

MACHINE LEARNING

for

DISASTER RISK MANAGEMENT

A guidance note on how machine learning can be used for disaster risk management, including key definitions, case studies, and practical considerations for implementation



The World Bank

1818 H Street NW
Washington DC 20433
www.worldbank.org

Disclaimer

This document is the product of work performed by the World Bank and GFDRR with external contributions. The findings, interpretations, and conclusions expressed in this document do not necessarily reflect the views of any individual partner organizations of the World Bank, Global Facility for Disaster Reduction and Recovery (GFDRR), the Executive Directors of the World Bank, or the governments they represent.

The World Bank does not guarantee the accuracy of the data included in this work. The boundaries, colors, denomination, and other information shown in any map in this work do not imply any judgment on the part of The World Bank concerning the legal status of any territory or the endorsement or acceptance of such boundaries.

Rights and Permissions

The World Bank supports the free online communication and exchange of knowledge as the most effective way of ensuring that the fruits of research, economic and sector work, and development practice are made widely available, read, and built upon. It is therefore committed to open access, which, for authors, enables the widest possible dissemination of their findings and, for researchers, readers, and users, increases their ability to discover pertinent information.

The material in this work is made available under a Creative Commons 3.0 By IGO License. You are encouraged to share and adapt this content for any purpose, including commercial use, as long as full attribution to this work is given. More information about this license can be obtained at:

<http://creativecommons.org/licenses/by/3.0/igo/>

Any queries on rights and licenses, including subsidiary rights, should be addressed to the Office of the Publisher, The World Bank, 1818 H Street NW, Washington, DC 20433, USA; fax: 202-522-2422; e-mail: pubrights@worldbank.org

Attributions

Please cite the work as follows:
GFDRR. 2018. Machine Learning for Disaster Risk Management. Washington, DC: GFDRR.
License: Creative Commons Attribution CC BY 3.0.

ACKNOWLEDGMENTS

This guidance note was prepared by Vivien Deparday, Caroline Gevaert, Giuseppe Molinaro, Robert Soden, and Simone Balog-Way.

The team is grateful for discussion, feedback and contributions from Sarah Antos, Cristoph Aubrecht, Alanna Simpson, Trevor Monroe, and his team, Keith Garrett and Sabine Chandradewi Loos. This publication was supported by the contributors of the case studies included in this publication. We thank them for their time and effort.

PHOTO CREDITS

Photos have been sourced from the following locations with full rights: [World Bank website](#)

TABLE OF CONTENTS

3	1. INTRODUCTION
6	2. A MACHINE LEARNING PRIMER
7	2.1 What Is Machine Learning?
10	2.2 Machine Learning Terminology
11	2.3 Supervised Machine Learning: Classification and Regression
12	2.4 Unsupervised Machine Learning
13	2.5 Deep Learning
14	3. APPLICATIONS AND OUTLINE OF A MACHINE LEARNING PROJECT
15	3.1 DRM Applications of Machine Learning
16	3.2 Outline of a Machine Learning Project
18	4. CONSIDERATIONS FOR IMPLEMENTING A MACHINE LEARNING PROJECT
19	4.1 Selecting Suitable Input Data
21	4.2 Evaluating Model Output
25	4.3 Expertise, Time, Infrastructure and Costs
26	4.4 Ethics: Privacy and Bias Considerations
28	5. MACHINE LEARNING IN THE COMMONS
29	5.1 Open Data
29	5.2 Open-source Software and Documented Methodology
30	5.3 Crowdsourcing and Capacity Building
31	5.4 Machine Learning for Sustainable Development: From Use Cases to Standardized Training Data
32	6. CASE STUDIES IN DISASTER RISK MANAGEMENT
34	6.1 Physical Exposure and Vulnerability
38	6.2 Social Exposure and Vulnerability
41	6.3 Risk Mapping and Damage Prediction
45	6.4 Post Disaster Event Mapping and Damage Assessment
47	7. GLOSSARY
48	8. REFERENCES AND RESOURCES
48	8.1 Online Resources
48	8.2 Videos and Talks
48	8.3 Infographics and Interactive Resources
48	8.4 Articles and Blogs
48	8.5 Conferences and Meetings
49	8.6 Challenges and Competitions
49	8.7 Other References, Articles and Textbooks

1. INTRODUCTION



1. INTRODUCTION

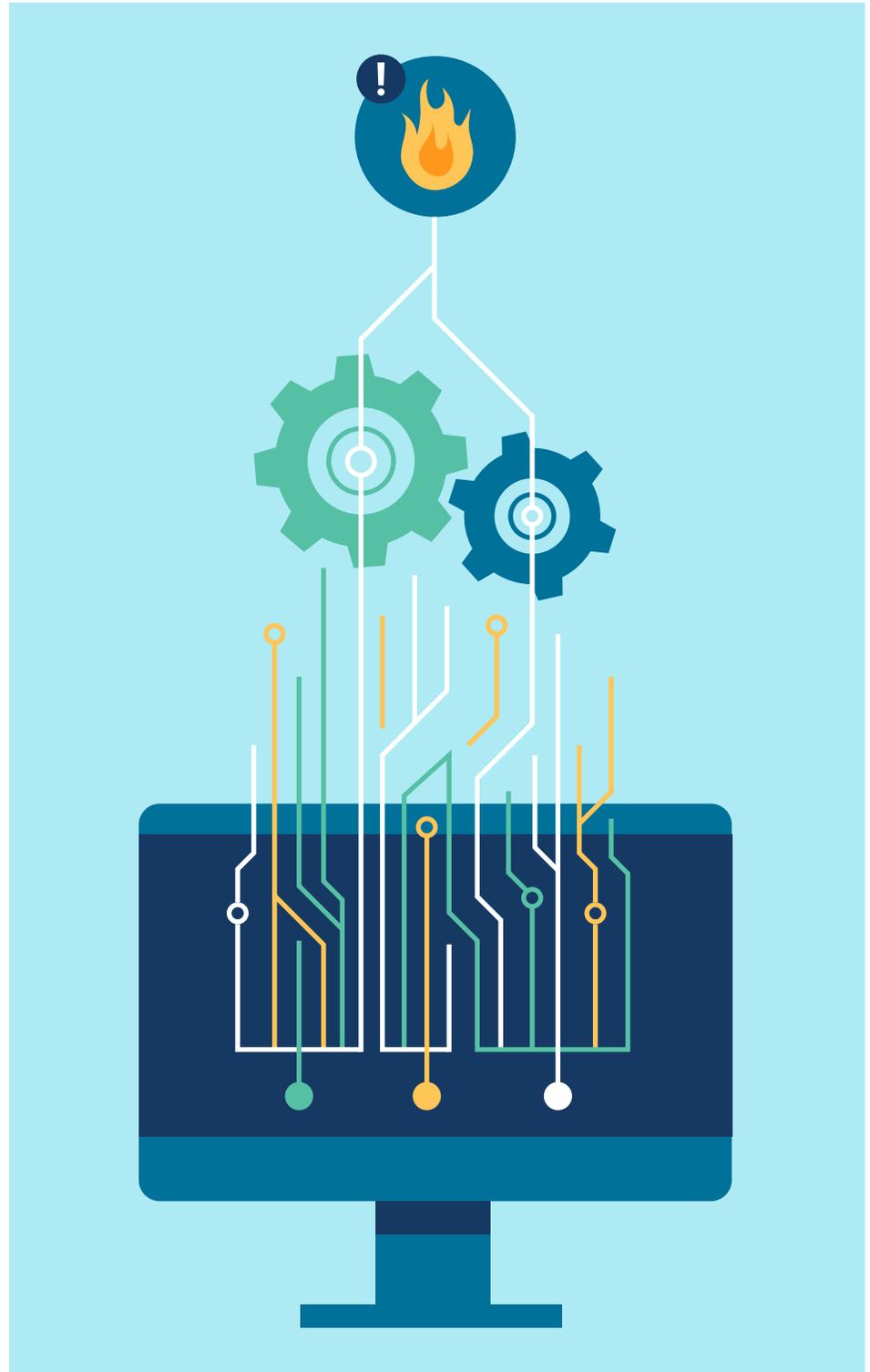
Evidence-driven disaster risk management (DRM) relies upon many different data types, information sources, and types of models to be effective. Tasks such as weather modelling, earthquake fault line rupture, or the development of dynamic urban exposure measures involve complex science and large amounts of data from a range of sources. Even experts can struggle to develop models that enable the understanding of the potential impacts of a hazard on the built environment and society.

In this context, this guidance note explores how new approaches in machine learning can provide new ways of looking into these complex relationships and provide more accurate, efficient, and useful answers.

The goal of this document is to provide a concise, demystifying reference that readers, from project managers to data scientists, can use to better understand how machine learning can be applied in disaster risk management projects.

There are many sources of information on this complex and evolving set of technologies. Therefore, this guidance note is aimed to be as focused as possible, providing basic information and DRM-specific case studies and directing readers to additional resources including online videos, infographics, courses, and articles for further reference.

A machine learning (ML) algorithm is a type of computer program that learns to perform specific tasks based on various data inputs or rules provided by its designer. Machine learning is a subset



of artificial intelligence (AI), but the two terms are often used interchangeably. For a thorough discussion of the differences and similarities of the terms ML and AI, see Section 2. As the name implies, an ML algorithm's purpose is to "learn" from previous data and output a result that adds information and insight that was not previously known. This approach enables actions to be taken on the information gathered from the data; sometimes in near real time, like suggested web search results, and sometimes with longer term human input, like many of the DRM case studies presented in this document.

Over the past few decades, there has been an enormous increase in computational capacity and speed and available sensor data, exponentially increasing the volume of available data for analysis.

This has allowed the capabilities of ML algorithms to advance to nearly ubiquitous impact on many aspects of society.

Machine learning and artificial intelligence have become household terms, crossing from academia and specialized industry applications into everyday interactions with technology—from image, sound, and voice recognition features of our smartphones to seamlessly recommending items in online shopping, from mail sorting to ranking results of a search engine. The same technology is being leveraged to answer bigger questions in society, including questions about sustainable development, humanitarian assistance, and disaster risk management.

When several ML algorithms work together, for example, when fed by a large quantity of physical sensors, it is possible for a computer to interact with the physical world in such a way that the computer system, or robot, appears to

be behaving intelligently. For example, self-driving cars, robotics that mimic and surpass human capacities, and supercomputers can now outperform humans on specialized tasks. The same expectation is, and should be, held for ML as it applies to improving our capacity to accurately, efficiently, and effectively answer pressing societal questions. The case studies in this guidance note range from the identification of hurricane and cyclone damage-prone buildings to mapping the informal settlements that house the most vulnerable urban populations.

For the understanding of disaster risk, machine learning applies predominantly to methods used in the classification or categorization of remotely sensed satellite, aerial, drone, and even street-level imagery, capitalizing on a large body of work on image recognition and classification. But applications also span other types of data: from seismic sensor data networks and building inspection records to social media posts. All the advancements made in the applications of ML can and are being used to solve bigger issues confronting humans, from making the most of our land to preparing for and recovering from crises.



Photo Credit: WB/DaLA team

2. A MACHINE LEARNING PRIMER



2. A MACHINE LEARNING PRIMER



MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Machine learning is a type of artificial intelligence. ML algorithms, some more simple and narrowly focused than others, have been a part of computer science since the late 1950s. Driven by computer vision, ML algorithms were pioneered in fields like satellite remote sensing and statistical data analysis. Now they power many different aspects of our everyday digital lives, from search engines to online shopping.

