Earth and Space Science

RESEARCH ARTICLE
10.1029/2022EA002379

Key Points:

- We discuss the reproducibility challenges faced in research by Deep Learning approaches using Big Data
- We provide advice for pre-screening papers (before experiments) to avoid poorly invested effort
- We present a recipe with a set of mitigation strategies to address common errors users (researchers, authors, reviewers) may encounter

Correspondence to:

P. L. P. Corrêa,
pedro.correa@usp.br

Citation:

Machicao, J., Ben Abbes, A., Meneguzzi, L., Corrêa, P. L. P., Specht, A., David, R., et al. (2022). Mitigation strategies to improve reproducibility of poverty estimations from remote sensing images using Deep Learning. *Earth and Space Science*, *9*, e2022EA002379. https://doi.org/10.1029/2022EA002379

Received 12 APR 2022
Accepted 24 JUL 2022

Author Contributions:

Conceptualization: J. Machicao, P. L. P. Corrêa
Investigation: A. Ben Abbes
Methodology: J. Machicao, A. Ben Abbes
Software: J. Machicao, A. Ben Abbes
Supervision: P. L. P. Corrêa, S. Stall, N. Mouquet, M. Chaumont, L. Berti-Equille, D. Mouillot
Visualization: J. Machicao
Writing – original draft: J. Machicao, A. Ben Abbes, L. Meneguzzi, A. Specht, R. David, G. Subsol, D. Vellenich, R. Devillers, N. Mouquet

Mitigation Strategies to Improve Reproducibility of Poverty Estimations From Remote Sensing Images Using Deep Learning

J. Machicao[1], A. Ben Abbes[2], L. Meneguzzi[1], P. L. P. Corrêa[1], A. Specht[3], R. David[4], G. Subsol[5], D. Vellenich[1], R. Devillers[6], S. Stall[7], N. Mouquet[2,8], M. Chaumont[5,8], L. Berti-Equille[6], and D. Mouillot[9]

[1]Escola Politécnica da Universidade de São Paulo (EPUSP), São Paulo, Brazil, [2]FRB-CESAB, Montpellier, France, [3]Terrestrial Ecosystem Research Network, University of Queensland, Brisbane, QLD, Australia, [4]European Research Infrastructure on Highly Pathogenic Agents, AISBL, Brussel, Belgium, [5]Research-Team ICAR, LIRMM, CNRS, University of Montpellier, Montpellier, France, [6]Espace-Dev (IRD-UM-UG-UR-UA-UNC), Montpellier, France, [7]American Geophysical Union, Florida, WA, USA, [8]University of Nîmes, Nîmes, France, [9]MARBEC, Univ Montpellier, CNRS, Ifremer, IRD, Montpellier, France

Abstract The challenges of Reproducibility and Replicability (R & R) in computer science experiments have become a focus of attention in the last decade, as efforts to adhere to good research practices have increased. However, experiments using Deep Learning (DL) remain difficult to reproduce due to the complexity of the techniques used. Challenges such as estimating poverty indicators (e.g., wealth index levels) from remote sensing imagery, requiring the use of huge volumes of data across different geographic locations, would be impossible without the use of DL technology. To test the reproducibility of DL experiments, we report a review of the reproducibility of three DL experiments which analyze visual indicators from satellite and street imagery. For each experiment, we identify the challenges found in the data sets, methods and workflows used. As a result of this assessment we propose a checklist incorporating relevant FAIR principles to screen an experiment for its reproducibility. Based on the lessons learned from this study, we recommend a set of actions aimed to improve the reproducibility of such experiments and reduce the likelihood of wasted effort. We believe that the target audience is broad, from researchers seeking to reproduce an experiment, authors reporting an experiment, or reviewers seeking to assess the work of others.

Plain Language Summary This paper aims to help researchers understand the challenges of reproducing Deep Learning (DL) publications, mitigate reproducibility gaps, and make their own work more reproducible. We build on the work of others and add recommendations organized by (a) the quality of the data set (and associated metadata), (b) the DL methodology, (c) the implementation methodology, and the infrastructure used. To our knowledge, this is the first initiative of its kind to address the problem of reproducibility in remote sensing imagery and DL problems for real-world tasks. We hope this paper lowers the barrier to entry for the DL community to improve research. Following the lifecycle mantra: reproduce!, then replicate! With the goal of improving reproducibility!

1. Introduction

A gold standard in modern science is to achieve full replicability of scientific experiments or, when not possible, achieving reproducibility (Peng, 2011). This ensures open and accessible research, which accelerates the progress of the scientific community (Pineau et al., 2020a). However, in order to make a research project reproducible, a lot of work is required to document, check and make the system created useable (Yen et al., 2021). This difficulty is reflected in the fact that more than 70% of researchers across most disciplines have failed to replicate other scientists' experiments and more than 50% have failed to replicate one of their own experiments (Baker, 2016).

> We refer to the definitions of Reproducibility and Replicability (R & R) given by the National Academies of Sciences, Engineering, and Medicine (2019), as follows: "*Replicability: is obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data. Two studies may be considered to have replicated if they obtain consistent results given the level of uncertainty inherent in the system under study. Reproducibility: is obtaining consistent results using the same input data; computational steps, methods, and code; and conditions of analysis.*"

In computer science, there are also challenges related to reproducibility, insufficient specification of the versioning of the libraries or frameworks used, lack of availability of codes, execution errors, discrepancies between GPU floating point numbers, and incompatibility between alleged and presented results. In Deep Learning (DL), a subfield of Machine Learning (ML) that aims to build learning capabilities in computers, these challenges of reproducibility are not different and can be more complex (Heil et al., 2021; McDermott et al., 2021; Pineau et al., 2020a; Renard et al., 2020). There are additional complexities to handle during a DL experiment, such as the large number hyperparameters of the training process and the size of data (with possibly missing datasets and changes to the data), changes to some of the algorithms, and consequently versioning control to deal with the many iterations during training. In summary, these sources of "variability" can be due to: (a) the data set, (b) the DL architecture, (c) the optimization procedure, (d) the hyperparameters for the optimization, and (e) the implementation and infrastructure (Renard et al., 2020). These reproducibility challenges may be aggravated by the authors publishing only positive results (Pineau et al., 2020a).

Awareness of the importance of Reproducibility and Replicability (R & R) has increased rapidly in recent years. This has resulted in a number of guidelines, recommendations, checklists, and workflow tools for researchers who wish to develop a new experiment or submit a manuscript (Krafczyk et al., 2021; Pineau, 2020b; Renard et al., 2020; Walsh et al., 2020). For example, Pineau (2020b) proposed a checklist that includes recommendations for the minimum information that a manuscript should contain to ensure that an experiment can be easily reproduced (Pineau et al., 2020a). Renard et al. (2020), considering the complexity of DL models, developed a set of guidelines for structural components of DL frameworks based on the sources of "variability" that can make reproducibility difficult. The ML and DL communities such as Paperswithcode (2021) and journals such as Association for Computing Machinery (2020) are working to reward and motivate the adoption of R & R practices. Recommendations have been proposed to better organize the steps of a ML workflow, such as DOME-ML (Data, Optimization, Model and Evaluation in Machine Learning) which aims to create standards for supervised ML validation in biology (Walsh et al., 2020). There are a growing number of workflow tools such as Apache Airflow (Kotliar et al., 2019), MLFlow (Zaharia et al., 2018), Collective Knowledge technology (Fursin, 2020) and Whole Tale (Brinckman et al., 2019), the last mentioned capturing the prospective provenance of all data products generated during a study. Other resources have been proposed such as PRIMAD (Freire et al., 2016), a model for computational reproducibility, and third-party libraries such as dToolAI (Hartley & Olsson, 2020). All of these tools aim to help researchers minimize inadvertent errors and achieve a reproducible result.

