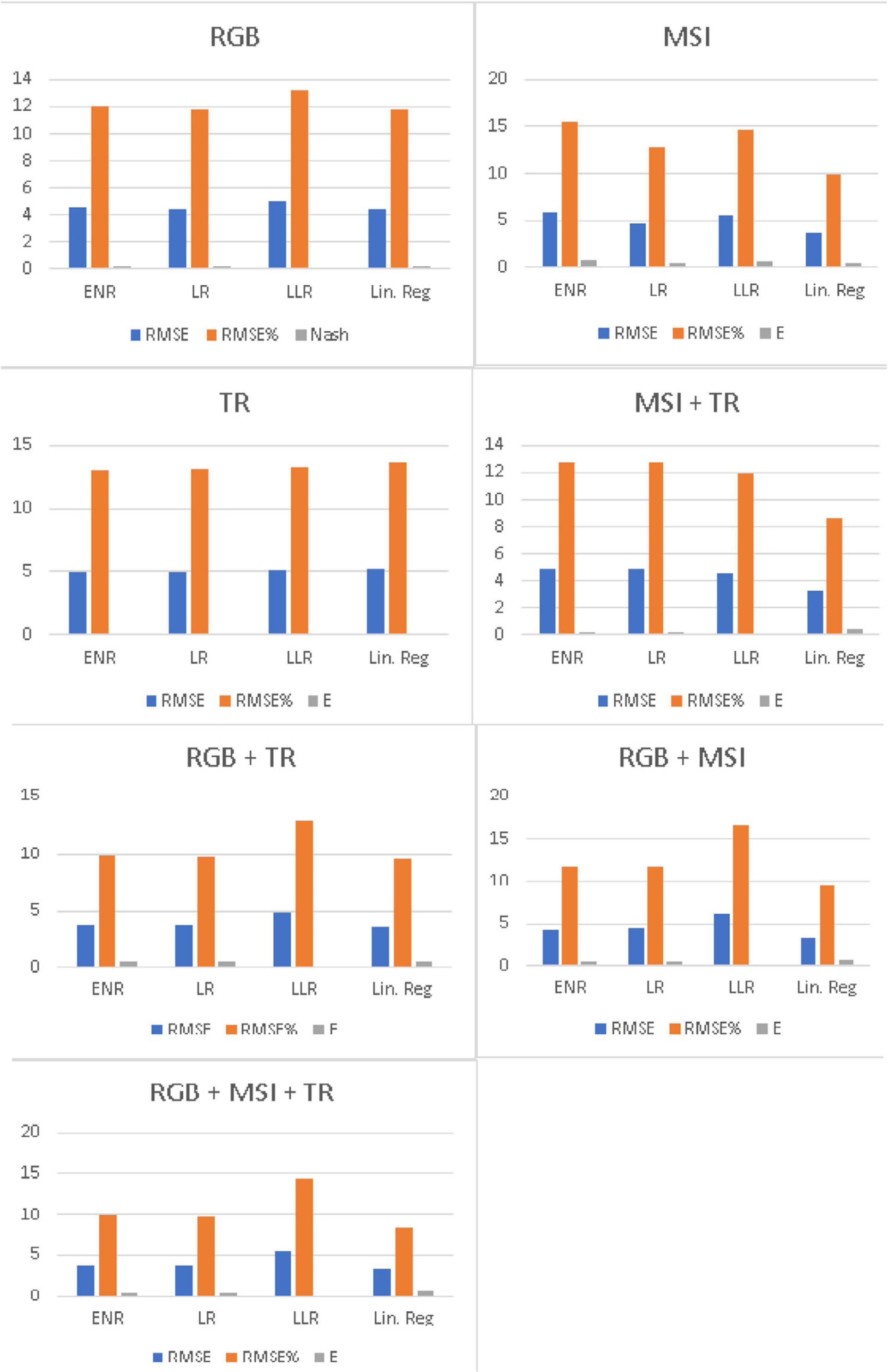


Fig. 7. Metric Comparison plants triats (Chl-a) prediction based on parameters (R, G, B, INT, IKAW, IPCA, Green, Red, RE, NIR, NDVI, GNDVI, NDRE and Tc) using Polly.