Artificial intelligence (AI) was founded as an academic discipline in 1956. Although AI is often used as a synonym for machine learning, there are some major differences that need disambiguation. AI has become a catch-all term that includes all machine learning software as well as artificial general intelligence (AGI), or strong AI. AGI refers to possible, future versions of AI computers that are generalized, self-aware, and indistinguishable to humans when tested. This is not the current state of AI and not the focus of this document.

2.1 WHAT IS MACHINE LEARNING?

Machine learning algorithms that are trained by humans based on pre-existing data are called “supervised,” whereas those that learn solely from data without human input are referred to as “unsupervised.” This traditional dichotomous separation is becoming more and more blurred every day, as projects employing ML algorithms make use of both types. Sometimes these methods are easily categorizable, such as when a project employs an unsupervised ML algorithm in one step and a supervised one in another. Other times, the actual ML algorithm is hybridized. Some examples of these ML algorithms are reinforcement learning, transfer learning, Generative adversarial networks (GANs), semi-supervised learning, and so forth (see box on page 8 for more information about reinforcement learning).

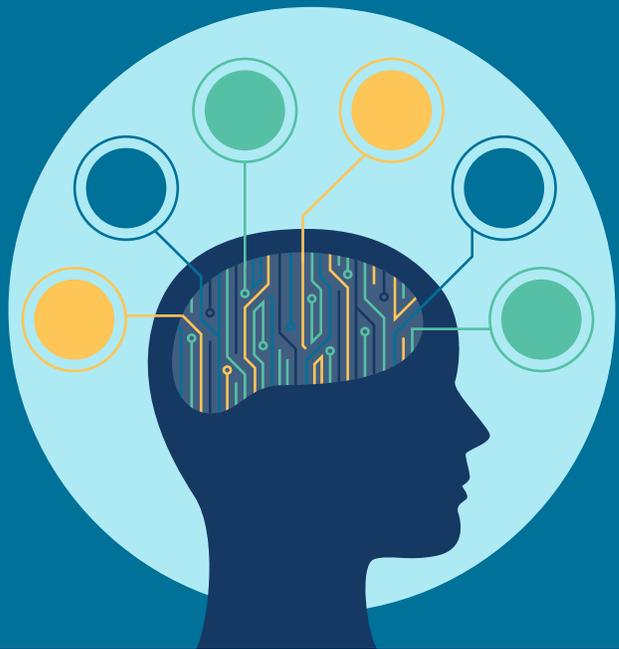
To understand the nuts and bolts of ML, we need to understand the basic difference between the two approaches of supervision and how they can be leveraged to obtain the answers that we are looking for. A list of definitions of terms used throughout this document is found on page 10. In supervised ML, a user inputs a training dataset (sometimes called a “labelled training dataset”) that identifies correct answers and incorrect answers to help the algorithm learn relevant patterns in the data. These patterns can be identified by categories. For example, the machine can learn that data A are images of cats and data B are images of chairs because the algorithm has been trained by a user that certain characteristics—whiskers and paws—indicate a cat and not a chair. Thus instead of a dataset being comprised

REINFORCEMENT LEARNING

Reinforcement learning is a type of machine learning that takes a page from behavioral psychology. Simply put, the training dataset and rules in an ML algorithm are not binary (yes or no) decisions, but rather they attempt to achieve a balance between data exploration and accuracy. In other words, the model is allowed to make mistakes and explore the data within certain parameters.

A famous example of a hybrid system is Google's DeepMind AI project, which relies on reinforcement learning—in this case, a hybridized artificial neural network combined with supervised learning methods. Bought by Google in 2014, DeepMind has been on the forefront of AI advancement and even developed programs that can defeat humans at complex games like Go.

<https://www.technologyreview.com/s/533741/best-of-2014-googles-secretive-deepmind-startup-unveils-a-neural-turing-machine/>



by anonymous data A and data B, thanks to ML we now know that data A and B are different (cats and chairs, respectively). In extending this concept to DRM, consider the identification of rooftops in a satellite image. The ML algorithm will need a training dataset that has both rooftops and non-rooftop areas, such as trees, identified and labelled. The ML algorithm will learn what characteristics are indicative of a rooftop from that training dataset and can then classify the rest of the image based on the training dataset.

In unsupervised ML algorithm, the algorithm uses statistical methods, like clustering analysis or neural networks, to attempt to group data with similar characteristics together, such as roofs of the same color or texture in the DRM example above. It is then up to the user to add semantic information (labels) to the data-driven results. In unsupervised ML, many other separations in the data might be discovered; not just group (A) and (B), but possibly (C), (D), and (E), etc. Therefore, this is also understood as an exploratory tool in which the user does not always know what can be learned from the ML algorithm. If we extend the analogy of a function here as well, if $y = f(x)$, in unsupervised

ML algorithms, we only input (x) and the ML algorithm applies a series of statistical methods to identify the best-fitting function (f) that splits the data into a result (y). This learned function can sometimes be applied to completely different datasets.

For example: There are three kinds of datasets required for ML algorithms; training, validation, and testing. Training datasets are used in the beginning stages to train the model to recognize features and patterns in the data. Validation datasets are used to determine the best model parameters and are used before the testing set. Testing datasets are kept separately from the model while its training so that it can be used after training to test the accuracy.

In order to identify patterns in data, individual pixels that make up the image are analyzed in a type of analysis called "image analysis". Object-based image analysis (OBIA), which has also proven its usefulness over the years as well, organizes neighboring pixels into meaningful groups. In both types of analyses a pixel can be described by color, texture, or other raster geographic information system (GIS) information such as elevation or temperature.



Photo Credit: World Bank

In OBIA, samples can also be described by their area, shape, or orientation.

Although recent developments are delivering very powerful ML algorithms, it is important to remember that a model is only as good as the data used to train it. First, the categories of data should be distinguishable according to the features provided. Also, the training dataset should be representative of the variability in features of the specific group of data. That is to say, if the target class is a building, the training data should include examples of the variety of building appearances.

It is important to note that training sets for ML algorithms can be geographically biased, and it is important to ensure geographic diversity for the training set. For example, buildings tend to appear differently in European cities than African cities. If an ML algorithm is trained using examples from one region, it will likely perform worse on data from a different region where objects appear differently. Such diversity should be taken into account when putting together the training dataset. (See table on page 10 for more details.)

Once you identify the characteristics/features of data that best explain the data (explanatory variables), you can then use both types of methods to identify relevant patterns.

On top of different types of ML algorithms, there are also a fair amount of ambiguous terms like artificial intelligence, machine learning, big data, and deep learning, among others. These have become somewhat interchangeable in the vernacular of development agencies, technology service vendors, and mainstream media alike. This document attempts to demystify these terms.

2.2 MACHINE LEARNING TERMINOLOGY

Despite the plethora of available ML algorithms and the even greater number of methods in libraries that can be developed into customized models, the general process for an ML algorithm is the same.

A thorough list of references on ML is available in section 8: References and Resources.

Term Used in Document	Alternative Terms	Definition
Feature	Attribute, variables, dimensions	Characteristics used to describe the data samples and predict the output; not to be confused with a “feature” in GIS, which refers to a physical “object” with specific attributes
Output variable	Predicted variable, target variable	Phenomenon you want the model to predict; the desired output of the model
Sample	Reference data	Set of samples with known features and class labels. This set of labelled samples should be divided into training, validation, and testing.
Training dataset		Labelled samples used to train the model, i.e., learn the relevant patterns in the features which are relevant to predict the output variable
Validation dataset	Cross-validation set	Labelled samples to validate the model before the testing set; used to help determine the best model parameters
Testing dataset		A set of labelled “unseen” training samples which are used to determine model accuracy; cannot be included in the model training
Cluster		Group of samples which are grouped together based on similarities identified by an unsupervised algorithm
Class	Group, bin, categories	A class, or group, is the result of the splitting of a dataset into two or more groups of data that share some common characteristic. The term “class” is most commonly used in supervised ML algorithms, for example in satellite remote sensing, where features like “rooftops” may be split into class A, and features like “vegetation” may be split in class B and so forth.

2.3 SUPERVISED MACHINE LEARNING: CLASSIFICATION AND REGRESSION

Supervised learning can be divided into **classification** and **regression** problems. In classification, the intended output is a semantic label or class. Tighter sentence: For instance in flood mapping classification problems would label each pixel in an image as “flooded” or “not flooded (see case study 6.4.1 Flood Mapping). Similarly, cyclone damage assessments may classify buildings suffering from “mild,” “medium,” or “severe” damage (see case study 6.4.3 Cyclone Damage Assessment). Regression problems aim to predict a continuous variable, such as predicting the poverty rate for each administrative unit based on characteristics such as type of buildings,

amount of green space, population density, or other traits. (See case study 6.2.1. Sri Lanka Poverty Mapping).

There are many different types of supervised ML algorithms, which sometimes have fundamentally different architectures. The most common classification algorithm is logistic regression, while the most common regression algorithm is linear regression. Some of the most well-known classification algorithms are random forests, gradient boosting, support vector machines (SVMs), naive, and gradient boosting Bayes networks (see box below). Random forests and SVMs can also be adapted to regression problems.

Name	Description	Advantages	Disadvantages
Random forest 	A group of decision trees. Each tree is a hierarchy of decisions which divide samples into two groups depending on the value of a single feature at a time	<ul style="list-style-type: none"> • Less susceptible to noise • Can handle large numbers of training samples 	<ul style="list-style-type: none"> • A decision tree’s disadvantage is high variance in its results, however random forests solve this problem by averaging many trees. The drawback: as you average many decision trees, it might be hard to interpret the results • Slower than other methods in the testing phase
Gradient boosting 	Similar to random forests, but trains each tree sequentially. The samples which have the highest uncertainty according to the results of the previous iteration are prioritized	<ul style="list-style-type: none"> • Studies suggest it can be more accurate than random forests 	<ul style="list-style-type: none"> • It is more challenging to train the algorithm
Support vector machine 	Uses kernel functions (a class of algorithms used for pattern analysis) to describe the non linear differences between training samples	<ul style="list-style-type: none"> • More suitable for situations with limited reference points • Can easily handle large numbers of input features • Can learn non-linear relations between features 	<ul style="list-style-type: none"> • Computational complexity when there is a large training set • Sensitive to noisy data
Naive Bayes 	A graphical model describing the probabilistic relations between feature values and class labels	<ul style="list-style-type: none"> • Simple to implement • Scales easily • Feature importance is easy to interpret 	<ul style="list-style-type: none"> • Assumes all features to be independent from each other, which is often not the case in real-world applications

2.4 UNSUPERVISED MACHINE LEARNING

In unsupervised ML, the machine takes an input dataset and applies a series of mathematical and statistical models to identify patterns, without the user providing labelled training data. One of the most common applications is clustering, where samples are grouped based on similarity. Other applications include dimensionality reduction and anomaly detection to reduce variance in a dataset and filter it for outliers.

Unsupervised methods are purely driven by the patterns in the data. The patterns are based on the statistical characteristics of the input samples. This means that the user doesn't need to provide labelled training sets (which can be costly and difficult to

put together), but also means that the patterns identified by the ML algorithm may or may not be useful for the user. Due to this uncertainty and the difficulty of understanding the performance of unsupervised ML algorithms, they are often used for data discovery and exploration.

Often, the results of unsupervised ML algorithms are fed into supervised ML algorithms, where human input and experience can help a dataset reach its targeted accuracy more quickly. There are three types of unsupervised machine learning, as described in the box below: K-means clustering, principal component analysis and t-SNE.