Of particular interest in this paper is the need to assess poverty. This is a primary goal of the 2030 Agenda for Sustainable Development (United Nations, 2015), through the first Sustainable Development Goal (SDG 1), which aims to "eradicate extreme poverty for all people everywhere" (United Nations, 2015). Census surveys are the historic basis for estimating poverty indicators such as wealth index, income, longevity, and education (Burke et al., 2021). The use of censuses as a basis for international comparison and benchmarking is limited in several ways: (a) they are collected intermittently (e.g., every 10 years) and not synchronously across the world; (b) different protocols are followed in different regions of the world, preventing direct comparisons of the data across countries or regions; and (c) they are very costly to conduct. There is therefore an urgent need for improved methods to provide more up-to-date and relevant information to decision makers and better enable them to predict changes in socio-economic variables. This will lead to more effective policies and better outcomes for citizens of different countries.

Various methods have been proposed to estimate poverty (Ghosh et al., 2013), most recently using remote sensing imagery and DL, and these have contributed to the understanding of poverty in regions that do not have quality census data (Ayush et al., 2020; Burke et al., 2021; Diou et al., 2018; Engstrom et al., 2017; Jean et al., 2016;

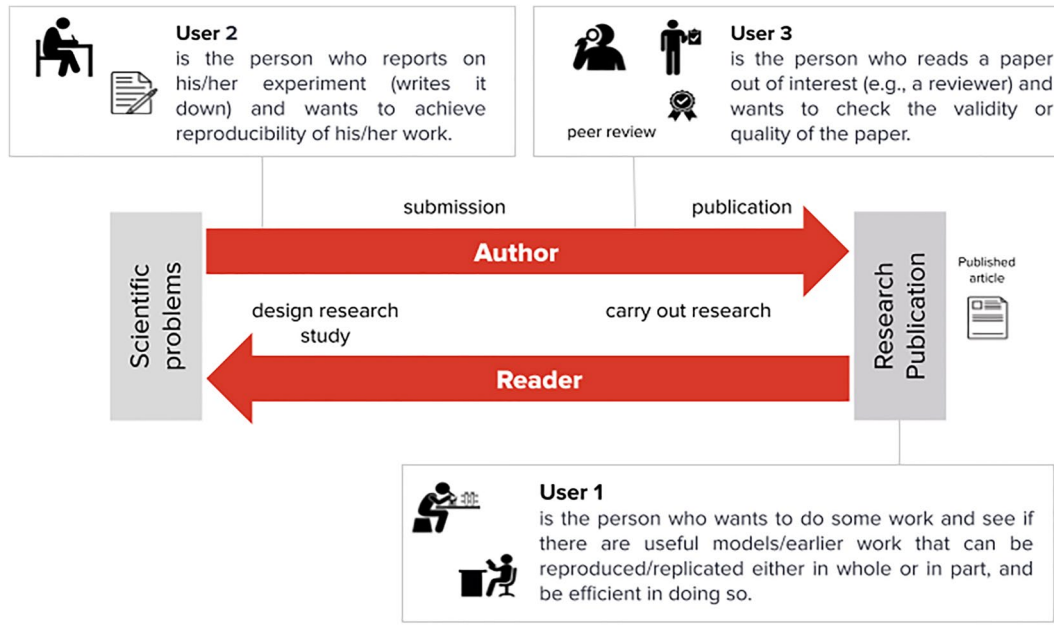

Figure 1. User profiles expected to benefit from this paper.

Machicao et al., 2022; Suel et al., 2019; Xie et al., 2016; Yeh et al., 2020). The use of multi-temporal satellite imagery involves, however, a vast amount of data, which is difficult to manage and consequently to reproduce.

Remote sensing imagery can be acquired by a variety of sensors, some of which are carried by satellites, while others are operated from aircraft or drones, or from vehicles. Remote sensing sensors are capable of capturing imagery at various spatial, spectral, radiometric and temporal resolutions (Engstrom et al., 2017) and provide information that can be analyzed for inherent patterns, shapes, and textures. DL methods can be used to enable recognition of features like vehicles, highways, agriculture landscapes, and infrastructures such as houses, buildings, and so on. These can potentially be related to the level of poverty or wealth in a target area, assuming the features extracted can accurately estimate poverty indicators. However, there are also challenges, especially when comparing between different locations or time periods. For example, a particular type of image may not be available everywhere or for sufficient time series, it may not be available due to embargoes, or it may not be useable due to atmospheric anomalies such as cloud obstruction. The main problem is to find methods to map poverty distribution that can be studied at a range of granularities, from the country, city, village, municipality level, or other level.

To test the reproducibility of complex experiments, we examined the general principles of reproducibility using three use cases of remote sensing and DL to assess poverty. We analyze these cases using published checklists or guidelines for reproducibility (Pineau et al., 2020a; Renard et al., 2020), and assess compliance with FAIR principles (Findability, Accessibility, Interoperability, Reusability; Wilkinson et al., 2016) using an approach proposed by Hartley and Olsson (2020).

This study emerged from a roundtable discussion (Correa et al., 2021) in which research pipelines (pre-, during, and post-publication). Therefore, we believe this study is useful to three types of audience (Figure 1): (a) individuals who are conducting some work and want to see if there are useful models or previous work that can be fully or partially reproduced or replicated, (b) individuals who want to report on their experiment to ensure the reproducibility of their work, and (c) individuals who are reading a paper (e.g., a peer review) and want to check the validity or quality of the work.

The paper is organized into three main parts: (a) the reproducibility challenges faced in reproducing three experiments, (b) the development of an approach to screen DL experiments before reproduction to avoid poorly invested effort, and (c) a recipe and a set of mitigation strategies ("fixes") to address common errors that the user (e.g., researchers, authors, reviewers) may encounter.

2. Materials and Methods

2.1. Introduction to the DL Workflow

In general, a DL workflow consists of the following steps: (a) data set acquisition, (b) pre-processing and cross validation strategies, (c) development of the DL experiment (including the estimation of the hyperparameters, training steps with the optimization procedure) and (d) evaluation of the DL experiment. All these steps needed to be (e) appropriately implemented within the chosen infrastructure. This is illustrated using the general flowchart in Figure 2.

These steps are expanded as follows:

1. **Data set.** Two types of data are needed to train a DL model, the census data set (the reference or "control" data to calculate the poverty indicators) and the remote sensing imagery to be trained.
2. **Pre-processing and cross validation.** At this stage the data set is prepared, that is, the data are cleaned, transformed, organized and annotated to be introduced to the DL algorithm. Then the data are divided into training, validation and test datasets following a "cross validation strategy".
3. **Deep Learning experiment**. First, the DL architecture and hyperparameters are configured, that is, the designer selects an architecture to use (by choosing a known DL model, designing a new architecture itself, or using a network architecture searcher through optimization). A typical architecture describes the number of nodes, the number of layers and their types (fully connected, foldable, pooling, dense, etc.) and the connectivity between layers. Second, the training process takes place. In this phase, some experiments can be planned to find the optimal set of parameters within the DL model. All the hyperparameters (including learning rate, batch size, etc.) must be carefully described in order to run the model.
4. **Evaluation.** At this stage, the performance of DL models is evaluated using a set of metrics according to the problem under study. There are various metrics such as precision and recall, area under the curve, sensitivity and specificity, Pearson's correlation coefficient, Kendall's tau coefficient, Cohen's kappa, and mean absolute error, to name a few. Various experiments could be run to find an optimal solution in terms of time and accuracy so that a final model can be released.
5. **Implementation and Infrastructure.** Here the full implementation and infrastructure details of the approach are determined. The designer describes the programming language, the computational structure, the computing power of the architecture, and the libraries and frameworks to be used.