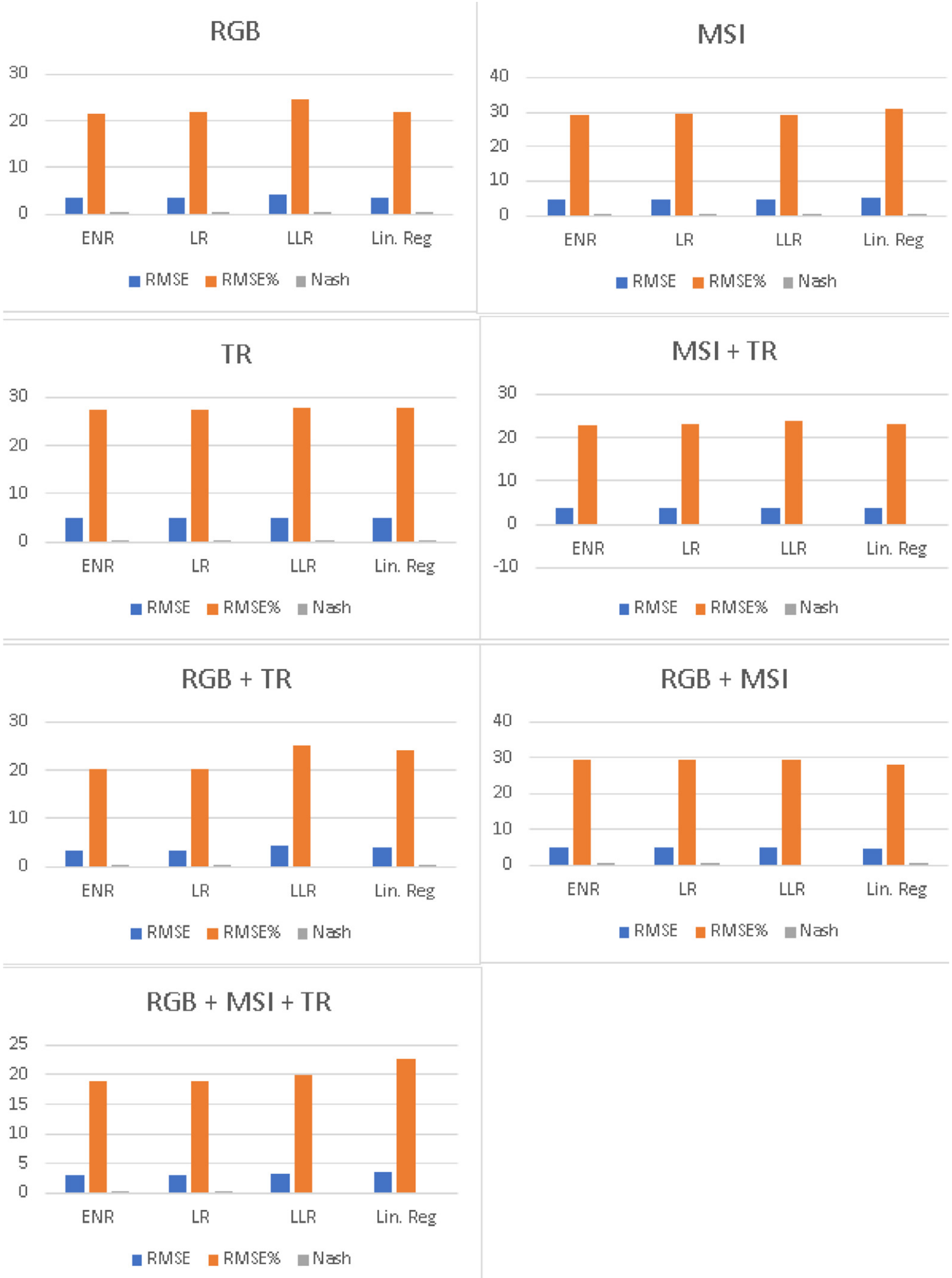


Fig. 8. Using Polly, we performed a metric comparison of plants triats (Chl-b) prediction, using based on the following parameters: R, G, B, INT, IKAW, IPCA, Green, Red, RE, NIR, NDVI, GNDVI, NDRE and Tc.

In terms of classification, our tool supports Gaussian Nave Bayes, Bernoulli Nave Bayes, Ada Booster, Decision Trees, Random Forest Trees, and Support Vector Classification.