Name	Description	Advantages	Disadvantages
K-means clustering 	A clustering technique which iteratively calculates the "average value" (e.g., centroid) of each cluster and assigns each sample to the nearest cluster	<ul style="list-style-type: none"> • Simple implementation, performs well • Distance metric can be defined by the user 	<ul style="list-style-type: none"> • User must define number of classes
Principal component analysis 	Transforms the data to features which maximize the variance (differences) between samples	<ul style="list-style-type: none"> • Can be used to retain the relevant information while decreasing data dimensionality 	<ul style="list-style-type: none"> • Resulting features are difficult to interpret
t-SNE 	Non linear data dimensionality reduction technique suitable for visualization purposes	<ul style="list-style-type: none"> • Helps understand patterns by visualizing similar groups • Captures complex similarities 	<ul style="list-style-type: none"> • Sensitive to hyperparameters • Computational complexity

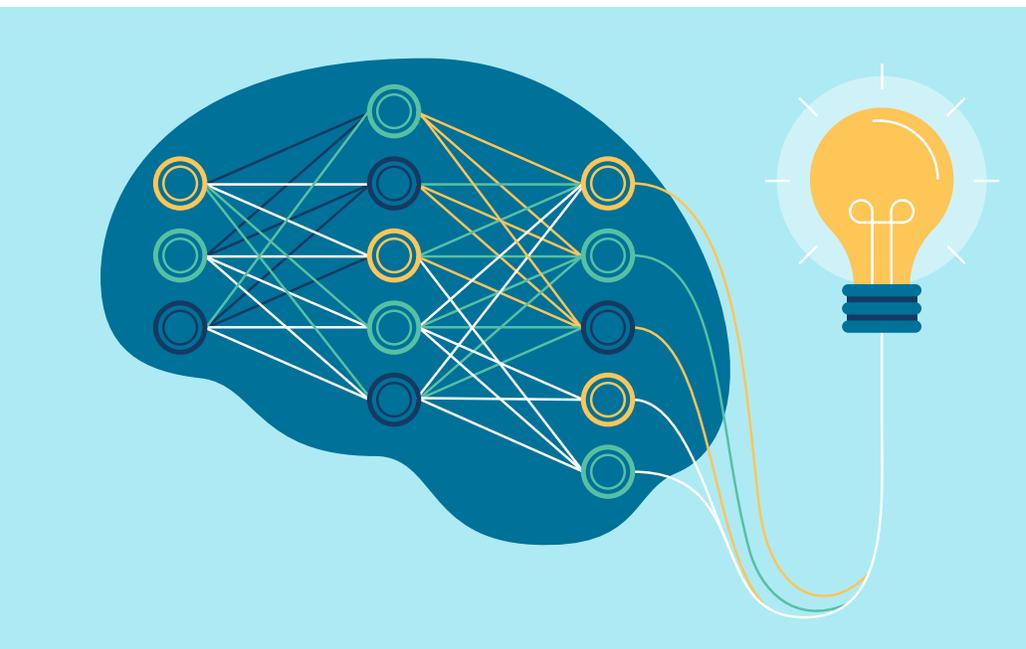
2.5 DEEP LEARNING

Artificial neural networks are also commonly referred to as deep learning. Neural nets, as they are called for short, work with several hidden layers that are nested between the inputs and outputs and are connected to each other through connections that resemble neurons in a brain. These neurons all have mathematical formulas that optimize the accuracy of the categorization, most notably using a method called backpropagation. Backpropagation is short for the backwards propagation of errors. It is a method used to calculate the gradients between optimal values (weights) in the “neurons”. The term “deep learning” comes from the fact that these hidden layers can be nested upon other hidden layers to some depth, but has nothing to do with the actual “depth” of the content. In other words, deep learning methods can be just as shallow as other ML methods. Deep learning can be applied to supervised and unsupervised ML tasks.

Recently, deep learning has gained much popularity as it is capable of obtaining unprecedented accuracies for large ML algorithm problems. Convolutional neural networks were developed for image classification,

making them useful for tasks involving remotely sensed imagery and/or other spatial data. The more recent fully convolutional neural networks (FCNs) are especially relevant for spatial applications, as they are more efficient for processing large scenes. New networks and models are continuously being developed for various applications, many of which are available as open-source libraries. However, they require much more training data and have significantly higher computational requirements than the other methods. It is therefore important to consider the complexity and available resources of the classification problem when choosing a suitable algorithm.

In fact, supervised decision tree algorithms can be visualized and explained in terms of a two-dimensional neural network. While adding nodes, or “decisions,” to a decision makes it deeper (more hierarchical decisions), the power of deep learning is that it can apply a number of hidden layers of nodes (decisions) that make the neural network wider and more intricate, effectively adding more and more dimensions/layers.



3. APPLICATIONS AND OUTLINE OF A MACHINE LEARNING PROJECT



3. APPLICATIONS AND OUTLINE OF A MACHINE LEARNING PROJECT

3.1 DRM APPLICATIONS OF MACHINE LEARNING

As ML approaches are proliferating in all fields of expertise, DRM is no exception—new applications are being developed every day. They are developed to improve the different components of risk modelling such as exposure, vulnerability, hazard, and risks, but also for prioritization of resources during disaster response and reconstruction.

A number of early applications have been looking at better understanding exposure to disasters from the physical side (see case study 6.1 Exposure and Physical Vulnerability) as well as from the socioeconomic side (see case study 6.2 Social Exposure and Vulnerability). These types of applications have relied mainly on the analysis of satellite imagery characteristics (in the visible wavelengths of the electromagnetic spectrum as well as radar and LiDAR), often coupled with the addition of georeferenced census data. Newer applications are starting to also leverage computer vision approaches to identify vulnerabilities from street view images (see case study 6.1.1 Guatemala City Building Earthquake Vulnerability). In the near future, by combining all these approaches and data sources, we can imagine having a detailed exposure database at scale that can be updated any time new imagery is available.

The traditional modelling of hazards, such as earthquakes, wildfires, and weather forecasts, is also being augmented by ML approaches. This application uses time coded data from hundreds or thousands of sensors

(whether physical, like weather stations or earthquake stations, or remote, such as satellites) and other geophysical characteristics to predict hazard output (see case study 6.4.3 Wildfire Prediction).

Another approach involves looking at the impact of the hazard on the exposure data, or in other words the risk. To do so, data is gathered on the exposure (see section above), and the damage prediction algorithm is trained using the impact of past events. Next, it infers and identifies the key aspects of exposure that have an influence on the disaster impact (see case studies 6.3.1 and 6.3.2 Flood Damage Prediction and Machine Learning-Powered Seismic Resilience for San Francisco). Once trained, those algorithms can be used to predict damage in other cities or countries.

Post-disaster event mapping and damage assessment are also emerging as key applications. Although difficult using optical data from satellites, some approaches are using higher-resolution optical imagery from Unmanned Aerial Vehicles (UAVs) (see case study 6.4.2 Cyclone Damage Assessment), while others use more complex data that are difficult for humans to interpret but are simple for machines to sift through to identify new relations, like radar data (see case studies 6.4.1 Flood Extent Mapping and 6.4.2 Cyclone Damage Assessment).

Other new applications involve prioritizing resources during response

or recovery phases. For instance, prioritization of building inspections using previous building inspection records and their outcomes, social media mining for response awareness and resource prioritization,¹ monitoring of rebuilding and recovery activities using computer vision or on-site pictures to control quality, supporting insurance claims using computer vision to identify crop or building damage from pictures, and many others.

¹<https://www.floodtags.com/>

3.2 OUTLINE OF A MACHINE LEARNING PROJECT

This section provides a brief overview of the general steps which must be followed to set up an ML project. The next section will describe how to prepare the inputs for the problem and evaluate the quality of the ML algorithm results and required project resources in more detail.



1. Project goals are defined: What do we want the ML algorithm to predict or classify? The objective of the DRM project should be translated to the output variable that is targeted. For example, ML can support a poverty mapping project by estimating a poverty index (see case studies 6.2.1 and 6.2.3). Building vulnerability can be translated into classifying the type of roof material used. More examples of how DRM objectives can be translated into ML projects are given in the case study section below.



2. Data/imagery sources: This obviously also depends on the objective. ML algorithms have been used for decades on satellite remote sensing imagery of many different kinds and resolutions. Currently, work in the DRM sector often involves using high-resolution (sub-meter spatial accuracy to 10 m or so) panchromatic and multispectral imagery from satellites, drones, and airplanes. However, as discussed above, ML algorithms can be applied to data of all kinds, so big data sources that are actively mined can come from household surveys (see case study 6.3.1 Flood Damage Prediction), census data (see case study 6.2.3 Stanford Poverty Study), social media (see case study 6.4.1 Flood Extent Mapping), tweets, and cell phone locations, to name a few.



3. Training/validation data collection: Labelled samples or reference data are required to train the model and validate the ML algorithm outputs. Projects using high-resolution satellite imagery often manually create this data. If the goal is to map roofs in satellite imagery, then a “training dataset” is manually drawn so that there is an input dataset for the ML algorithm that teaches it what roofs look like. Crowdsourcing can be used to speed up this process (see case study 6.4.2 Cyclone Damage Assessment). Field data are another important source of reference data, such as using household surveys to validate the poverty level (see case study 6.2.3 Stanford Poverty Study). The collection of these labelled samples is often the most expensive part of ML projects.



4. Exploration of dataset: Exploratory data analysis is an important step, as it helps determine which algorithm to use and the best data to include. This analysis also clarifies which input variables are correlated with each other, what is most closely related with the output variable, the distributions of variables, or even whether you can combine/transform input variables. This step also cleans the data of outliers, which could otherwise skew the results dramatically by altering the variance in the data in disproportionate ways.



5. Choice of algorithm: When choosing an algorithm, there is no silver bullet or one-size-fits-all solution. The best way to decide is to analyze which algorithms have been used to tackle similar problems in the past. The choice of algorithm may also depend on the size of the training set, number of features, and computational resources available.

Sometimes, multiple models are applied and the best performing one is selected; however, it is important to understand and compare models that have been tuned optimally so that the comparison is actually assessing the model effectiveness and is not biased by the parametrization behind it. If the review or application of various models is too time consuming, then the support of specialized experts should be sought in order to start on the right foot. For example, Task Team Leaders (project managers) who have no background in ML, statistics, or computer science should seek the advice of data science experts at the beginning of the project.



6. Developing the code and running the algorithm: Some ML algorithms are already developed in a programming language and available in user interfaces, such as the image classification algorithms available through the ENVI software for remote sensing, through the Google Earth Engine, or DigitalGlobe online platform GBDX for cloud-computing remote sensing image classification. There are a number of readily available ML algorithms inside remote sensing and GIS software packages, some of which are free—like the GRASS GIS plug in for QGIS.

In addition, any number of ML algorithms from open or proprietary libraries can be combined and customized to achieve any project's goals. In custom applications, ML algorithms can be programmed in a variety of programming languages and tools that range from R, Matlab, and ESRI arcPy to GDAL and GRASS. Increasingly open-source platforms like TensorFlow² have matured and remote sensing-specific ML tools like Mapbox's RoboSat are openly available on Github.³

On top of that, a number of customized ML services are available on cloud computing platforms like AWS, Azure, and Google Cloud services. In fact, some of the WB projects showcased in this document have been run on these platforms.