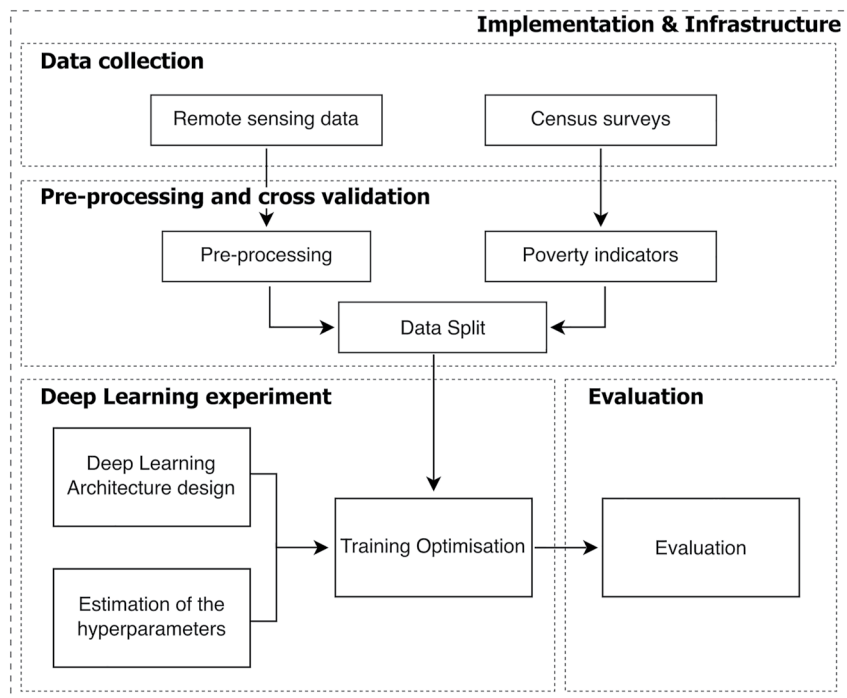

Figure 2. Flowchart of a method for poverty estimation using remote sensing data and Deep Learning (DL) approaches. The general components are in bold while the steps of the DL workflow are shown in the boxes.

2.2. Examination of Reproducibility of an Experiment

Three aspects are involved in the evaluation of the reproducibility of an experiment reported in a paper: the information provided in the paper, the specificity of the DL experiment, and the "FAIRness" of the study.

To assess the adequacy of the information provided in the paper about the experiment we used the Machine Learning checklist proposed by Pineau (2020b), noticing that the paper and the code are two separate research artifacts, each with their own checklist. Second, we assessed the steps of the DL workflow using the recommendations of Renard et al. (2020), focusing on the sources of potential difficulties ("variabilities") in each case study. Finally, we assessed the "FAIRness" of the DL code using sub-principles following the work of Hartley and Olsson (2020).

2.2.1. Pineau's Checklist

A checklist was proposed for the machine learning community by Pineau et al. (2020a, 2020b) to assist and standardize data and code descriptions for publication, covering items needed to ensure reproducibility. By this means they aimed to ensure that presented results are sound and reliable, and the method by which they were obtained is replicable. This checklist is based on the most frequent obstacles to reproducibility.

2.2.2. Renard's Variabilities and Recommendations

Renard et al. (2020) proposed recommendations for researchers while conducting a DL experiment so that before publication the authors could consider the reproducibility issues ("variabilities") which can influence and be influenced by each other and due the stochasticity conditions of their parameters, leading to difficulties for reproducibility.

The main difference between the checklist of Pineau (2020b) and the recommendations of Renard et al. (2020) is the stage at which they should be applied. Pineau's checklist should be used during the submission process of a paper to ensure they properly and completely describe the process, while Renard et al. (2020)'s recommendations should be used during the DL experiments themselves. They are complementary.

2.2.3. Hartley & Olsson FAIR Sub-Principles Selection

Hartley and Olsson (2020) have proposed specific FAIR sub-principles that were relevant to the particular challenges of DL (Wilkinson et al., 2016): Findability (F1, F2), Accessibility (A1), Reusability (R1, R1.2), and Interoperability (I3). F1 ((Meta)data are assigned a globally unique and persistent identifier) aims to ensure that training data are permanently identifiable. That is, digital objects should have unique and persistent identifiers such as PURL/ePIC, ORCID, DOI, RAid, PIC, etc. F2 (data are described with rich metadata) is required to ensure that users of the model can trace its provenance. Adherence with A1 ((Meta)data are retrievable by their identifier using a standardized communications protocol) and R1 (Metadata are richly described with a plurality of accurate and relevant attributes) will allow the software to use the metadata to train the model, and R1.2 ((Meta)data are associated with detailed provenance) requires the encoding of the hyperparameters for training, as well as the details of the data preprocessing and provenance applied. Finally, I3 ((Meta)data include qualified references to other (meta)data) is required to ensure that the DL trained model is linked to its training data and because the (meta)data contains qualified references to other (meta)data.

3. Examination of the Use Cases

We took the following approach:

1. First, we selected papers focused on DL, remote sensing imagery and poverty estimation, according to specific criteria (detailed on Section 3.1).
2. We read the papers to understand their objectives, methods and main conclusions. We made a summary of each paper.
3. We conducted a "naive" review of the papers systematically evaluating each of them against Pineau's checklist, Renard's variabilities and Hartley & Olsson's FAIR sub-principles.
4. We reproduced the workflow reported in the papers and reported any challenges in reproducibility according to the five common steps or components of the DL Workflow (Figure 2).
5. We identified the main challenges and constraints from these papers and presented them accordingly. Finally, we propose a set of mitigation strategies to overcome the main reproducibility challenges and help researchers achieve their goals.

3.1. Selection and Description of the Use Cases

Three papers and their experiments (Jean et al., 2016; Suel et al., 2019; Yeh et al., 2020) were selected due to (a) their focus on the use of DL on satellite imagery to estimate poverty, (b) the methods used (e.g., common architecture such as convolutional neural network model—CNN), (c) the fact that all of them were published in high impact journals and consequently might be assumed that they had been subject to a high standard of review, and (d) because all of them have their source code published.

3.1.1. Use Case 1 (Jean et al., 2016)

Jean et al. (2016) is one of the pioneering methods of the use of DL to predict household expenditure and asset wealth in developing countries using satellite images (daylight and nightlight). In that work, they used several villages as their smallest granularity ("clusters") from five African countries (Nigeria, Tanzania, Uganda, Malawi, and Rwanda). Due to the reduced number of labeled samples, Jean et al. (2016) used nightlight intensity as a data proxy ("preliminary task") to train DL models using transfer learning.

The method used by Jean et al. (2016) consisted of five steps and our actions to reproduce them are as follows:

1. **Data set.** In the original paper, four datasets were identified, two for the wealth index and two for the satellite imagery. To obtain a poverty indicator, they obtained household expenditure data from the World Bank's Living Standards Measurement Study (LSMS) survey and the Demographic and Health Surveys (DHS). For the satellite imagery (daytime and nightlight satellite images), they used Google Static Maps (GSM) and the National Oceanic and Atmospheric Administration's National Geophysical Data Center (NGDC, 2010; NOAA, 2014).
2. **Pre-processing and cross validation.** Jean et al. (2016) used the k-fold cross validation method. In some experiments, $k = 5$ and $k = 10$ were used separately. They used two other approaches to train the models: In-Country and Out-of-Country (OOC) where no survey training data is available. The In-country approach showed uniformly better results, but the OOC approach showed good results, often close to the In-country approach.
3. **Deep Learning experiment**. A three-step method was used by the authors. First, a pre-trained CNN, more specifically a VGG-F with 8 layers (Simonyan & Zisserman, 2015), pre-trained on a large data set called ImageNet (Russakovsky et al., 2015) was used. The main strategy in Jean et al. (2016) is to use a proxy for economic activity, nightlight satellite imagery which is known to be significantly correlated (Henderson et al., 2012; Pinkovskiy & Sala-i-Martin, 2016). By learning to infer nightlight from a daytime image, the DL pipeline transforms the input image into a specific feature vector. Finally, a ridge regression model was used to predict poverty indicators from the corresponding CNN feature vector. The CNN training procedure uses a mini batch of size 64 with gradient descent with momentum: 0.9, weight decay: 0.0005 and initial learning rate: 0.0001. The models were trained during 25 epochs according to their Github code.
4. **Evaluation**. To quantify the predictive performance of the estimation model, Jean et al. (2016) used the coefficient of determination (r-squared) and root mean squared error (RMSE). To validate the performance of the transfer learning for the nightlights, the authors conducted 100 experiments in cross validation. They also conducted an experiment in which daytime were randomly assigned to locations where surveys were conducted, and the same model was re-trained for incorrect images. That experiment was repeated 1,000 times for each of the countries and the performance of r-squared was compared with randomly shuffled images.
5. **Implementation and Infrastructure**. The main software used was Python 2.7 with caffe, GDAL (Geospatial Data Abstraction Library) and R 3.2.4.