These algorithms require to be tuned over different parameters as the performance of the algorithms highly depends on the chosen parameters for model creation. Using the tool we were able to modify a number of parameter values for the selected algorithm (Fig. 6). The tool uses default values if in case no parameter is tuned.

Table 1
R, G, and B represent red, green, and blue bands of RGB sensor, respectively. INT is color intensity index, IKAW is Kawashima index, IPCA is principal component analysis index, PH is plant height, VF is vegetation fraction; Green, Red, RE, and NIR represent green, red, red-edge and near-infrared bands of the multispectral sensor, respectively. NDVI is normalized difference vegetation index, GNDVI is green normalized difference vegetation index, NDRE is normalized difference red-edge index; Tc stats for plant canopy temperature.

Image	Index	Acronym	Equation
RGB	Red	R	R
	Green	G	G
	Blue	B	B
	Color intensity	INT	(R + G+B)/3
	Kawashima index	IKAW	(R B)/(R + B)
	Principal component analysis index	IPCA	$0.994\|RB\| + 0.961\|GB\| + 0.914\|GR\|$
MSI	Green Band	Green	Green
	Red Band	Red	Red
	Red-edge Band	RE	RE
	Near-infrared Band	NIR	NIR
	Normalized difference vegetation index	NDVI	(NIR R)/(NIR + R)
	Green Normalized difference vegetation index	GNDVI	(NIR G)/(NIR + G)
	Normalized difference red edge	NDRE	(NIR RE)/(NIR + RE)
TR	Canopy Temperature	Tc	Tc

Table 2
List of features extracted from UAV-based RGB, multispectral, and thermal images.

Sensor/Info.	Features	Formulation	References
MSI (Sp Features)	Green (G), Red (R), Red-edge (RE), Near-infrared (NIR)	The raw value of each band	/
	Ratio vegetation index	RVI = NIR / R	(Tucker, 1979)
	Green chlorophyll index	GCI = (NIR /G) - 1	(Gitelson et al. 2005)
	Red-edge chlorophyll index	RECI = (NIR /RE) - 1	(Gitelson et al. 2005)
	Normalized difference vegetation index	NDVI = (NIR - R) / (NIR + R)	(Rouse Jr et al. 1974)
	Green normalized difference vegetation index	GNDVI = (NIR - G) / (NIR + G)	(Gitelson et al. 2003)
	Green-red vegetation index	GRVI = (G - R) / (G + R)	(Tucker 1979)
	Normalized difference red-edge	NDRE = (NIR RE) / (NIR + RE)	(Gitelson and Merzlyak 1997)
	Normalized difference red-edge index	NDREI = (RE - G) / (RE + G)	(Hassan et al. 2018)
	Simplified canopy chlorophyll content index	SCCCI = NDRE / NDVI	(Raper and Varco 2015)
	The enhanced vegetation index	EVI=2.5*(NIR-R)/(1+NIR-2.4*R)	(Huete et al. 2002)
	Two-band enhanced vegetation index	EVI2=2.5*(NIR-R)/(NIR+2.4*R+1)	(Jiang et al. 2008)
	The enhanced vegetation index	EVI=2.5*(NIR-R)/(1+NIR-2.4*R)	(Huete et al. 2002)
	Optimized soil adjusted vegetation index	OSAVI=(NIR-R)/(NIR-R+L) (L=0.16)	(Rondeaux et al. 1996)
	Modified chlorophyll absorption in reflectance index	MCARI=[(RE-R)-0.2*(RE-G)] *(RE/R)	(Daughtry et al. 2000)
	Transformed chlorophyll absorption in reflectance index	TCARI=$3*[(RE-R)-0.2*(RE-G)*(RE/R)]$	(Haboudane et al. 2002)
	MCARI/OSAVI	MCARI/OSAVI	(Daughtry et al. 2000)
	TCARI/OSAVI	TCARI/OSAVI	(Haboudane et al. 2002)
	Wide dynamic range vegetation index	WDRVI=$(a*NIR-R)/(a*NIR+R)(a=0.12)$	(Gitelson 2004)
RGB	Plant Height (m) Band	PH = DSM - DEM	/
MSI (St features)	Vegetation fraction (%)	VF=(Number of Crops ∈ the plot / Total number of plot pixels) * 100	(Torres-Sanchez et al. 2014)
	-	-	-
	-	-	-
TR(Th features)	Normalized relative canopy temperature index	$NRCT=T_i-T_{min}/T_i-T_{max}$	(Elsayed et al. 2015)
	-	-	-
	-	-	-
MSI+RGB +TIR (Te features)	Gray-level co-occurrence matrix (GLCM)	/	(Haralick and Shanmugam 1973)
	-	-	-
	-	-	-
	-	-	-