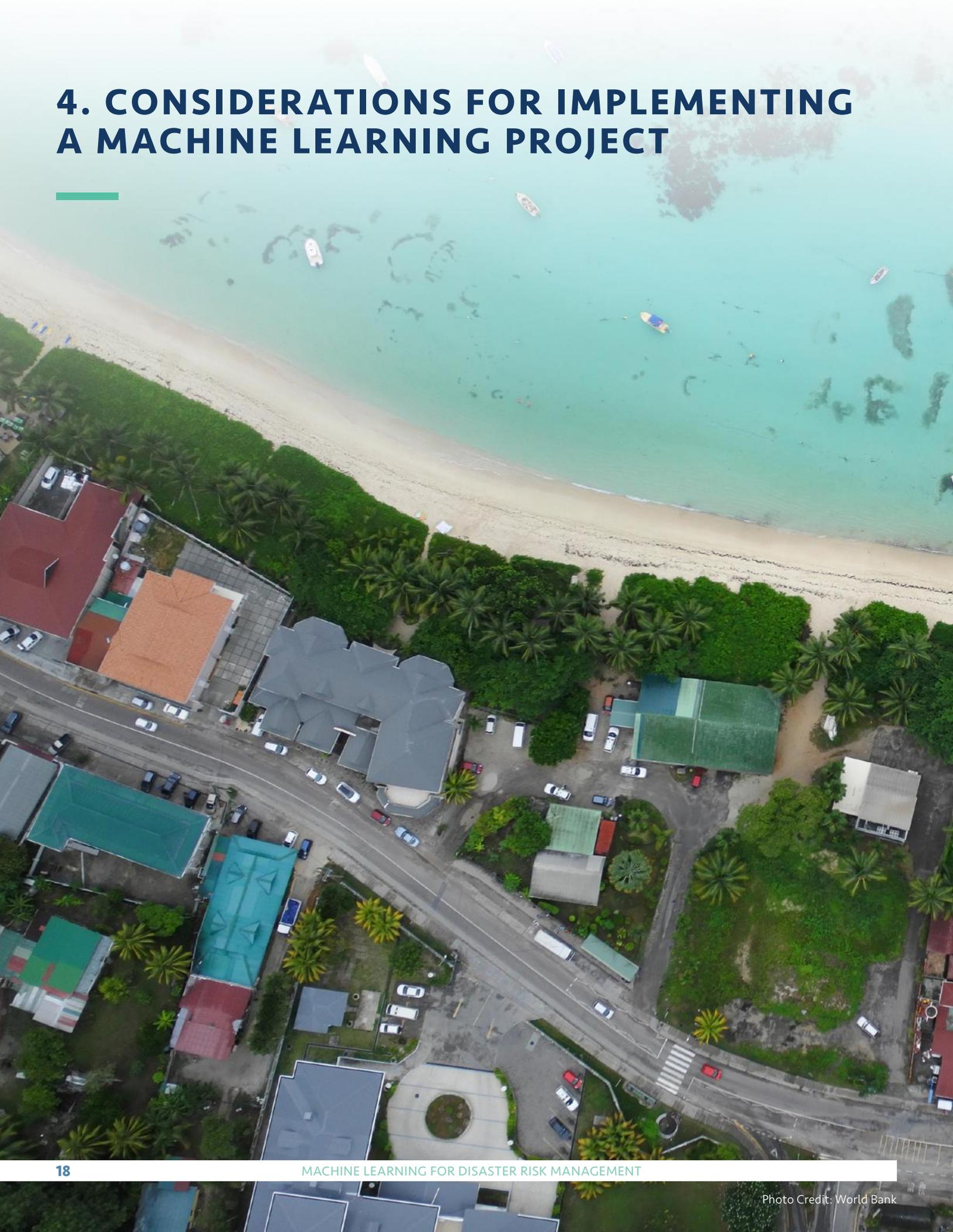


7. Validation, reinforcement, and re-running: Any ML algorithm produces an output that needs to be validated for accuracy. This is usually achieved by comparing the output data to a validation dataset that is considered the “truth,” or accurate within a range that is acceptable for the project's goals. For example, a map of all the roofs in an image drawn by human photo interpreters can be compared to the ML algorithm output to assess its accuracy. Modifying the training dataset and parameters needed to run the algorithm might yield more accurate results, so the intermediate results are used to rerun the ML algorithm with the goal of increasing accuracy. Section 4.2 discusses how to assess the model's accuracy in more detail.



8. Final data output: The final data output is achieved once the accuracy of the output dataset is deemed adequate for the goal of the project. The final output accuracy needed can differ greatly, and there's no quick rule of thumb. A final accuracy of 50%, for example, means that the model is no better than random chance at predicting the variable of interest, which means the model is useless. The concept of accuracy is different and possibly ill posed when talking about unsupervised ML algorithms, where the goal might be data discovery. However, in unsupervised cases, something can be learned from output data even if the ML algorithm does not give an accurate classification or explanation of the final variable.

4. CONSIDERATIONS FOR IMPLEMENTING A MACHINE LEARNING PROJECT



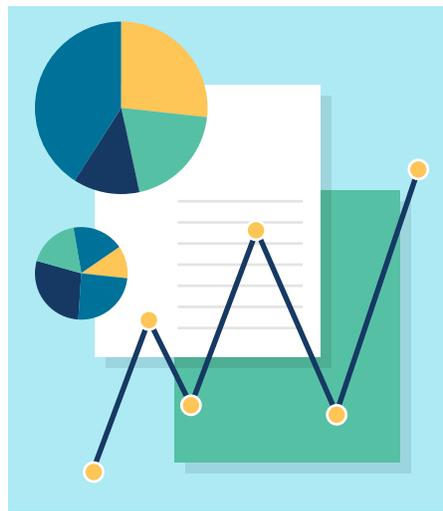
4. CONSIDERATIONS FOR IMPLEMENTING A MACHINE LEARNING PROJECT

There are several issues that need to be considered when planning an ML project. We have divided these into the subsections below: selecting suitable input data; evaluating model output; expertise, time, infrastructure, and costs; and ethics: privacy and bias considerations.

4.1 SELECTING SUITABLE INPUT DATA

Once the project goal is defined, the first step is to do a quick data inventory. Which data do I have that might help me predict my output variable? If I need to find additional data, which characteristics should I take into account when selecting suitable input data? It is good to think about the relevance of the dataset for your proposed goal and how different datasets provide different pieces of information about the problem. A number of open data sources are available, such as NASA and ESA satellite imagery, and derived geospatial products, such as OpenStreetMap, OpenAerialMap, World Bank Open Data, UNdata, the GEOSS portal, and the Humanitarian Data Exchange.

In the age of “big data,” more is not always better. ML can combine many different types of data, but adding irrelevant data may incur additional costs without improving the model predictions. At the same time, ML cannot magically obtain good predictions if the input data does not adequately relate to the targeted output. In general, it is good to check which similar data studies or projects that have been used. To help select relevant data, the following section provides an overview of some of the different data characteristics that can be taken into account to guide the selection of suitable input data.



Direct and indirect relations, and data best left out

The input data can be directly, indirectly, or not related to the output goal. One example of direct relations is identifying buildings in submeter satellite imagery. Here, roofs are visible in the satellite imagery, and one can easily assume that there is generally a building below a roof.

An indirect relation means that the information captured by the input data is somehow related to the output goal. For example, one can try to identify informal areas or poverty from satellite imagery. Buildings in informal settlements or poorer neighborhoods generally look different (low rise, corrugated iron roofs, very narrow footpaths) from buildings in planned

areas (often larger, more regularly spaced, gridded road networks). See case study 6.2.2. Informal Settlement Mapping for an overview of visual differences between formal and informal areas. However, it is important to remember that the physical representation of the buildings in the imagery does not actually have a direct link to the income of the building's inhabitants. The buildings in the imagery (input data) are therefore indirectly related to the poverty index (goal). Complex ML algorithms are capable of combining many sources of input data which are indirectly related to the objective to create reasonable predictions. However, it is important to remember that these relations are not always causal, but may simply show a correlation. Especially when predicting socioeconomic variables, the relationships between the input data and targeted output may vary strongly from one location to another.

Finally, some data is best left out of the analysis, even though it may be available. Data which are weakly related or inaccurate (i.e., "noisy" data) should also be left out or otherwise emphasized by the model. These issues can all be determined in the exploratory data analysis phase. Each additional type of input data makes the ML model more complex, so, adding irrelevant or low-quality data introduces unnecessary data-processing costs and may even lower the quality of the output predictions.

Image characteristics

In more traditional Earth observation or remote sensing applications, there are a number of image characteristics which are relevant when selecting datasets to use. Temporal resolution indicates the frequency with which images are captured over the same area. The timing of the data collected may also influence its suitability. For example, imagery collected during the winter or dry season may not be suitable for agricultural monitoring applications.



Photo Credit: World Bank

Spectral resolution has to do with which wavelengths of the electromagnetic spectrum (which "colors," when talking about the visible wavelengths) are observed by the imagery. Many cameras capture the same part of the spectrum as the human eye, often referred to as RGB, or Red-Green-Blue. Other sensors can observe parts of the spectrum which the human eye cannot see, but which can be very relevant and useful for a wide range of applications. For example, multispectral imagery containing Near-Infrared reflectance (NIR) is useful for discerning vegetation, and hyperspectral imagery is often used to identify different types of minerals for geological applications.

Also important is the spatial resolution. This defines the real-world size of each pixel in the image on the ground. A higher resolution means that the image pixel represents a smaller area on the ground. This allows smaller objects to be identified in the images. The resolution should be high enough to identify the object that is needed. However, it should be taken into account that higher resolution imagery also means larger file sizes and more computational complexity. Depending on the application of the ML project and unit of analysis, it may not always be necessary to select the data with the highest spatial resolution (or temporal or spectral, for that matter).

IMAGE CHARACTERISTICS

Temporal resolution: How often is the dataset acquired of the same area?

Spectral resolution: Which parts of the electromagnetic spectrum (essentially, which “colors”) are captured?

Spatial resolution: What is the actual size of each pixel on the ground?

Geographic coverage: What is the area over which the imagery is acquired?

Temporal coverage: What is the total timespan of the archive of the imagery/data available?

Context: Certain types of sensors are only adequate within certain physical contexts. For example, radar data that is valuable for building detection can be problematic in stony, hilly areas.

Other data

Sometimes, the output variable may be predicted more accurately with the support of additional data sources. Objects in urban settings often have many different colors and textures, so the addition of elevation or LiDAR information may be quite useful. Radar imagery can be useful for identifying changes in surfaces or obtaining data despite cloudy conditions. Socioeconomic studies may include census surveys aggregated at the administrative unit level. Terrestrial or street-view imagery can be used to provide information which cannot be seen from above, such as building wall material. Recently, social media information is also included, such as using tweets or crowdsourced geotagged images to identify flooded areas (see case study 6.4.1 Flood Extent Mapping). Tabular data, such as results from household surveys, can be used for assessing flood damage (see case study 6.3.1. Flood Damage Prediction). Again, a good starting point is to look at similar projects and find out which data they have used.

When including other data, it is important to link the unit of analysis. ML algorithms can be applied to pixels, vectors (such as building footprints), or samples. When integrating data from different sources, they should all be linked back to the same unit of analysis if we wish to use them in the same ML algorithm. For example, census data per administrative unit can be linked to

a vector file showing the administrative boundaries, giving the census data spatial dimensions, and enabling it to be combined with imagery.

When an ML algorithm involves many features, unexpected patterns can end up being the most important. Therefore, experimenting with combining multiple features can be one of the most crucial steps of feature engineering.

4.2 EVALUATING MODEL OUTPUT

Training, validation, and testing data

The division of the data into training, validation, and testing sets is key to evaluating the performance of an ML algorithm. The training set is used to teach the model to distinguish the classes we wish to predict. Each ML algorithm requires a number of model parameters to be set. By checking the accuracy of the trained model on the validation set, we can compare the different model parameter settings and choose the best ones for our particular problem. The third group is the testing set. This set should not be touched during model development and is only used at the end to check the accuracy of the final model output. In some cases, the testing dataset is actually a new dataset, such as in a case where you want to apply a previously developed model to a new region.