3.1.2. Use Case 2 (Suel et al., 2019)

Suel et al. (2019) used images taken at ground level instead of satellite images. They developed a DL model using Google Street View (GSV) images to extract features that could predict poverty statistics such as income, education, unemployment, housing, living environment, health, and crime. Their experiments focused on different regions of London and other cities in England.

Similar to the previous use case, the method used by Suel et al. (2019) was also aligned with Figure 2 flowchart, consisting of the following five steps (see below):

1. **Data set**. In the original paper, Suel et al. (2019) identified five datasets, three of which relate to the wealth indices, which are government statistics for the total population at a fine scale of Lower Layer Super Output Area (LSOA) from three sources (Census https://www.ons.gov.uk/census/2011census, English Indices of

Deprivation https://www.gov.uk/government/statistics/english-indices-of-deprivation-2015, and Greater London Authority household income estimates https://data.london.gov.uk/dataset/household-income-estimates-small-areas); one data set relates to postcodes (from the Office for National Statistics and Postcode Directory for the United Kingdom), and one data set relates to street view imagery using GSV services. From government statistics, they used 12 indicators to label each image.

2. **Pre-processing and cross validation**. The Street View data set contains 4 images per postcode, each with a different direction, providing a 360° view of each measured point. For each indicator, they normalized these values and deciles determined, with decile 1 corresponding to the worst-off 10% of LSOAs, and decile 10 corresponding to the best 10% in London. They used a k-fold cross validation algorithm ($k = 5$), with 4 of these folds used for training and the remainder for testing.
3. **Deep Learning experiment**. The training process was based on a transfer learning process, the authors used a pre-trained VGG-16 network model (Simonyan & Zisserman, 2015), which they used as a feature extractor. From this, a 4096-D output is generated for each image, with 4 images per location (4 different viewpoints of the same location). This 4096-D vector is applied to a custom network that processes each of the four images individually on three different layers, and then aggregates the images from the average. And finally, in the output layer, the sigmoid activation function is used to generate a probability for each of the deciles.

In order to train the neural network, the cost of the cross entropy function was optimized. An Adam optimizer with a learning rate of 5e−6 and with an execution of 100,000 training iterations was used. The prediction is computed from the continuous average for each LSOA (multiple postal codes) before applying the activation function (sigmoid in this case). Finally, they converted to a decile and compared to the original decile using various metrics.

4. **Evaluation**. For the evaluation of the model, the Pearson's correlation coefficient, Kendall's tau coefficient, Cohen's kappa, and mean absolute error (MAE), Adjacent Accuracy (with adjacency interval in {±0 (regular accuracy), ±1, ±2}) were used.
5. **Implementation and Infrastructure**. Suel et al. (2019) used TensorFlow in Python as main software, although no version information was provided.

3.1.3. Use Case 3 (Yeh et al., 2020)

Yeh et al. (2020) used a novel method to understand economic well-being in 19,669 African villages across 23 countries in Africa. They estimate well-being indicators from satellite imagery using DL techniques. An interesting point of this work is the use of a very large amount of data (surveys, images) over different time periods.

In this use case, we only look at reproducing the part of their work that uses DHS surveys. Forty-three surveys from nationally representative DHSs conducted between the years 2009 and 2016 in 23 African countries were used.

1. **Data set**. In the original paper, four datasets were used, two of which were related to the wealth index (DHS and LSMS) and two of which related to the associated satellite daytime imagery (Landsat and nightlight images). The wealth index was constructed from the first principal component of the DHS responses using principal component analysis (PCA). It included the number of rooms occupied in a house, whether the house has electricity, the quality of flooring in the house, water supply and whether the house has a toilet, ownership of a telephone, radio, television, cars, and motorcycles. Based on the geolocalization and the date of each village, satellite images were automatically downloaded from Google Earth Engine (GEE). In this study, Yeh et al. (2020) used Landsat and nightlight images which centered on each village. A composite of three years was chosen for the studied period 2009–2016 (2009–2011, 2012–2014, and 2015–2017).
2. **Pre-processing and cross validation**. To cross-validate, Yeh et al. (2020) splitted the data into five folds. The aim was to train each model on 3-folds, validate on a fourth, and test on a fifth. To avoid the overlap of satellite images of the villages, they used two configurations: Out-Of-Country and In-Country. For the OOC split, they assigned entire countries to a split. For the In-Country splits, they allowed different clusters within a country to be assigned to different splits. They used both Leave-one-group-out and Leave-one-fold-out Cross-Validation strategies. The deep models were trained on random subsets of 5%, 10%, 25%, 50%, and 100% of the total training data and repeated over 3 trials with different random subsets.
3. **Deep Learning experiment**. Two CNNs using Resnet-18 architecture (He et al., 2016) were independently trained on the Landsat and nightlight images, and the models developed were fused in their final fully connected layer. A pre-trained CNN model based on Resnet-18 architecture was used. The first convolutional

layer was modified to take into account the multi-band of Landsat images, and the final layer to output a scalar for regression. They used a learning rate of a range of values: 1e−2, 1e−3, 1e−4, 1e−5.

4. **Evaluation**. The CNN models were trained with the Adam optimizer and a RMSE loss function. A batch size of 64 and the learning rate was decayed by a factor of 0.96 after each epoch. The models were trained for 150 epochs for In-Country and for 200 epochs for OOC.
5. **Implementation and Infrastructure**. Python 3.7 with TensorFlow r1.15, and R 3.6 were used.

3.2. Examination of the Reproducibility of the Use Cases

A comparison of these three use cases using the three assessment in shown in Table 1: (a) the checklist on the ideal components of a Machine Learning paper by Pineau (2020b) (Section 2.2.1), (b) the recommendations for reporting a DL experiment by Renard et al. (2020) (Section 2.2.2), and (c) the FAIR sub-principles relevant to DL by Hartley and Olsson (2020) (Section 2.2.3). This assessment table helps identify gaps in the reproducibility of a paper (User 1), serves as a checklist for someone writing up their work (User 2) or assists the person reviewing the work of others (User 3).

The criteria are formulated as questions and divided into the following categories: C1 "the description of the data set", C2 "FAIR sub-principles", C3 "the description of the DL architecture and hyper-parameter optimization process", C4 "the infrastructure and implementation", C5 "reported experimental results and theoretical claim", and C6 "the shared code".

From this table, we can identify some shortcomings by calculating the proportion of "Partial" or "No" responses relative to the total number of surveys for each criterion. Ordered from highest to lowest, C2, C3, C1, and C5 are the criteria with the most issues. For example, for all items in criterion C1, there are 4 "Partially" and 3 "No" determinations, from a total of 18 surveys, corresponding to almost 39% of inadequacy in data set description. There was a 83% lack of compliance with FAIR criteria (C2), while 45% of the criteria were "No" or only "Partially" filled for the description of the architecture and algorithms (C3). 2 "No" (33%) for C4, 2 "Partially" and 3 "No" (33%) for C5, and 1 "Partially" and 5 "No" (33%) for C6. Most "No" responses were for FAIR compliance. It is important to note that the lack of compliance of the metadata to FAIR principles is not only for the data used, but also for the data generated from the DL models.