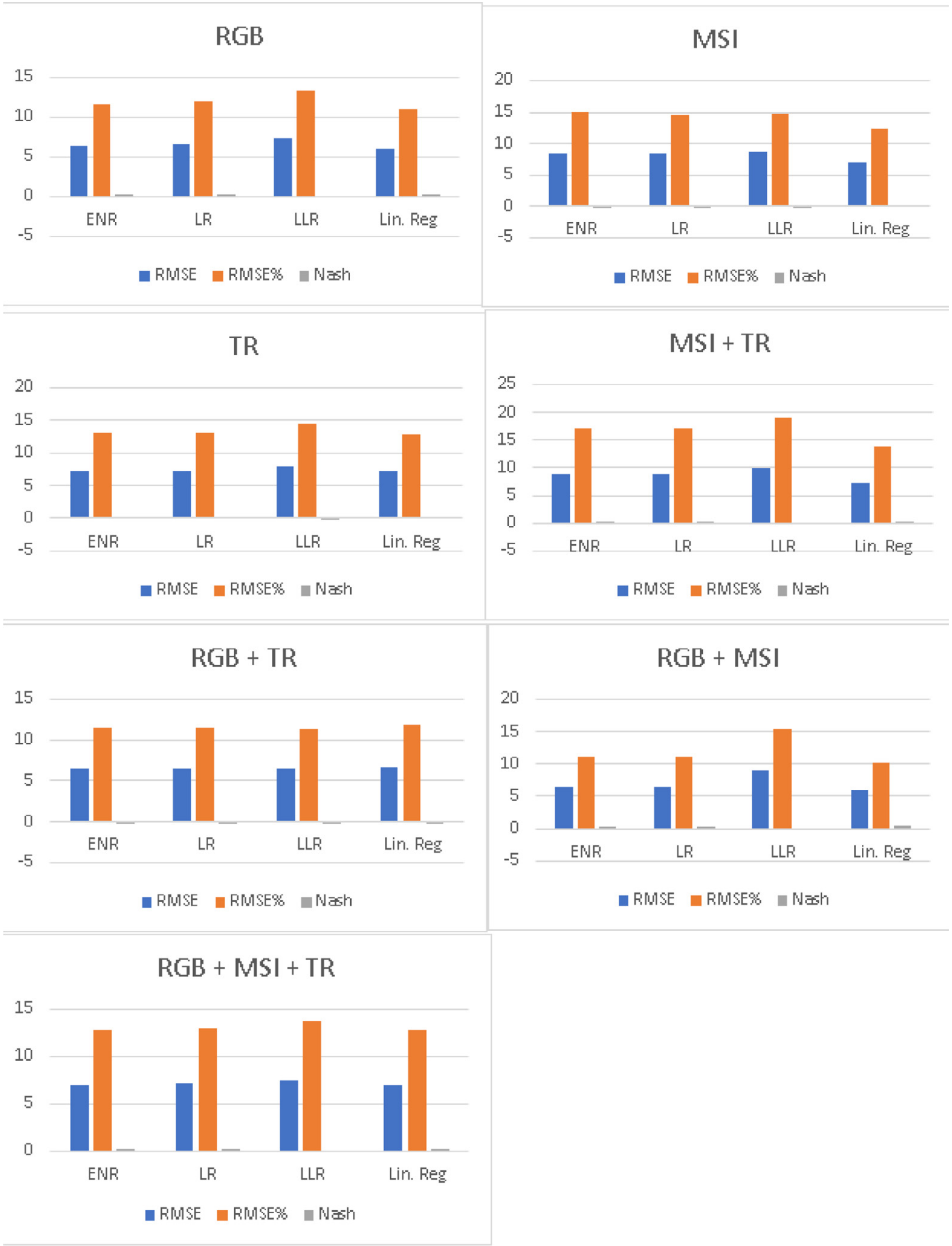


Fig. 9. Metric Comparison of plants triats (Chl-a+b) prediction based on the following parameters: R, G, B, INT, IKAW, IPCA, Green, Red, RE, NIR, NDVI, GNDVI, NDRE and Tc.