There is no specific rule regarding how to divide the reference data into these training, validation, and testing datasets.

One rule of thumb is to randomly allocate 50%, 25%, and 25% of the data to each set, respectively. The exact ratios may differ. Benchmarks to compare algorithms often require users to submit their model results for a set of data for which they are not given the reference labels.

Not only the quantity, but also the heterogeneity of the training samples are important for ML algorithms. However, a tipping point can be reached where too much data heterogeneity leads to unpredictable results. Likewise, a feature in one geographic region can resemble a completely different feature in another geographic context, so it's often necessary to have different models for different areas, even when the same output results are being targeted. Flood damage prediction models have been shown to obtain higher accuracies when trained using flood events of various magnitudes and geographical locations (see case study 6.3.1. Flood Damage Prediction).

Deep learning models aiming to assess cyclone damage to buildings had a significantly lower accuracy when applied to images of a different geographical region (see case study 6.4.2. Cyclone Damage Assessment). Ideally, similar quantities of samples should be available for the different classes. "Negative" examples are also important to include. For example, when training a classifier to recognize roofs, it can be essential to also collect a second dataset that contains "everything but roofs" so that the ML algorithm can learn with higher accuracy to separate roofs from everything else in the imagery.

Accuracy metrics

The accuracy of an ML algorithm can be described by a number of different quality metrics. For classification

problems, a confusion or error matrix can be used to show the relationship between the number of samples per class in the reference data and how they are classified by the output data. In general, the overall accuracy of an algorithm is calculated by dividing the total number of correctly classified samples by the total number of samples.

An algorithm's precision (i.e., correctness or user's accuracy) is the number of true positives divided by the sum of true positives and false positives per class. This describes the probability that a pixel is classified as part of the correct class. An algorithm's recall (i.e., completeness or producer's accuracy) is the number of true positives divided by the sum of true positives and false negatives. This number tells us the probability of a pixel being correctly classified. Both are important because they can indicate whether the class is being overpredicted or underpredicted. Regression problems may often use the mean average error or root mean square error as accuracy metrics.

In some cases, it is not possible to obtain a quantitative error metric for the model. The "true" value may simply not exist, such as for unsupervised clustering ML algorithms. Visual interpretation can be used to evaluate the output of clustering methods to decide whether the algorithm generates meaningful clusters.

It is more common, however, that the true value is simply not known, and so alternative data sources may be used to validate a model. For example, geotagged crowdsourced images can be used to validate the flood extent an ML algorithm generated from satellite imagery (see case study 6.4.1. Flood Extent Mapping).



Interpreting model results

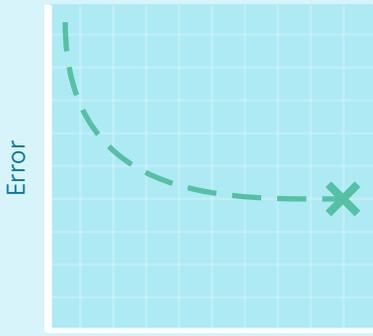
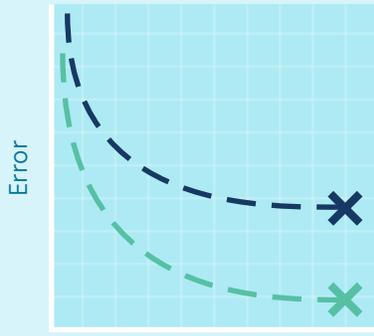
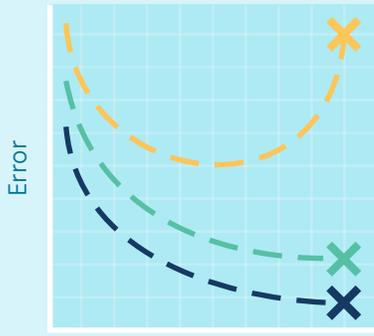
As a user, you can also get an idea of what is happening in the model by comparing the accuracies that are obtained for the training, validation, and testing datasets (see table on page 24). If the training error is high, then consider getting additional data. This could mean obtaining more training samples or, perhaps, a different type of data which is more capable of distinguishing between the different classes. You can also try a different classification algorithm or use the validation set to select the best model parameters.

If the training error is low, but the validation error is high, you could be overfitting the model. Overfitting happens when the model is too complex for the input training data. If limited training data are available, a very complex model might actually be learning which class should be assigned to each individual sample rather than learning the underlying patterns which distinguish the different classes. To avoid overfitting, try getting more training data. You can do this by collecting more reference data from external sources or introducing slight variations into the training data you already have. Deep learning algorithms, for example, may rotate or flip input

samples to easily increase the variation in training data. Another option is to reduce the model complexity by changing the model parameters.

It's important to note that some ML algorithms, especially deep learning ones, do not give us an idea of which input variables are important, or which relationships between variables led to a specific outcome. On the contrary, when using ordinary least squares linear regression or decision trees, for example, it is clear which features best explain a specific output of the model.

If the training and validation errors are low, but the testing error is high, then there may be a bias in the training samples. That is to say that the training samples are not representative of the testing dataset. This may be the case when applying a model which has been developed for one project to a different project. For example, a building detector in the Netherlands may not function well for a city in Africa because the buildings may look quite different. If this is the case, then consider obtaining more representative training data, or start the process from scratch by dividing your new data into new training, validation, and testing sets.

 <p>Iterations</p>	 <p>Iterations</p>	 <p>Iterations</p>
<p>Scenario: High training error</p>	<p>Scenario: Low training error, but high validation error</p>	<p>Scenario: Low training and validation errors, but high testing error</p>
<p>Problem Initial model isn't suitable</p>	<p>Problem Overfitting</p>	<p>Problem Training bias</p>
<p>Possible solution</p> <ul style="list-style-type: none"> • Get more input data (more samples or complementary data) • Change the model or model parameters (for deep learning, train longer) 	<p>Possible solution</p> <ul style="list-style-type: none"> • Obtain more training data • Reduce model complexity 	<p>Possible solution</p> <ul style="list-style-type: none"> • Obtain more representative training data • Retrain model (if applying to a new project) • Sometimes necessary to use an entirely different model.

It should be emphasized that the overview in the table above is a simplification of the process. Although it gives a general overview of the main issues, the possible problems and solutions are, of course, much more nuanced than the table demonstrates. However, as a nonexpert, it is important to remember that the testing dataset which is used to describe the model accuracy should not be used to train the model. Having insight to the accuracies of the training, validation, and testing sets can help understand whether the model is accurate and which steps can be taken to improve it.

4.3 EXPERTISE, TIME, INFRASTRUCTURE AND COSTS

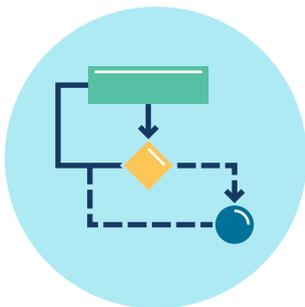
The hardware and software needs of projects using ML algorithms on big data vary widely. A small project or prototype can be envisioned using free software, minimal coding, and WB information technology (IT) infrastructure, but larger and more articulate projects require considerable expertise and IT infrastructure. Projects that require more expertise in coding, parameter tuning, etc., will inevitably incur larger costs and time frames. Several factors may impact the cost of an ML project.



Training dataset: Does the training data already exist? Or does it have to be created manually? How much training data is needed for the algorithm to be trained? In most projects, particularly in areas where the data may be scarce, the creation of a required representative training dataset may be one of the main drivers of cost for the project, as it can involve intense manual labor. In other cases, it may be readily available. This can easily be the most important and expensive part of the project, as the model, and any result that comes from it, is inherently tied to the quality of the data that is input. An old adage goes: “garbage in, garbage out.” In other words, a model is only as good as its input data. Creating a relational database of the input features with their labels will also take time, and depending on several factors, those databases could simply be a file on a computer or a networked, cloud-based item.



Imagery: When using imagery, can you rely on openly available imagery? Or do you need commercial higher resolution imagery? The latter case may involve buying large, expensive swaths of imagery, or at least paying for on-demand access to the imagery to run the algorithm within platforms like GDBX, Google Earth Engine, Descartes Lab, Orbital Insight, or Airbus OneAtlas. Resources such as GloVIS of USGS/NASA and EO Browser of the European Space Agency have been and are instrumental in accessing earth observation data, but they still need to be downloaded.



Algorithm: Do you need a new algorithm, or can an existing one be tweaked and trained to fit your goals? More and more satellite imagery segmentation and recognition may be available out of the box, whether in an open-source format, or for a fee—but newer and more advanced applications may require more extensive work to develop new algorithms or combine existing ones. The time it takes to tune an algorithm’s parameters varies by case.



Processing resources: Depending on the amount of data, the size of the area of interest, the type of data, and the algorithm, the resources necessary to process a project can vary. Some can run on a laptop or desktop computer with a good graphics processing unit (GPU), while others require the storage and computing capacity of a server. Others benefit from deployment in large, pay-per-use cloud computing services.

Resources such as Google Earth Engine have pioneered and fundamentally changed the way that the processing of Earth imagery can be done by employing the power of the cloud, bypassing many time-consuming and expensive steps in data downloading, archiving, preprocessing, and processing, not to mention keeping archives of imagery updated for recurring tasks.

4.4 ETHICS: PRIVACY AND BIAS CONSIDERATIONS

PRIVACY

In terms of privacy, ML poses threats at different levels, first due to the amount and details of the data handled that can be private or high resolution, and also due to the predictive power of the ML algorithm applied to those large amounts of data. For instance, ML can identify individuals better than humans can and can do it at scale. But more unexpectedly, they can also reveal things about people that they may not know, or that their immediate circles may not know.

In everyday life, large amounts of data are potentially mined and used according to contracts and conditions that we enter when using personal devices such as smartphones, with their multitude of sensors, or participating in online activities—be it simply e-mail or using social networks. ML methods are used by companies and organizations to manipulate our user data and provide additional services. For example, our social network feed learns from our activities what to show next and which ads we are most likely to click on, and our favorite online retailer learns from our tastes to offer us other items we may like. In this sense, privacy is a data concern and a sharing concern which simply extends to ML because it uses data and is often

applied online in cloud environments. Specifically in DRM, ML can pose privacy risks as the volume of data increases and the spatial resolution of imagery used, such as drone data, increases. It is easy to detect individuals in drone imagery. In street-level imagery, faces are discernible, as well as potential building attributes that may pose security risks. Again, the discussion here is more about the privacy concerns of acquiring, storing, and sharing personal data than ML, per se. Certainly, suitable privacy guidelines should be followed according to the type of input data utilized. For example, the ethical usage of drones for development applications is discussed in a separate World Bank Group guidance note.

For these reasons, as DRM projects employ ML to create and use data from remote sensing as well as other sources, it is important to note that this data can hold private information and, therefore, should be adequately dealt with. In addition, the concept of privacy varies widely over regions and social groups, so global best practices and standards should always be supplemented according to the specific project in question.



BIAS

All data has biases. All models are incomplete, as they are approximate representations of the real world. Paying attention to these biases is necessary for both improving ML approaches and using their results responsibly. Even significant societal biases like racism, sexism, or economic bias have been shown to affect algorithmic modelling. Especially in the case of DRM, these biases can have important repercussions if they are the basis on which the vulnerability of populations is assessed.

It is important to understand that ML algorithms are not bias-free, because in some cases, like deep learning, they obtain results without human interaction. It is crucial, therefore, to have diverse training datasets and to keep in mind whether the reality being modelled is from a data-poor area with geo-scarce information, and what the connection is with underrepresented and vulnerable populations and development goals. Disasters impact vulnerable groups disproportionately, and any bias involving the information of characteristics of these groups can have a big impact.