Using an examination of the DL workflows of the three use cases (Sections 3.1.1–3.1.3) and the analysis shown in Table 1, it is clear that there are common challenges in the achievement of effective reproducibility. The most important of these relate to three steps given on the DL flowchart (Figure 2), (a) the quality of the data set (and the metadata associated with it), (b) the DL architecture and hyper-parameter optimization, and (c) the implementation and the infrastructure used.

3.2.1. The Quality of the Data Set

A major limitation of the Jean et al. (2016) work, as shown in Table 1 criterion C1.1, was that Google Static Maps imagery is constructed as a mosaic of aerial photographs and it loses the period of composition, so basically it was not possible to obtain the same images. In addition, due to policy restrictions, there is no persistent identifier (criterion C2.1 from Table 1) to the raw data set in their repository (https://github.com/nealjean/predicting-poverty/).

Similarly, the first challenge in reproducing the experiments from Suel et al. (2019), as shown in Table 1 criterion C1.5, is that it is currently not possible to guarantee the availability of the same images through Google Street View, as it is the policy of the service to provide only the most recent image. This problem was also reported in the work of Diou et al. (2018). As a result, there is no persistent link to their data set. The README file of the source code repository (https://github.com/esrasuel/measuring-inequalities-sview) lacks some information, such as only some cities cited in the experiment are detailed, making it difficult to compare the results, and the "headers" (entities) in the wealth indices data set are not compatible with the "headers" found in the raw data (linked to the government sites).

The main challenge in the work of Yeh et al. (2020) is that the original data (DHS surveys) are not made available (criterion C1.4 from Table 1). In addition, the description of some parameters (e.g., chunk size, the number of images by file) is inadequate (criteria C1.3 from Table 1).

Table 1
Systematic Description of the Three Use Cases Based on Pineau's Checklist, (2020b) [a], the Recommendations From Renard et al., 2020 [b], and the FAIR Sub-Principles (Following Hartley & Olsson, 2020) [c]

Criteria description	Use cases		
	Jean et al. (2016)	Suel et al. (2019)	Yeh et al. (2020)
C1 The description of the data set			
1.1 Have you given the relevant statistics, such as the number of remote sensing images? [b]	Partially	Yes	Yes
1.2 Is the data set (e.g. remote sensing data, census surveys) open access? [b]	Public	Public	Public
1.3 Are there clear details of training/validation/test splits? [a]	Partially	Yes	Yes
1.4 Is there an explanation of any data that was excluded, and all images pre-processing steps? [a]	Yes	Yes	No
1.5 Is there a link to a downloadable version of the data set or simulation environment? [a]	Yes	Partially	Yes
1.6 For new data collected, is there a complete description of the data collection process, such as instructions to annotators and methods for quality control? [a]	Partially	No	No
C2 FAIR sub-principles (Findability, Accessibility, Interoperability, Reusability)			
2.1 Are (Meta)data assigned with a globally unique and persistent identifier? (F1) [c]	No	No	Yes
2.2 Are data described with rich metadata? (F2) [c]	Partially	Partially	No
2.3 Are (Meta)data retrievable by their identifier using a standardized communications protocol? (A1) [c]	Partially	Yes	No
2.4 Do (Meta)data include qualified references to other (meta)data? (I3) [c]	No	Yes	No
2.5 Are Metadata richly described with a plurality of accurate and relevant attributes? (R1) [c]	No	No	No
C2.6 Are (Meta)data associated with detailed provenance? (R1.2) [c]	Partially	Partially	No
C3 The description of the DL architecture and hyper-parameter optimization process			
3.1 Is there a clear description of the mathematical setting, algorithm, and/or model? Which DL architecture (and type of measure) was used? [a, b]	Yes, CNN	Yes, CNN	Partially, pre-trained CNN
3.2 Does the paper use a Cross-Validation strategy? [b]	Yes	Yes	Yes
3.3 Is there a clear explanation of assumptions? [a]	Yes	Yes	No
3.4 Is there an analysis of the complexity (time, space, sample size) of any algorithm? [a]	No	No	Yes
3.5 If an optimization procedure was used, is it completely detailed? [b]	Partially	Yes, Adam Optimizer	Yes, Adam optimizer
3.6 Were the Hyper-Parameters chosen manually? [b]	No	No	Yes
3.7 Does the paper clearly mention the Learning rate? [b]	No	Yes	Yes
3.8 Does the paper clearly mention the Batch size? [b]	Partially	Partially	Yes
3.9 Does the paper use Dropout regularization? (which value?) [b]	Partially	No	No
3.10 Are there the range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results? [a]	Partially	Partially	Yes
3.11 Is there an exact number of training and evaluation runs? [a]	Partially	Yes	Yes
C4 The infrastructure and implementation			
4.1 Does the paper detail the infrastructure adequately? [a,b]	No	No	Yes
4.2 Which framework was used? [b]	R and Python Caffe (according to README file)	TensorFlow/PyTorch	TensorFlow

Table 1
Continued

Criteria description	Use cases		
	Jean et al. (2016)	Suel et al. (2019)	Yeh et al. (2020)
C5 Reported experimental results and theoretical claim			
5.1 Is there a clear definition of the specific measure or statistics used to report results? [a]	Yes	Yes	Yes
5.2 Is there a description of results with central tendency (e.g. mean) & variation (e.g. error bars)? [a]	Yes	Yes	Yes
5.3 Is there a description of the average runtime for each result, or estimated energy cost? [a]	No	Partially	Yes
5.4 Is there a clear statement of the claim? [a]	Yes	Yes	No
5.5 Is there a complete proof of the claim? [a]	Partially	Yes	No
C6 The shared code			
6.1 Is the shared code Open source? [a,b]	Yes	Yes	Yes
6.2 Is there a specification of dependencies? [a]	Yes	No	No
6.3 Is there a training code? [a]	Partially	Yes	Yes
6.4 Is there an evaluation code? [a]	Yes	Yes	No
6.5 Is there a (Pre-)trained model(s)? [a]	Yes	No	Yes
6.6 Is there a README file that includes a table of results accompanied by a precise command to run to produce those results? [a]	Yes	Yes	No

Note. The first column lists the criteria used to assess each use case. The letters a, b, c after each criterion correspond to the papers from which the criteria were compiled. The answers "Partially" and "No" indicate that these are reproducible aspects that require attention.

3.2.2. The Description of the DL Architecture and Hyper-Parameter Optimization

There was incomplete information about the full set of hyperparameters in the manuscript of Jean et al. (2016) or in their supplementary material (criteria C3 from Table 1). The code implementing the training of the models is missing, and only the final trained model is provided as a checkpoint. However, the model is downloadable by a non-persistent link which is not a FAIR procedure.

In Yeh et al. (2020), as shown in criteria C3 from Table 1, a full description of the DL experiment was missing and their official GitHub repository (https://github.com/sustainlab-group/africa_poverty) was not as updated as an alternative GitHub (https://github.com/chrisyeh96/africa_poverty_clean). In addition, the workflow steps are poorly described and the parameter settings were missing.

3.2.3. Implementation and Infrastructure

The infrastructure information was not described in the Jean et al. (2016) article, as shown in criterion C4.1 from Table 1, but R and Python Caffe scripts were included in the README file in the GitHub repository (https://github.com/nealjean/predicting-poverty). The main challenge was to reproduce the training procedures presented in their repository. However, the authors link to a third repository (https://github.com/jmather625/predicting-poverty-replication) that shows an alternative version of their work using PyTorch instead, which better reproduces the training pipeline but does not achieve exactly the same performance results.

Suel et al. (2019) referred to using Python and TensorFlow in their paper (criterion C4.1 from Table 1), but the README file of their GitHub repository (https://github.com/esrasuel/measuring-inequalities-sview) was missing this information.

4. Mitigation Strategies and FAIR Advice (for User 1 & 2)

We suggest a set of mitigation strategies for improving the reproducibility of the data and code associated with DL experiments and provide advice on satisfying FAIR criteria.