4.3. Multi-sensor aerial images and crop grain yield dataset

Even in this data collection phase, we acquired multi-sensor aerial images from the same soybean (Glycine max) experimental field at the University of Missouri Bradford Research Center near Columbia, Missouri, USA. In this case, however, we used A DJI S1000 + octocopter UAV integrated with Parrot Sequoia multispectral, Mapir Survey-2 RGB, and FLIR Vue Pro R 640 thermal cameras were employed to collect multispectral (MSI), RGB, and thermal images (TR) over this field. Soybean grain yield data was obtained by a small-plot combine harvester (ALMACO SPC40, ALMACO, Nevada, IA) in October 2017 from this field. The Pix4Dmapper software (Pix4D SA, Lausanne, Switzerland) was employed to orthorectify, mosaic and

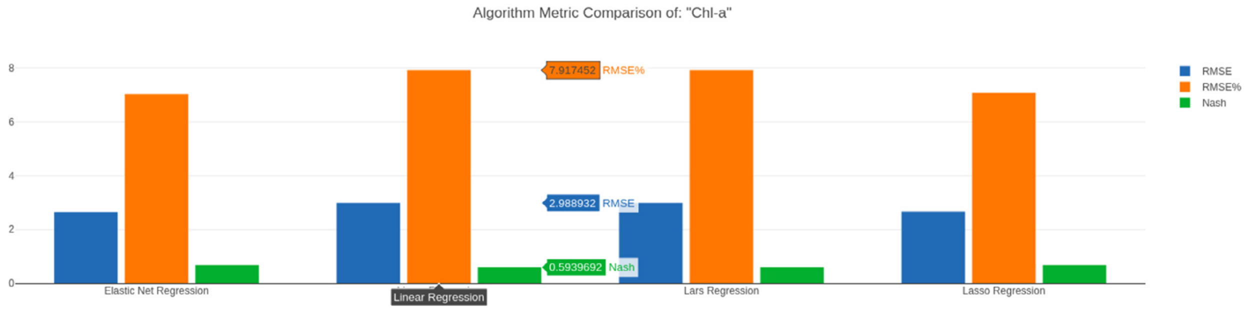


Fig. 10. Regression metric analysis to find detailed information about the algorithm coverage.

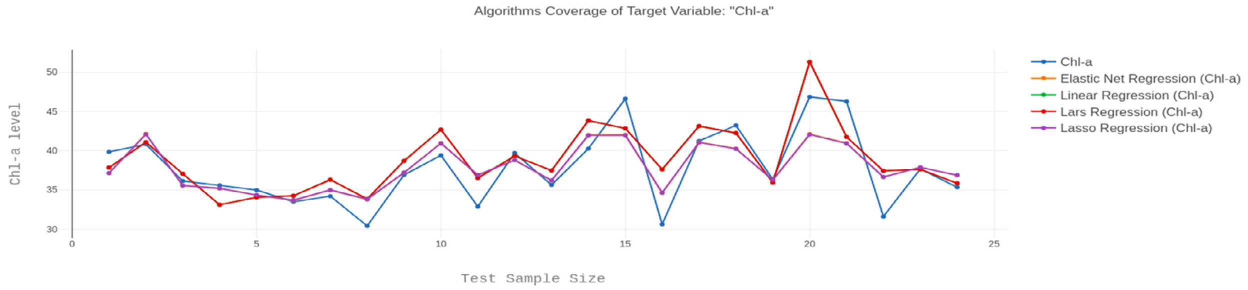


Fig. 11. Performance analysis of machine learning algorithm with respect to their accuracy in predicting the target label correctly.

radiometrically correct the UAV RGB, MSI and TR images. UAV imagery-based canopy spectral features (Sp) mainly represented by VIs were calculated from MSI; Additionally, canopy structural features (St) such as PH and VF were extracted from RGB and MSI images, and the commonly used grey level co-occurrence matrix (GLCM) texture features (Te) were computed; Moreover, the normalized relative canopy temperature feature (Th) was derived as well. For each of these raster feature layers, average pixel values were calculated for each corresponding grain yield harvest plot by zonal statistics and exported to excel format data.