To alleviate those issues, algorithmic accountability and algorithmic transparency are two principles that address the degree to which the results of an ML algorithm can be understood. Especially when ML algorithms result in concrete decisions (e.g., insurance rates on houses, the prioritization of investments, or protection measures), it is important that the public can understand why they qualify or do not qualify for a certain subsidy or policy. Similarly, if the driving factors behind ML models are understood, one should understand that by making the results of an algorithm public, they may unintentionally be publicizing underlying factors which are more sensitive from a privacy point of view.

For a very thorough collection of resources on this topic please see this article: [Toward ethical, transparent and fair AI/ML: a critical reading list at](https://medium.com/@eirinimaliaraki/toward-ethical-transparent-and-fair-ai-ml-a-critical-reading-list-d950e70a70ea)

<https://medium.com/@eirinimaliaraki/toward-ethical-transparent-and-fair-ai-ml-a-critical-reading-list-d950e70a70ea>



5. MACHINE LEARNING IN THE COMMONS



5. MACHINE LEARNING IN THE COMMONS

As promoted by the Principles for Digital Development⁴ and the Open Data for Resilience principles,⁵ using open innovation through the use of open standards, open data, and open-source software can greatly benefit sustainable development. Depending on the context and needs of a project, some data may have privacy issues, or the extent to which open-source is used may be different. But, overall, embracing open innovation can greatly increase the use of public resources by avoiding duplication, fostering education, creating new knowledge, and providing opportunities by empowering individuals worldwide with open data and tools. These principles also apply to the development and use of ML algorithms. Throughout the process of ML projects, several aspects can benefit from being documented and shared: the training data, the algorithm, the methodology, and the output data as described in the following sections. They can come together for open ML approaches to support sustainable disaster risk management and development goals.

5.1 OPEN DATA

Open data are data that are technically and legally open and shared in a machine-readable format using open standards, with proper documentation on origin and quality, as well as with clear licenses that allow for reuse of the information. This final point is often neglected, but it can be crucial when it comes to using data in an emergency situation with little time to figure out licensing issues. Open data provide several benefits: data can be created once and be reused many times for multiple purposes, ensuring economy of scale, avoiding duplication, and maximizing the use of resources.

In the context of ML, training data should be shared using open standards so it can be reused by others to train other algorithms. Training is often one of the most costly aspects of putting together an ML project, as it can be tedious and manual to assemble. Therefore, sharing training data can catalyze potential ML applications. For instance, for satellite imagery and labelled data, several standards such as Cloud Optimized GeoTiff⁶ (COG), and SpatioTemporal Asset Catalog⁷ (STAC), as well as repositories such as MLHub (<https://www.mlhub.earth>) are being developed to allow sharing and interoperability of tools

and training datasets across projects and the industry. **On the other end of the process, the output—the data generated by the ML algorithm should also be shared—as it can be critical for many development decisions.**

For instance, base exposure datasets, such as building infrastructure and roads, can be used by many sectors for many different decisions. In this context, it is again important to use best practices to share the output data using open standards, documentation, and proper licensing.

5.2 OPEN-SOURCE SOFTWARE AND DOCUMENTED METHODOLOGY

Given the amount of data handled, the time needed to train algorithms, and the computing power required, there is a natural trend of centralization of data and algorithms in the cloud under proprietary licenses to be used as a platform or software as a service for a fee when it is run and used.

Although this approach can be economical and more practical for the end users, it can limit the potential of those tools for development, innovation, and education. Even if deployed in production as software as

⁴<https://digitalprinciples.org/>

⁵<https://opendri.org/resource/opendri-policy-note-principles/>

⁶<https://trac.osgeo.org/gdal/wiki/CloudOptimizedGeoTIFF>

⁷<https://github.com/radiantearth/stac-spec>

a platform, some domains of ML such as computer vision have a tradition of open sourcing the code used in order to foster sharing of knowledge and increase innovation. Similar to open data, open-source allows for economy of scale by enabling many computer programmers and scientists to collaborate on the same code and improve the same code together. This collaboration will also improve the quality of the code, as more people are checking it and running it. It is key to develop open-source tools where possible and invest in software as a public good,⁸ especially where economy of scale can be achieved across an institution or institutions. Contributions and support to open-source software can materialize in different ways—not only code, but also documentation, user and developer events, user design, and others, as shown with previous example such as GeoNode.⁹ Some examples of open-source software for ML are TensorFlow and remote sensing-specific tools like Mapbox’s RoboSat.

5.3 CROWDSOURCING AND CAPACITY BUILDING

In the last decade or so, there has been a huge growth of volunteer and networked communities of individuals mapping data together. In general, these are called collaborative or crowdsourced maps, and they have created everything from OpenStreetMap to satellite image feature recognition in humanitarian efforts. These networks of humans have been brought together and allowed to work collaboratively by ever-evolving software that enables seamless work in crowds that can be formed by individuals all over the world connected by the Internet.

This is particularly important in the context of the enormous growth and ubiquity of ML methods in computer science, and those that are increasingly applied to disaster risk management.

For the applications described in this note to be successful, ML algorithms will need more and more “label” data so they can be supervised and the accuracy of their results validated. This is an area where it will be key to hybridize the work of humans and computers so that their efforts can be optimized to achieve the maximum efficacy on a project. Crowdsourcing platforms like OSM already have provided over a decade of experience in leveraging large networks of people to manually add features and labels to maps where computers could not do so.

There is an obvious link between harnessing the power of the crowd to provide much-needed training data for ML algorithms. For example, Google has been using re-CAPTCHA to train image recognition algorithms. Involving the crowd to provide annotated data can help provide a large amount of information to help train accurate models. Especially when involving people from various parts of the globe, there is a possibility to add local knowledge and avoid biases such as those described above.

Beyond generating training data, there are also a number of projects looking at a hybrid approach, where the algorithm’s output solely aims to aid the human. Some of those examples relevant to development issues include AI-assisted road tracing by Facebook,¹⁰ where the ML algorithm output predicted roads, but humans will ensure their accuracy and topology before entering them in OpenStreetmap. Then this dataset can be used for many types of accessibility studies. Similarly, DevSeed has set up a similar electric grid mapping system¹¹ for the World Bank, claiming that it made the human mapper 33 times more efficient. Overall, this approach can ensure high data quality while making the human’s tasks less tedious.

⁸<https://digitalprinciples.org/resource/howto-calculate-total-cost-enterprise-software/>

⁹<https://opendri.org/resource/opendri-geonode-a-case-study-for-institutional-investments-in-open-source/>

¹⁰https://wiki.openstreetmap.org/wiki/Al-Assisted_Road_Tracing

¹¹<https://devseed.com/ml-grid-docs/>

However, it is important to avoid a situation where all advanced AI knowledge, software, and data are centralized with a few large Silicon Valley companies. Education and capacity building should be stimulated. Building on the Open Data for Resilience principles, when developing a DRM project, it is also important to consider new ways of involving local universities and knowledge centers. Increased human capital and access to computing resources will help pave the way for new mapping techniques and significant advancements in the disaster risk management area. That human capital, together with ML algorithms, will certainly pave the way for the future of mapping in the disaster risk management arena.

Of particular concern to the World Bank GFDRR is data openness and transparency, capacity building, and the role of crowdsourcing, such as the OpenStreetMap community.

In terms of crowdsourcing, there is tremendous potential in using the networks and tools already established to go from mapping to training and testing ML algorithms. While this feature has not been used widely to date, we believe that it could provide a future avenue for generating large ML algorithm training and testing datasets. The OSM map-filling, capacity building, and networking events known as “mapathons” could be envisioned as “trainathons”—the difference being that the final output of the training and validation of an ML algorithm could be to fill the map of an area or label it with much higher speed and scale. The OSM ecosystem’s existing tools that allow nested validation by expert mappers, and also the easy tiling/prioritization of mapping areas like in HOT OSM’s ID editor, would already provide the most important data needed for a successful project using ML.

At the same time, this conversation revolving around training and testing data would allow local communities to build their capacity and have a say in and ownership over their own data and results of an ML algorithm.

5.4 MACHINE LEARNING FOR SUSTAINABLE DEVELOPMENT: FROM USE CASES TO STANDARDIZED TRAINING DATA

Putting together all these components in an open and interoperable way creates potential for networked global data systems using ML algorithms to provide enormous societal benefits in disaster risk management as well as, more broadly, the Sustainable Development Goals (SDGs).

Concretely working toward the creation of an open framework encompassing the different use cases, the training data required, and the algorithm to be trained, all following open standards, will provide the structure to scale ML efforts across geography and sectors. It will also provide transparency and opportunities for capacity building, crowdsourcing, and knowledge sharing. GFDRR is joining efforts such as [MLHub](#) to create a network of distributed repositories that provide access points to openly share ML training data, models, and standards. This also supports the key role that open data and software, collaborative networks, crowdsourcing, and capacity building have to play together in the future of ML algorithms to support DRM and SDGs alike.

6. CASE STUDIES IN DISASTER RISK MANAGEMENT



6. CASE STUDIES IN DISASTER RISK MANAGEMENT

The following case studies fall into four categories: **6.1** physical exposure and vulnerability, **6.2** social exposure and vulnerability, **6.3** risk mapping and damage prediction, and **6.4** post-disaster event mapping and damage assessment.

These case studies were selected as they provide an overview of how ML can support various aspects of DRM. They represent different geographical regions, various input datasets and units of analysis, and various ML algorithms. An overview is provided of the key characteristics of each case study: the objective, input data and reference data used, scale of analysis, the algorithm used, who performed the analysis, results and lessons learned

and where to find more information. This selection is not comprehensive and will be updated regularly, as this is a booming field with many new applications of ML being developed on a monthly basis—new upcoming applications involve prioritizing building inspection, social media mining for response awareness, monitoring of rebuilding and recovery activities, support to insurance claims, and many others.



6.1 PHYSICAL EXPOSURE AND VULNERABILITY

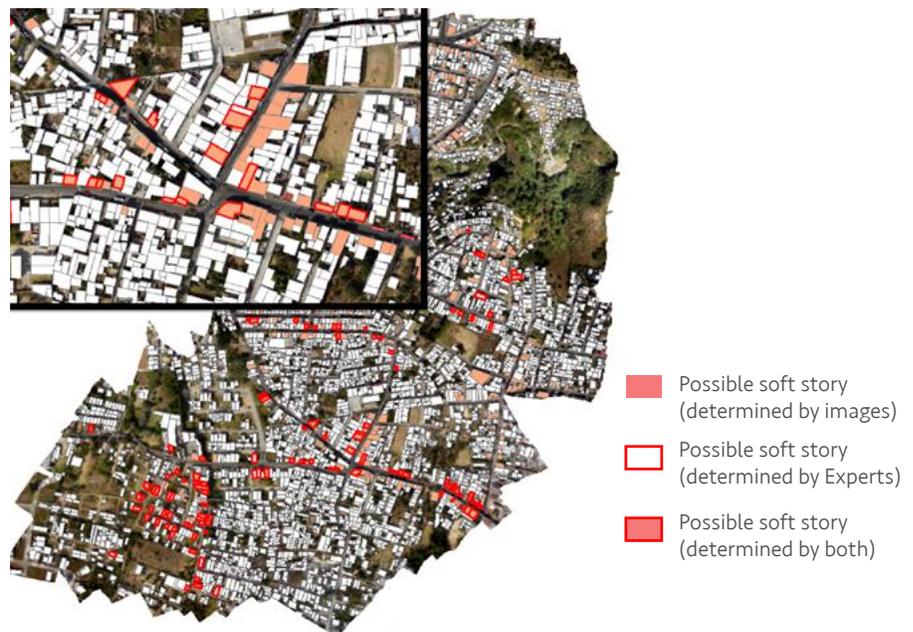
6.1.1 Guatemala City building earthquake vulnerability

Detecting seismic vulnerability in urban areas is critical. Identifying high-risk buildings can save lives and help prioritize retrofitting investments. However, sending large teams of surveyors into the field is time consuming and expensive. Instead, this case study leverages imagery from satellites and drones, and street-view images from 360° street cameras to identify homes that are a high risk for collapse during an earthquake. Digital elevation models from satellite imagery helped identify buildings located on steep slopes, which are at higher risk for mudslides.