In this section, we have organized suggestions for the reader according to the general DL workflow (Figure 1) and the criteria listed in Table 1, specifically C1, C3, C4, and C5. We found correspondence between "Quality of the data set" (Section 4.1) with criteria C1, DL experiment (Section 4.2) with criteria C3 and C5; "Implementation and Infrastructure" (Section 4.3) with criteria C4, and "Advice about FAIR principles criteria" (Section 4.4) with criteria C2.

4.1. Quality of the Data Set (Particularly Relevant for User 1)

To ensure efficient reproducible or replicable research, it is advisable to use optimal data quality, a comprehensive review of data quality dimensions can be found at Sidi et al. (2012). For User 2, we advise that you check that you have reported your experiments well, as detailed in Table 1 criteria C1. For User 1, it is advisable to:

- **Validate the data set used in the study**. Make some preliminary statistical analyses to validate the quality of the data set and the sample size required. Random sampling can be used to check if your data corresponds to the same samples covered in the original article. If you have missing values, create a new value by using missing values methods (more information can be found in Enders, 2010).
- **Use a sample of the data set**. For a first experiment, one could prepare the same procedure presented from the original article, and test using a sample of the data set.
- **Check parameters of the datasets or code.** Check the experiment parameters and those in the data source used. For example, check the parameters for satellite images acquisition (e.g., sensor product, start and end date, spectral resolution, spatial resolution, etc.), the variables and data collections used for census surveys. Also check for the temporary images that were downloaded and were excluded on the original paper. When working with an API (Application Programming Interface), such as GEE or GSM, internet latency or delay may be encountered which can create request errors and some images may be lost even with many trials. At this point, the task of reproduction evolves to a replication task.
- **Check the preprocessing steps**. If the preprocessing steps are not clearly defined in the original paper, for example, with respect to the satellite images, one may need to compile the images acquired across various years and then average them. There may also be a temporal mis-match between sources that needs to be reconciled, and other options may need to be pursued. Census surveys correspond to one specific year and will rarely match precisely with the imagery date.
- **Verify the data construction method**. When it is not possible to obtain the original data (e.g., DHS) because of data agreement policies or other reasons, check for the availability of data which looks similar and then to build some statistical methods, such as averaging or interpolation, to effectively check similarity. It is then possible to continue with the construction of this new data.
- **Test different configurations of data split**. This exercise is important to ensure unbiased solutions. We recommend using different strategies depending on the size of the data set. For the data splitting one can test several configurations. For instance, split a data set for training and testing data set using proportions of 95% and 5% for validation, or 90% and 10% when the data is high volume, as is commonly used in the DL community. A comprehensive review of data splitting can be found at Xu and Goodacre (2018). Other data split strategies can be used for different kinds of experiments when needed.

4.2. Deep Learning Experiment

A clear description of the DL experiment is commonly omitted. For example, Renard et al. (2020) found that fewer than 10% of the articles reviewed sufficiently described the hyperparameters and the data set to enable the work to be reproduced. The complexity of many DL experiments and repeated iterations resulting in many versions of code makes vigilance essential for accurate reporting. We advise that User 2 checks that their experiments are reported according to criteria C3 and C5 from Table 1. For User 1, here are some recommendations to overcome issues when reproducing of replicating DL experiments:

- **Look for workflows**. For User 1, we suggest careful review of the workflow used to conduct the experiments, meanwhile for User 2, we advise using a workflow to record the process, etc. Thus, check it for details of the DL architecture, procedures, and all possible information useful to reproduce the paper.
- **Look for model architecture as source code**. If there is access to the already trained model (e.g., "checkpoint", "h5 file") it might be possible to obtain the hyperparameters that were used in the training process or

even the details of the DL architecture (layers, size of inputs and/or data augmentation information, size of outputs, drop out, loss function, type of optimizer, weights initialization, learning rate, etc.). We also suggest checking whether the original input data used to train the model is published with it.

- **Look for different setups of experiments**. We suggest carefully organizing various reproducibility experiments according to the different setup found on the original paper, because they may vary the hyperparameters. It is also useful to reproduce the lead experiment so that it is possible to verify the correspondence reproducibility results.

4.3. Implementation and Infrastructure

Technology is continuously developing and updates occur frequently. The infrastructure needed for DL (specialized computer hardware, programming languages, frameworks, libraries, and so on) may have changed or been replaced by newer versions. For User 2, we advise that you check that you have properly reported your experiments as described in Table 1 criteria C4. For User 1, we provide some recommendations to account for changes in infrastructure and implementation components in DL experiments.

- **Internet flaws**. When dealing with APIs, expect some internet flaws, for instance GSM or GEE API may lose some images while downloading, but we have the possibility to know which package failed and we can retrieve the downloading process, which should be noticed that there may be some additional costs. We suggest programming a batch list advertising the already downloaded files.
- **Bugs**. It is expected that some bugs will be found in the scripts. Perhaps some source codes were provided that do not match the version mentioned in the manuscript. We suggest looking at the import libraries and checking on blogs such as Stackoverflow (https://stackoverflow.com/) to find the best adjustments.
- **Versioning**. Use the same versions as the original in order to avoid "deprecated" versions. For example, many researchers migrate TensorFlow code from TensorFlow 1.x to TensorFlow 2. Such differences can prevent successful reproduction, and it can be laborious to convert code.
- **Source code is available but there are many branches**. For instance when we have two source code hosted links (e.g., as described in Section 3.2.2), we suggest trying to merge them and compare each step. If the package libraries are missing, set up a computer with empty libraries and then install packages (after aligning them).
- **Programming language**. If the source code is in a particular programming language with which you are not expert, use another trustable version. For instance, if you wish to use Python instead of R, as used in Jean et al. (2016), there is a Pytorch implementation https://github.com/jmather625/predicting-poverty-replication.

4.4. Advice About FAIR Principles (*Particularly Relevant for User* 2)

The discussion in this section is based on the profiles of User 2 (Figure 1) and how they could employ FAIR principles when engaging in DL experiments, which is linked to criteria C2 from Table 1. The following points are called "advice" because they are the minimum criteria that a DL-based project must have.

- **Some FAIR principles are mandatory in data set design**. If the data are not compliant, they cannot be re-used. DL studies typically use large numbers of comparable samples to train and then test models. DL experiments specifically designed to analyze very large datasets are usually based on originally structured data (e.g., often large sets of images) and are associated with a specific set of attributes defined by the producer(s) of that data (human or machine). It is important to ensure your (meta)data are assigned with a globally unique and persistent identifier (FAIR principle F1), that your (meta)data are retrievable by their identifier using a standardized communications protocol (FAIR principle A1) and that your (meta)data are released with a clear and accessible data usage license (FAIR Principle R1.1).
- **Some FAIR principles need scientific community agreement and information about data set quality**. FAIR principle F2 states "Data described with rich metadata", which means that for good reproducibility of the study, at least the metadata (and ideally the data) will be sufficient, efficient, and appropriate. These qualities do not only depend on the producer: sufficient and appropriate metadata depend largely on the methods and tools that a user will use to reproduce the results of the model. In most cases, the processes of (meta)data enrichment and curation depend on the vocabularies used by the scientific community that wants to reproduce the model (I2), and are part of the DL process (see e.g., "Pre-processing and Data split" in Figure 2). This

1	**Pre-read the paper**	Pre-read the paper of interest to better understand the objectives, methods and main conclusions.
2	**Make a 'naive' review**	Make a 'naive' review of the papers by using column 1 from Table 1 as a template, and then analyze the major issues found.
3	**Fix things before running**	Fix things before running. Do search for those reproducibility issues found on Step 2.
4	**Reproduce the workflow**	Reproduce the workflow reported in the paper of interest, while using the proposed mitigation strategies.