5. Serverless Computing to Speedup Web-based Plant Scientist Operations

Serverless computing is a modern network virtualization technology that is promising to speed up time-to-response for web services by only virtualizing what is needed at the application level. Several cloud providers such as Amazon (with Amazon Lambda) IBM, (with Openwhisk), or Google (with Google Functions) today offer this service. Since its commercial inception, when Amazon launched its Lambda platform, serverless became a mainstream service, attracting big customers such as Netflix, Codepen, Nord Storm, and many more.

Our analysis shows that, while commercial solutions are ready to accept contracts, the opensource serverless scene is not always the fastest alternative to virtual machines. In this experimental section, we assessed when is serverless the best Cloud solution to analyze large datasets. The outcome of our experimental evaluation was then taken into account to build our system, currently available at [12].

Using the back-end of our system architecture, we conducted a performance comparison to test when, or until when, serverless is faster than standard container solutions such as Docker. For our tests, we used the opensource platform OpenWhisk [13] and tested its performance with python3.6 as a runtime.

Experiment Design. Starting from datasets collected with Flying Ad hoc Network (FANETs), we process an integrated dataset for smart agriculture using multiple iterations of multiple machine learning algorithms, namely, Linear regression, Support Vector Regression, Elastic Net regression, Lars regression, and Lasso regression. We then compared the dataset execution time on both Docker and a serverless platform. We conducted 20 iterations for each experiment, and we report the running time.

Main Findings. We note that whenever a machine learning algorithm is trained against a large number of attributes or a large dataset, the execution time of serverless is much better than that of Docker, but surprisingly, when a small dataset or a low number of attributes is provided to train, Docker containers outperforms serverless. The results of the experiments conducted are shown in the graphs below (Fig. 13).