A combination of satellite and drone imagery helped identify rooftop material, suggesting underlying construction techniques which are more vulnerable to seismic activities. The availability of street-view imagery is unique, as it can be used to identify soft-story constructions which are vulnerable to seismic activities.

This case study is a good example of how different physical factors of vulnerability can be extracted from various data sources and the unique capabilities of street-view imagery. The deep learning algorithm trained on the street-view imagery caught 85% of the buildings which were flagged by expert engineers as vulnerable.

Underlying DRM goal	Quickly identify seismically vulnerable “soft-story” homes
Which input data were used	<ul style="list-style-type: none"> • Drone imagery (eBee, RGB, 4 cm) • Point cloud elevation data • Street-view imagery (Trimble MX, (30 megapixel)
Reference data	OpenStreetMap road layer
Unit of analysis	Pixel/object (building)
Scale of analysis	Neighborhood-level (three neighborhoods of approximately 10 km ² in Guatemala)
Which algorithm was used	Deep learning
Who completed the analysis	GOST/GSURR
Results and lessons learned	<ul style="list-style-type: none"> • This method screens a neighborhood of 5,000 homes and is able to identify some 500 that need further inspection and possible retrofitting/strengthening. • Of the “soft-story” buildings flagged by engineers (who viewed them from the outside) this method caught 85% of them. • This detailed databases as potentials for input into exposure databases, locating and prioritizing retrofitting/housing upgrading projects. • Automatic detection of large first-floor openings was done with data collected by the team—but to scale up, Google Street View and/or Mapillary should be considered. • Satellite imagery was also explored to see if 50-30cm could be used to measure the height of buildings. NTT was hired and delivered a layer that was good but tended to lump households together, especially in dense neighborhoods.

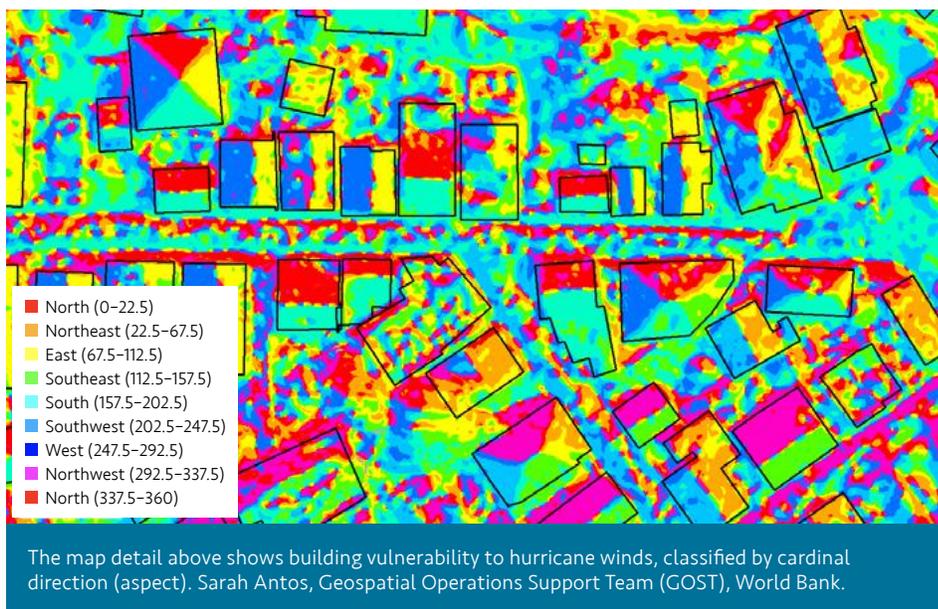


The map above illustrates the “Rapid Housing Quality Assessment”, done by Sarah Antos, Geospatial Operations Support Team (GOST), World Bank

6.1.2 St. Lucia building hurricane vulnerability

What kind of damage would Saint Lucia experience if it was hit by a Category 5 storm? Using a recent detailed damage assessment conducted in neighboring Dominica, physical characteristics such as roof material and the shape and size of buildings were used to predict the vulnerability of individual structures in Saint Lucia.

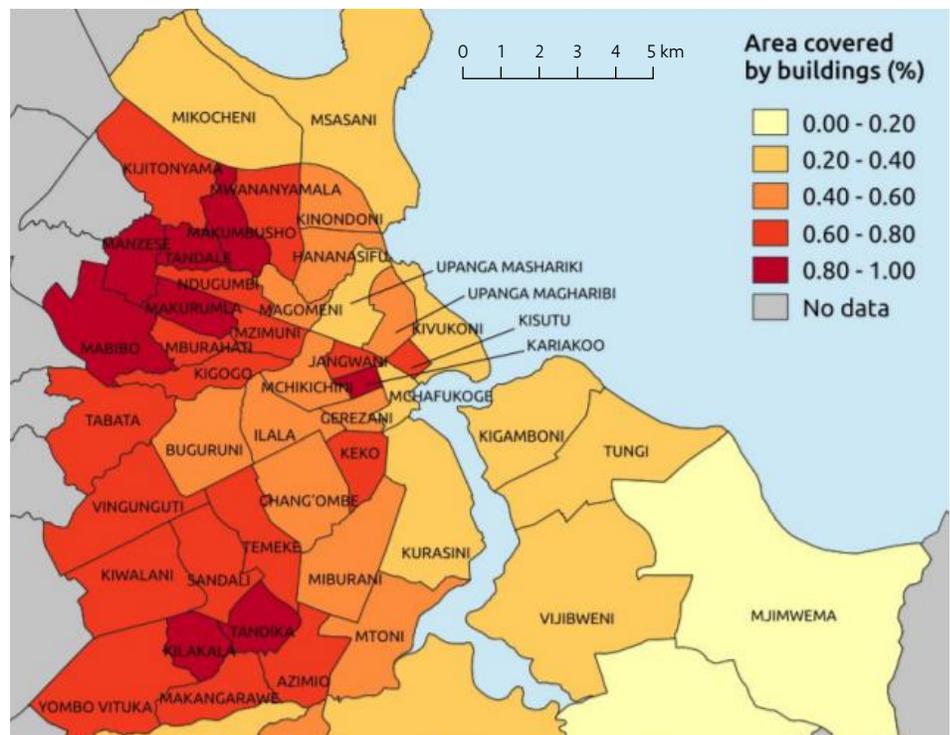
Underlying DRM goal	Estimate hurricane rooftop vulnerability in small island states (Caribbean)
Which input data were used	<ul style="list-style-type: none"> • Drone imagery (eBee, RGB, 4 cm) • Point cloud elevation data • Street-view imagery (Trimble MX, (30 megapixel)
Reference data	OpenStreetMap building footprints that were downloaded from the Charmin geonode
Unit of analysis	Pixel/object (building)
Scale of analysis	City-level (three cities of approximately 9 km ² in Saint Lucia)
Which algorithm was used	Conditional random field model—several python libraries combined with MpGlue
Who completed the analysis	GOST/GSURR
Results and lessons learned	<ul style="list-style-type: none"> • Using the variables (and combination of them) that most powerfully predicted damage in Dominica, each structure in Saint Lucia was given an estimate of destruction. For example, you can expect a general 40% damage; however, if the roof is smaller and has only two panels (gables) you can expect more than 40%. • Volume, roof shape, and roof type were all influential. Large, highly pitched roofs with PVF2-coated metal sheeting tended to do the best. • Algorithm predicted roof shape (hip vs. gable) more easily than material, due to the three-band drone camera. • We are now working to add valuation information and general “quality” index, such as “rustiness.”



6.1.3 Monitoring urban growth through floor space index

Regular, cloud-free satellite images combined with ML algorithms can be used to monitor horizontal and vertical urban growth. This study uses crowdsourced building footprints and height information to train an ML model to be used for urban monitoring in Dar es Salaam.

Underlying DRM goal	Urban growth monitoring, focussing on built-up area and building height
Which input data were used	<ul style="list-style-type: none"> Satellite imagery (RGB, 3.7 m) Digital surface model (DSM) extracted from stereoscopic satellite imagery (0.8 m)
Reference data	OpenStreetMap building footprints and height attribute collected during the Ramani Huria project
Unit of analysis	Pixel
Scale of analysis	City-level (5,280 km ² in Dar es Salaam, Tanzania)
Which algorithm was used	Deep learning, convolutional neural networks
Who completed the analysis	Planet
Results and lessons learned	<ul style="list-style-type: none"> The study shows how to combine OSM reference data and machine learning methods. Building footprints were extracted to an accuracy of 77%; the correct number of floors predicted for 23% of the buildings. Difficulties were caused by densely built (informal) areas. Results would likely improve with higher resolution imagery.
More information	Executive Summary: Monitoring Urban Change with Satellite Imagery and Analytics pp. 36, 37, 40, 41, 43-46.



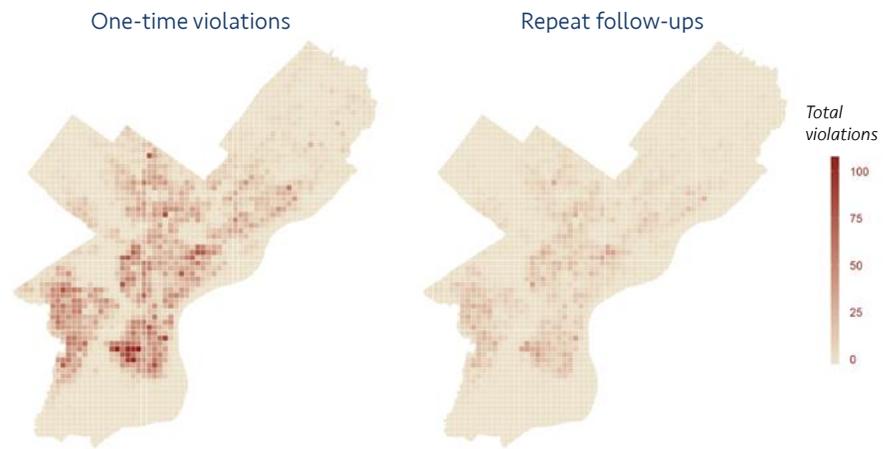
Building to Non-Building Ratio—Inner-City Wards of Dar es Salaam, August 2017
Figure 12 in WB report

6.1.4 Targeting high-risk buildings for building inspection

Building inspection is an important measure to mitigate the risks of fire. However, as cities grow, it becomes increasingly difficult to prioritize which buildings should be inspected. Some emerging methods combine geospatial data and building attributes to determine which buildings present the greatest risk.

An example by Azavea focuses on the likelihood of a building which failed a past violation to fail again. The model makes use of open data from the [OpenDataPhilly portal](#), which provided information on more than 55,000 building inspections in more than 25,500 locations. Various features of the building inspections were considered, such as the duration between inspections, type of violation, location variables, the total number of violations, building vacancy, and tax delinquency. A feature selection was performed to remove variables which were not relevant for predicting a repeat violation. The model results indicate that repeated violations of building inspections could be predicted with an accuracy above 74%.