Figure 3. Recipe for successful reproduction.

means that FAIR criteria that take into account a decision of the scientific community (with higher variability if the team's view changes), which is especially reinforced at DL reproducibility. It is important to ensure that your (meta)data includes qualified references to other (meta)data (I3), that your metadata are richly described with a plurality of accurate and relevant attributes (R1) and that your (meta)data are associated with detailed provenance (R1.2).

- **Accept uncertainty when achieving FAIR, because a data set is never perfect**. It is almost impossible to achieve a perfect FAIR because of inherent variability in its quality (it could be incomplete or biased). This is not only due to the variability of the community's judgment, but also due to the complexity of the data and the impossibility to fill the data completely and homogeneously (e.g., changing sensors, process development, artifacts, operating systems, etc.). These matters will always introduce uncertainties in the outcome of the model between datasets, but also with the same data set but a different user/community user. We advise that User 2 checks that their experiments are reported according to FAIR principles I3 and R1.2.

Based on these arguments, we can easily imagine that increasing "FAIRness" in the DL study necessarily requires an iterative process to consolidate the results, as suggested for all other scientific studies and data (David et al., 2020).

4.5. Recipe for Successful Reproduction of a DL Experiment

Before evaluating a paper reporting a DL experiment, or creating your own project, we propose a four step process: (a) pre-read the paper to identify the context and broad content. If this looks promising, (b) make a naïve review using the template we provide in this paper, and if the method continues to look promising, (c) fix any flaws and gaps identified, and (d) reproduce the workflow (Figure 3).

5. Discussion and Conclusion

The requirement for reproducibility and replicability of experiments, let alone those in which new techniques such as DL are employed, is very recent. There are few instances where R & R has been practically tested, and more work is needed to develop best practices and make this a real possibility for the achievement of truly open, defensible science. We have presented an approach that researchers can use to ensure their project is indeed FAIR, that readers can use to check the robustness of a reported method, and that those wishing to reproduce a method can test whether there is sufficient information available to do so.

To identify common problems with reproducing reported DL experiments, we analyzed the reproducibility of three case studies which used DL and remote sensing data to estimate poverty. To do this we amalgamated three published approaches, the checklist organized by the ideal structure of a paper during submission (Pineau, 2020b),

the DL workflow recommended during experimentation (Renard et al., 2020), and compliance to the FAIR sub-principles for DL (Hartley & Olsson, 2020).

Although the three use cases were proposed for a specific task (poverty estimation), we believe that the evaluation methods could be applied to more general DL tasks, where difficulties might include (a) a lack of data set specificity (and the metadata related with it), (b) inadequate description of the DL experiment, (c) a lack of details of the implementation and the infrastructure used. We also feel that these recommendations can be extended to other domains.

To publish a reproducible DL experiment, we recommend that creators (User 1) provide a downloadable version of the data set and simulation environment used, the data management procedure, data and code provenance, and data set versioning. In addition, it is important to try to overcome the challenge of effectively sharing and exchanging learning models, as ML and DL models have inherent characteristics of data and software that could also be overcome with the emergence of new services such as DLHub.org and OpenML.org (Katz et al., 2020).

One might think that the established FAIR principles are an appropriate guide to ensure effective reuse of data and software (Wilkinson et al., 2016). In practice, however, what is true for data is not directly true for all other digital objects (Katz et al., 2020). For example, the principle of interoperability may not have the same meaning when dealing with data or software (Lamprecht et al., 2020). Existing FAIR recommendations do not provide specific guidance for detailed domain-specific implementation, especially for the case of DL (Hartley & Olsson, 2020). Suggestions for the design of FAIR computational workflows have recently been made by Goble et al. (2020), who propose some specific guidelines for the creation and use of DL models in science, explaining how they relate to the FAIR principles.

There have been some recent efforts to establish standards for FAIR ML and DL models (Pineau, 2020b; Walsh et al., 2020). Although learning models typically use highly structured data that respect FAIR principles by design, they face greater variability in terms of their reproducibility and, to a lesser extent, their replicability. In addition to the inherent "variabilities" of DL (Renard et al., 2020) that pose a challenge to R & R, there are other aspects of variability that should be better discussed in future work, such as the fact that scientific (meta)data are not only related to the difficulty of reaching consensus on vocabularies, but also to the fact that the variability of the description of these very large datasets is closely related, for example, to their temporal and spatial scale or to the technical variability of their acquisition.

To our knowledge, this is a first-of-its-kind initiative that presents a problem of reproducibility in remote sensing imagery and DL problems for real-world tasks. However, there are still some limitations to this work. It tries to be general, but it is an initiative for the DL community. Therefore, we encourage other researchers to propose further mitigations to make it more general and comprehensive. We hope this paper lowers the barrier to entry for the DL community to improve research. Following the life cycle mantra: reproduce!, then replicate! With the goal of improving reproducibility!

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Data Availability Statement

The source code availability used for reproducibility can be found as follows:

- Jean et al. (2016), https://github.com/nealjean/predicting-poverty/
- Suel et al. (2019), https://github.com/esrasuel/measuring-inequalities-sview
- Yeh et al. (2020), https://github.com/sustainlab-group/africa_poverty and https://github.com/chrisyeh96/africa_poverty_clean.

Acknowledgments
This project was conducted as part of the Belmont Forum PARSEC project, funded under the Collaborative Research Action (CRA) on Science-Driven e-Infrastructures Innovation (SEI), with funding from FAPESP, the ANR, JST and the NSF, with collaborators from Australia, and support from the synthesis centre CESAB of the French Foundation for Research on Biodiversity. In Brazil the PARSEC project is also supported by the Grant 2018/24017-3, São Paulo Research Foundation (FAPESP). J.M. is grateful for the support from FAPESP (Grant 2020/03514-9). R. David was supported by the EOSC-Life European program (grant agreement No. 824087).

References

Association for Computing Machinery. (2020). Artifact review and Badging—Current. Artifact review and Badging version 1.1. Retrieved from https://www.acm.org/publications/policies/artifact-review-and-badging-current

Ayush, K., Uzkent, B., Burke, M., Lobell, D., & Ermon, S. (2020). Generating interpretable poverty maps using object detection in satellite images. In *Proceedings of the twenty-nineth international joint conference on artificial intelligence*. https://doi.org/10.24963/ijcai.2020/608

Baker, M. (2016). 1, 500 scientists lift the lid on reproducibility. *Nature*, *533*(7604), 452–454. https://doi.org/10.1038/533452a

Brinckman, A., Chard, K., Gaffney, N., Hategan, M., Jones, M. B., Kowalik, K., et al. (2019). Computing environments for reproducibility: Capturing the "Whole Tale". *Future Generation Computer Systems*, *94*, 854–867. https://doi.org/10.1016/j.future.2017.12.029

Burke, M., Driscoll, A., Lobell, D. B., & Ermon, S. (2021). Using satellite imagery to understand and promote sustainable development. *Science*, *371*(6535). https://doi.org/10.1126/science.abe8628

Corrêa, P. P., Machicao, J., Stall, S., David, R., Ben Abbes, A., Berti-Equille, L., et al. (2021). Round table Brazil—PARSEC Project—workflow management and reproducibility. https://doi.org/10.5281/zenodo.4637090

David, R., Mabile, L., Specht, A., Stryeck, S., Thomsen, M., Yahia, M., et al. (2020). FAIRness literacy: The Achilles' heel of applying FAIR principles. *Data Science Journal*, *19*(1), 32. https://doi.org/10.5334/dsj-2020-032

Diou, C., Lelekas, P., & Delopoulos, A. (2018). Image-based surrogates of socio-economic status in urban neighborhoods using deep multiple instance learning. *Journal of Imaging*, *4*(11), 125. https://doi.org/10.3390/jimaging4110125

Enders, C. K. (2010). *Applied missing data analysis: Methodology in the social sciences*. Guilford Press.