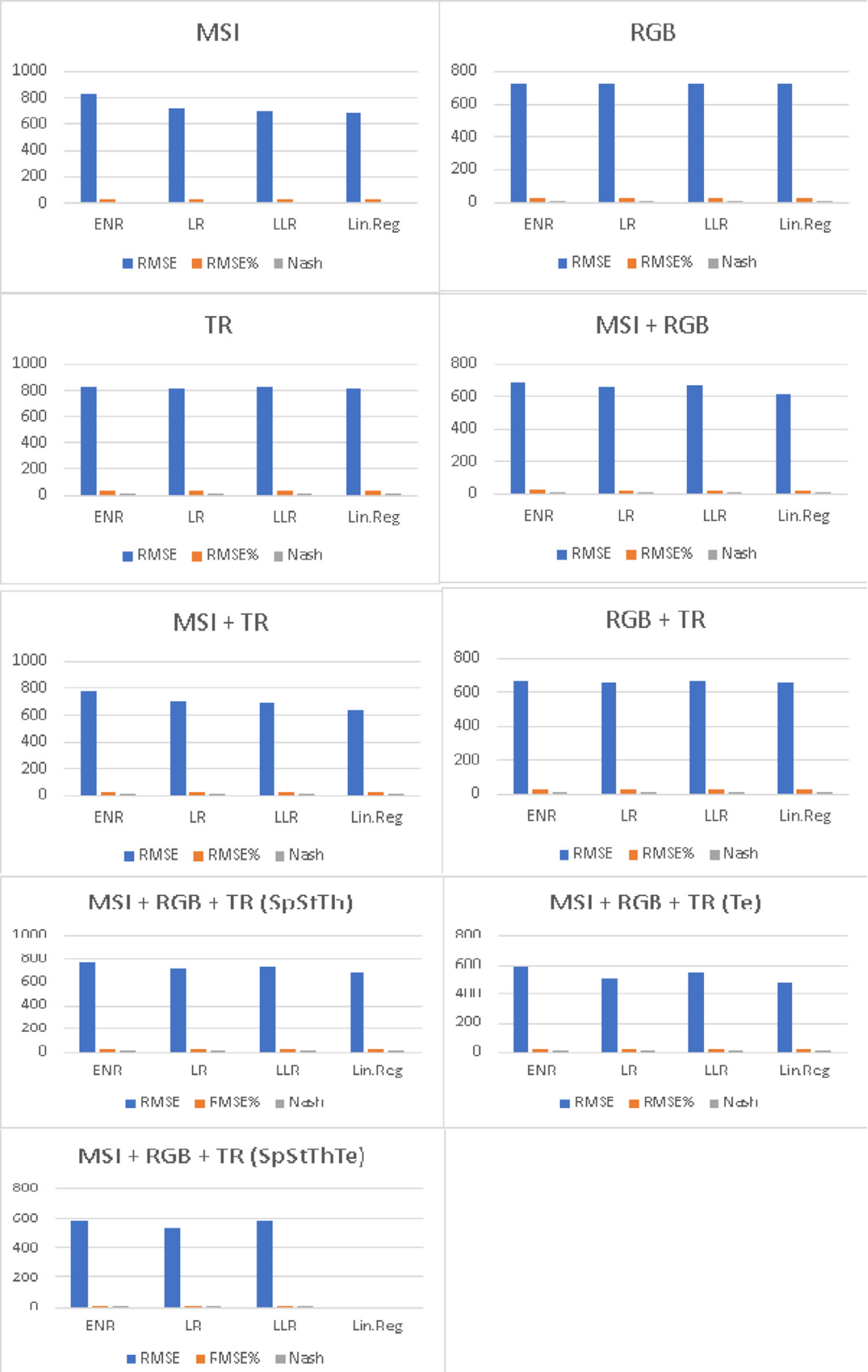
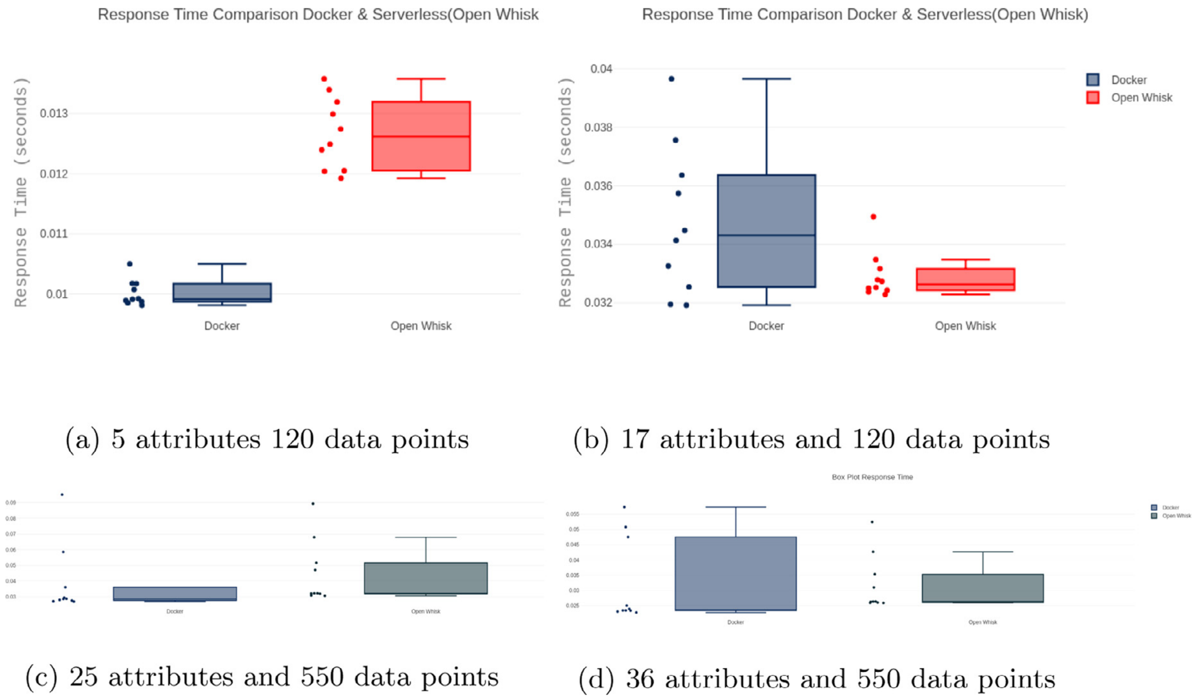


Fig. 12. Validation statistics of different models for grain yield prediction: *Sp represents spectral features, St represents structure features, Th represents thermal features, Te represents texture features, SpStTh represents spectral, structure and thermal features, SpStThTe represents spectral, structure, thermal, texture features. ENR represents Elastic net regression, LR represents lasso regression, LLR represents LassoLars regression and Lin. Reg represents Linear regression.