Engstrom, R., Hersh, J., & Newhouse, D. (2017). *Poverty from space: Using high-resolution satellite imagery for estimating economic well-being*. World Bank. https://doi.org/10.1596/1813-9450-8284

Freire, J., Fuhr, N., & Rauber, A. (2016). Reproducibility of data-oriented experiments in e-Science (Dagstuhl Seminar 16041). Schloss Dagstuhl - Leibniz-Zentrum Fuer Informatik GmbH, Wadern/Saarbruecken, Germany. https://doi.org/10.4230/DAGREP.6.1.108

Fursin, G. (2020). The collective knowledge project: Making ML models more portable and reproducible with open APIs, reusable best practices and MLOps. Retrieved from http://arxiv.org/abs/2006.07161

Ghosh, T., Anderson, S., Elvidge, C., & Sutton, P. (2013). Using nighttime satellite imagery as a proxy measure of human well-being. *Sustainability*, *5*(12), 4988–5019. https://doi.org/10.3390/su5124988

Goble, C., Cohen-Boulakia, S., Soiland-Reyes, S., Garijo, D., Gil, Y., Crusoe, M. R., et al. (2020). FAIR computational workflows. *Data Intelligence*, *2*(1–2), 108–121. https://doi.org/10.1162/dint_a_00033

Hartley, M., & Olsson, T. S. G. (2020). dtoolAI: Reproducibility for Deep Learning. *Patterns*, *1*(5), 100073. https://doi.org/10.1016/j.patter.2020.100073

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *Computer vision—ECCV 2016* (pp. 630–645). Springer International Publishing. https://doi.org/10.1007/978-3-319-46493-0_38

Heil, B. J., Hoffman, M. M., Markowetz, F., Lee, S.-I., Greene, C. S., & Hicks, S. C. (2021). Reproducibility standards for machine learning in the life sciences. *Nature Methods*, *18*(10), 1132–1135. https://doi.org/10.1038/s41592-021-01256-7

Henderson, J. V., Storeygard, A., & Weil, D. N. (2012). Measuring economic growth from outer Space. *The American Economic Review*, *102*(2), 994–1028. https://doi.org/10.1257/aer.102.2.994

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, *353*(6301), 790–794. https://doi.org/10.1126/science.aaf7894

Katz, D. S., Pollard, T., Psomopoulos, F., Huerta, E., Erdmann, C., & Blaiszik, B. (2020). FAIR principles for Machine Learning models. *Zenodo*. https://doi.org/10.5281/zenodo.4271996

Kotliar, M., Kartashov, A. V., & Barski, A. (2019). CWL-airflow: A lightweight pipeline manager supporting common workflow language. *GigaScience*, *8*(7), 1–8. https://doi.org/10.1093/gigascience/giz084

Krafczyk, M. S., Shi, A., Bhaskar, A., Marinov, D., & Stodden, V. (2021). Learning from reproducing computational results: Introducing three principles and the reproduction package. *Philosophical Transactions of the Royal Society A*, *379*(2197). https://doi.org/10.1098/rsta.2020.0069

Lamprecht, A.-L., Garcia, L., Kuzak, M., Martinez, C., Arcila, R., Martin Del Pico, E., et al. (2020). Towards FAIR principles for research software [JB]. *Data Science*, *3*(1), 37–59. https://doi.org/10.3233/DS-190026

Machicao, J., Specht, A., Vellenich, D., Meneguzzi, L., David, R., Stall, S., et al. (2022). A deep-learning method for the prediction of socio-economic indicators from street-view imagery using a case study from Brazil. *Data Science Journal*, *21*(1). https://doi.org/10.5334/dsj-2022-006

McDermott, M. B. A., Wang, S., Marinsek, N., Ranganath, R., Foschini, L., & Ghassemi, M. (2021). Reproducibility in machine learning for health research: Still a ways to go. *Science Translational Medicine*, *13*(586). https://doi.org/10.1126/scitranslmed.abb1655

National Academies of Sciences, Engineering, and Medicine. (2019). *Reproducibility and replicability in science*. The National Academies Press. https://doi.org/10.17226/25303

National Geophysical Data Center. (2010). Version 4 DMSP-OLS Nighttime Lights Time Series.

NOAA National Geophysical Data Center. (2014). F18 2013 nighttime lights composite.

Paperswithcode. (2021). ML reproducibility challenge. Retrieved from https://paperswithcode.com/rc2021

Peng, R. D. (2011). Reproducible research in computing science. *Science*, *334*(6060), 1226–1227. https://doi.org/10.1126/science.1213847

Pineau, J. (2020). The Machine Learning reproducibility checklist (v2.0, Apr.7 2020. Retrieved from www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist-v2.0.pdf

Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F., et al. (2020). Improving reproducibility in Machine Learning research (A report from the Neurips 2019 reproducibility program). ArXiv Retrieved from http://arxiv.org/abs/2003.12206

Pinkovskiy, M., & Sala-i-Martin, X. (2016). Lights, camera … income! Illuminating the national Accounts-household surveys debate. *Quarterly Journal of Economics*, *131*(2), 579–631. https://doi.org/10.1093/qje/qjw003

Renard, F., Guedria, S., Palma, N. D., & Vuillerme, N. (2020). Variability and reproducibility in deep learning for medical image segmentation. *Scientific Reports*, *10*(1), 1–16. https://doi.org/10.1038/s41598-020-69920-0

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. [cs] Retrieved from https://arxiv.org/abs/1409.0575

Sidi, F., Shariat Panahy, P. H., Affendey, L. S., Jabar, M. A., Ibrahim, H., & Mustapha, A. (2012). Data quality: A survey of data quality dimensions. In *2012 International conference on information retrieval & knowledge management (CAMP). IEEE*. https://doi.org/10.1109/infrkm.2012.6204995

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. [cs] Retrieved from https://arxiv.org/abs/1409.1556

Suel, E., Polak, J. W., Bennett, J. E., & Ezzati, M. (2019). Measuring social, environmental and health inequalities using Deep Learning and street imagery. *Scientific Reports*, *9*(1), 6229. https://doi.org/10.1038/s41598-019-42036-w

United Nations (2015). Transforming our world: The 2030 Agenda for sustainable development. Retrieved from https://sdgs.un.org/publications/transforming-our-world-2030-agenda-sustainable-development-17981

Walsh, I., Fishman, D., Garcia-Gasulla, D., Titma, T., Harrow, J., Psomopoulos, F. E., & Tosatto, S. C. E. (2020). DOME: Recommendations for machine learning validation in biology. In *Group, The ELIXIR Machine Learning Focus Group* (Vol. *1–21*). arXiv: 2006.16189. Retrieved from http://arxiv.org/abs/2006.16189

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., et al. (2016). The FAIR guiding principles for scientific data management and stewardship. *Science Data*, *3*(1), 160018. https://doi.org/10.1038/sdata.2016.18

Xie, M., Jean, N., Burke, M., Lobell, D., & Ermon, S. (2016). Transfer learning from deep features for remote sensing and poverty mapping. In *30th AAAI conference on artificial intelligence, AAAI 2016* (pp. 3929–3935). Retrieved from http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12196

Xu, Y., & Goodacre, R. (2018). On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, *2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

Yeh, C., Perez, A., Driscoll, A., Azzari, G., Tang, Z., Lobell, D., et al. (2020). Using publicly available satellite imagery and Deep Learning to understand economic well-being in Africa. *Nature Communications*, *11*(1), 2583. https://doi.org/10.1038/s41467-020-16185-w

Yen, A., Flowers, B., Luo, W., Nagesh, N., Tueller, P., Kastner, R., & Pannuto, P. (2021). A UCSD view on replication and reproducibility for CPS & IoT. *CPS-IoTBench 2021 - Proceedings of the 2021 Benchmarking Cyber-Physical Systems and Internet of Things* (pp. 20–25). https://doi.org/10.1145/3458473.3458821

Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., et al. (2018). Accelerating the Machine Learning lifecycle with MLflow. *IEEE Data Engineering Bulletin*, *414*(2018), 39–45. https://databricks.com/wp-content/uploads/2020/08/ieee_mlflow.pdf