(a) 5 attributes 120 data points

(b) 17 attributes and 120 data points

(c) 25 attributes and 550 data points

(d) 36 attributes and 550 data points

Fig. 13. Data Processing Performance on Serverless vs Container depends on datasets size: Serverless is not always faster: The experiments was performed on 5 attributes and 120 data points (a), 25 attributes 550 data points (c) and it is seen that Docker outperformed serverless; in the second experiment was performed on 17 attributes and 120 datapoints (b), 36 attributes 550 data points (d) which makes serverless perform faster than Docker.

6. Conclusion

In this paper, we presented Polly, an online opensource tool for rapid data analysis and integration. Our study focused on integrating and predicting parameters of an agricultural dataset obtained using UAV and hyperspectral cameras on a soybean field in MO. We demonstrated our tool using regression on such datasets. We also analyzed the performance of our backend tool comparing the response time using serverless computing as well as Docker containers finding the perhaps surprising result that the overhead of Serverless computing is not justified (and so Docker containers outperforms serverless computing) when the number of attributes to process is fairly small. Hence, from our small performance analysis, a take-home message is that data size and number of attributes (e.g. in a regression) play a significant role in altering the performance of the chosen backend technology. This should be taken into account even when researchers do not use our platform. Our tool is freely available at [12].

Declaration of Competing Interest

- This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.
- The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript
- The following authors have affiliations with organizations with direct or indirect financial interest in the subject matter discussed in the manuscript:

CRediT authorship contribution statement

Waqar Muhammad: Conceptualization, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Flavio Esposito:** Conceptualization, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Maitiniyazi Maimaitijiang:** Conceptualization, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Vasit Sagan:** Conceptualization, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Enrico Bonaiuti:** Conceptualization, Formal analysis, Data curation, Writing - original draft, Writing - review & editing.

Acknowledgment

This work has been partially supported by the National Science Foundation Awards CNS-1647084, CNS-1836906, IIA-1355406 and IIA-1430427. Authors would like extend their appreciation to many researchers involved in designing UAV systems in the Remote Sensing Lab at Saint Louis University, establishing the experimental field and conducting field work that provided data for this research. The contribution by EB was supported by ICARDA and the CGIAR Research Program on Grain Legumes and Dryland Cereals (CRP-GLDC).

References

[1] Y. Kang, S. Khan, X. Ma, in: Climate change impacts on crop yield, crop water productivity and food security - a review, 2009.
[2] S. Wolfert, L. Ge, C. Verdouw, M.-J. Bogaardt, Big data in smart farming a review, Agricultural Systems 153 (2017) 69–80, doi:10.1016/j.agsy.2017.01.023.
[3] M. Maimaitijiang, A. Ghulam, P. Sidike, S. Hartling, M. Maimaitiyiming, K. Peterson, E. Shavers, J. Fishman, J. Peterson, S. Kadam, J. Burken, F. Fritschi, Unmanned Aerial System (UAS)-based phenotyping of soybean using multi-sensor data fusion and extreme learning machine, ISPRS Journal of Photogrammetry and Remote Sensing 134 (2017) 43–58, doi:10.1016/j.isprsjprs.2017.10.011.
[4] Rapidminer https://rapidminer.com/.
[5] Weka: https://www.cs.waikato.ac.nz/ml/.
[6] Spss ibm statistics software. https://www.ibm.com/analytics/spss-statistics-software.
[7] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A.C. Arpaci-Dusseau, R.H. Arpaci-Dusseau, Serverless computation with openlambda, in: 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16), USENIX Association, Denver, CO, 2016.
[8] Serverless computing https://www.serverlesscomputing.org/,
[9] Tableau: https://www.tableau.com/.
[10] Power bi by microsoft https://powerbi.microsoft.com/en-us/.
[11] G. Dwyer, S. Aggarwal, J. Stouffer, Flask: Building Python Web Services, Packt Publishing, 2017.
[12] W. Mohammad, F. Esposito, Polly: A Tool for Rapid Data Integration and Analysis in Support of Agricultural Research and Teaching. available at http://cs.slu.edu/projects/polly,
[13] IBM openwhisk https://openwhisk.apache.org/.