Underlying DRM goal	Building regulation violations
Which input data were used	<ul style="list-style-type: none"> • Cases and descriptions of previous building violations • Locations • Building vacancy • Tax delinquency
Reference data	Building inspection reports from the City of Philadelphia’s Department of Licenses and Inspections.
Unit of analysis	Building
Scale of analysis	City-level (Philadelphia, USA)
Which algorithm was used	Gradient boosting and random forests
Who completed the analysis	Azavea
Results and lessons learned	<ul style="list-style-type: none"> • The model was able to predict repeated building violations with an accuracy of 74%. • The results can help building inspectors allocate resources effectively by targeting high-risk buildings.
More information	Predicting Building Inspections Predicting Building Code Compliance with Machine Learning Models



Follow-up Building Inspection Results
 Relative densities of buildings that passed after initially failing an inspection (left) and those that failed (right) Source: Azavea, Data: Philadelphia Department of Licenses and Inspections

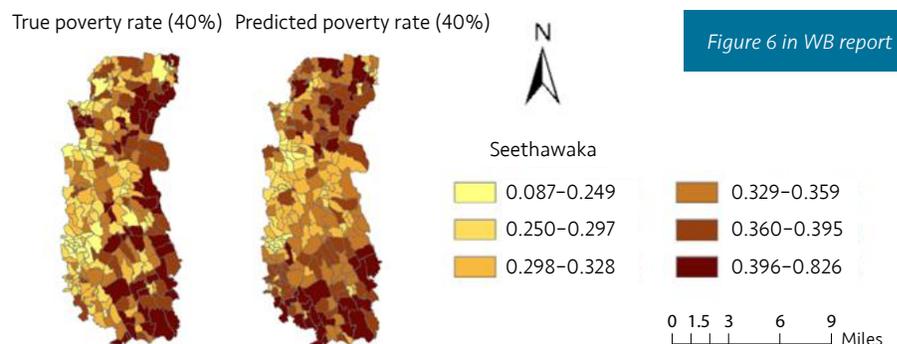
6.2 SOCIAL EXPOSURE AND VULNERABILITY

6.2.1 Sri Lanka poverty mapping

Poverty data are in scarce supply and difficult to collect. This study investigates the suitability of features derived from very high-resolution satellite imagery to estimate poverty at a local level in Sri Lanka, allowing these estimations to be extrapolated to areas not covered by surveys. A unique partnership with OSM provided access to a large amount of labelled data to support the ML algorithm.

A large number of object- and pixel-based features describing agricultural land, cars, building density and vegetation, shadows, road and transportation networks, roof types, and textural/spectral characteristics were extracted from the imagery. A linear regression was established to determine the relationship between these features and poverty levels taken from census data.

Underlying DRM goal	Estimate poverty levels
Which input data were used	<ul style="list-style-type: none"> Satellite imagery (RGB, < 0.5 m) <ul style="list-style-type: none"> Object-based features <ul style="list-style-type: none"> Number of buildings Number of cars Fraction roads paved Shadow pixels (building height) Crop type/extent Roof type Pixel-based features <ul style="list-style-type: none"> Vegetation index PanTex (settlement density) Texture (HoG, LBP, Line Support Region, Gabor filter, Fourier transform, SURF)
Reference data	Two poverty lines (10th and 40th percentiles of the national per capita consumption distribution) which were obtained from 2011 census data
Unit of analysis	Pixel/object (administrative unit)
Scale of analysis	Regional (3,500 km ² covering 1,250 administrative units in Sri Lanka)
Which algorithm was used	<ul style="list-style-type: none"> Deep learning (convolutional neural networks) to calculate percentage of built-up area, number of cars, shadow pixels, and crop type for each administrative unit Support vector machines and visual identification to obtain information regarding roof type, paved and unpaved roads, and railroads
Who completed the analysis	WB poverty team working with Orbital Insight, LAND INFO Worldwide Mapping, LLC, and the George Washington University Department of Geography
Results and lessons learned	<ul style="list-style-type: none"> Analysis can explain 60–61% of the variation in a small area (compared to 15% when using night lights analysis). Building density, built-up area, and shadows were some of the most influential features describing variations in poverty. Normalized error rates of 0.25–0.5 of poverty rates when applying the model to geographically adjacent areas. Project cost \$90,000 total.
More information	Graesser J B, Cheriyyadat A M, Vatsavai R, Chandola V, and Bright E A. 2012. Image Based Characterisation of Formal and Informal Neighborhoods in an Urban Landscape. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5.



6.2.2 Informal settlement mapping

In 2012, a peer-reviewed paper by Graesser et al. mapped the informal settlements in four major cities, using an automated ML algorithm to classify satellite imagery. According to the authors, in remote sensing imagery, informal settlements share unique spatial characteristics that distinguish them from other types of structures like industrial, commercial, and formal residential areas. After a thorough literature review of remote sensing methods that have been used for similar objectives, the authors used several low-level image features at multiple scales to characterize local neighborhoods, separated based on a series of spatial, structural, and contextual features.

Graesser et al. outlined how formal and informal neighborhoods can be visibly separated given enough spatial resolution of the imagery used. Informal settlements often share unique spatial, structural, and contextual features that separate them from other types of urban neighborhoods. These characteristics can include:

- A high heterogeneity in building orientation (most buildings aren't "neatly" oriented along a planned space [e.g., a road])
- A high variance in building materials used and density of the structures (as opposed to formal settlements, where there would be more homogeneity of these features in a neighborhood)
- Small building size (as opposed to larger buildings with more stories in formal settlements)
- Irregular and narrow streets (as opposed to wider and straighter planned streets)
- Informal neighborhoods that are often closer to hazardous zones like landfills, airports, railroads, and steeper slopes

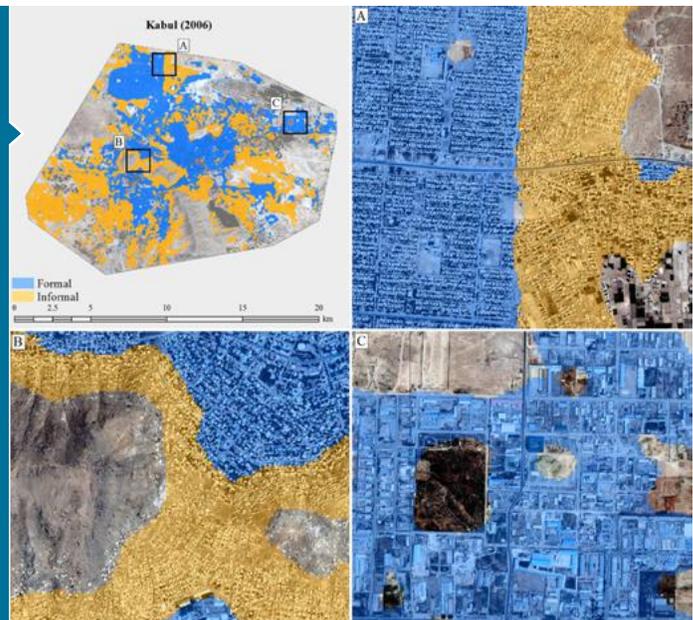
Underlying DRM goal	Identification of informal settlements
Which input data were used	<ul style="list-style-type: none"> • Satellite imagery (RGB, < 0.5 m) <ul style="list-style-type: none"> ◦ Pixel-based features <ul style="list-style-type: none"> ▪ Vegetation indices ▪ GLCM PanTex (settlement density) ▪ Texture (HoG, lacunarity, linear feature distribution, line support region, SIFT, yextons)
Reference data	Manual labelling of imagery
Unit of analysis	Pixel
Scale of analysis	City (74 km ² of Kandahar, Afghanistan; 203 km ² of La Paz, Bolivia; 220 km ² of Kabul, Afghanistan; 348 km ² of Caracas, Venezuela)
Which algorithm was used	Decision trees
Who completed the analysis	Graesser J B, Cheriadat A M, Vatsavai R, Chandola V, and Bright E A of Oak Ridge National Laboratory
Results and lessons learned	<ul style="list-style-type: none"> • Texture features in submeter satellite imagery were found to be suitable for distinguishing formal vs. informal areas in cities. • ML algorithm had an accuracy of 85–92% for the four cities. • Authors suggest that methods which take multiple neighboring pixels into account may improve results. • The study relates social vulnerability to the physical appearance and arrangement of buildings and roads; this will depend on local context, and one should take care when applying the models to other areas.
More information	Graesser J B, Cheriadat A M, Vatsavai R, Chandola V, and Bright E A. 2012. Image Based Characterisation of Formal and Informal Neighborhoods in an Urban Landscape. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5.

Classification results for Kabul. The results have been smoothed using an 11 x 11 majority filter.

(a) Formal and informal (Type I) residential

(b) Informal (Type II) residential built on slopes

(c) Nonresidential



6.2.3 Stanford poverty study

Poverty mapping based on census data is often expensive and difficult to collect at a large spatial scale and update frequently. This study aims to use remote sensing to predict the ratio of households above the poverty line in Uganda.

This study shows an alternative strategy of how to use deep learning when limited training samples are available. First, a deep learning model, which learned image features from an object detection challenge (ImageNet) is used. Due to the lack of survey data, the researchers use night light data (a proxy for economic development) to train the model to learn relevant features, which are then used to form a logistic regression model predicting poverty levels.

Underlying DRM goal	Poverty mapping
Which input data were used	<ul style="list-style-type: none"> • Deep learning model trained on ImageNet • NOAA nightlights imagery • Google Maps imagery
Reference data	<ul style="list-style-type: none"> • Night-time lights • Governmental household surveys
Unit of analysis	Pixel (1 km x 1 km grid), object (districts)
Scale of analysis	National (Uganda)
Which algorithm was used	Deep learning (fully convolutional neural network) and logistic regression classifier
Who completed the analysis	Stanford University
Results and lessons learned	<ul style="list-style-type: none"> • Proposed method can predict poverty levels with 72% accuracy. This is comparable to results of the logistic regression when using survey-based features to predict the surveyed poverty levels. • This shows how a proxy dataset can be used to develop a machine learning model when not enough reference data are available.
More information	Transfer Learning from Deep Features for Remote Sensing and Poverty Mapping Stanford researchers use dark of night and machine learning to shed light on global poverty

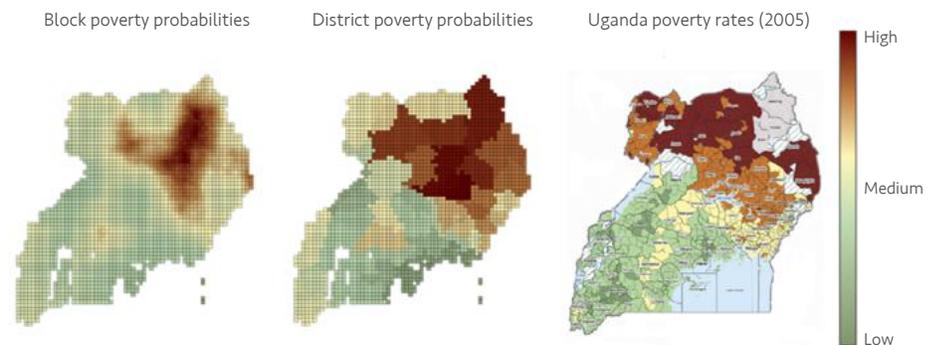


Figure 3 from the Arxiv paper

Left: Predicted poverty probabilities at a fine-grained 10 km x 10 km block level. Middle: Predicted poverty probabilities aggregated at the district level. Right: 2005 survey results for comparison (World Resources Institute 2009)

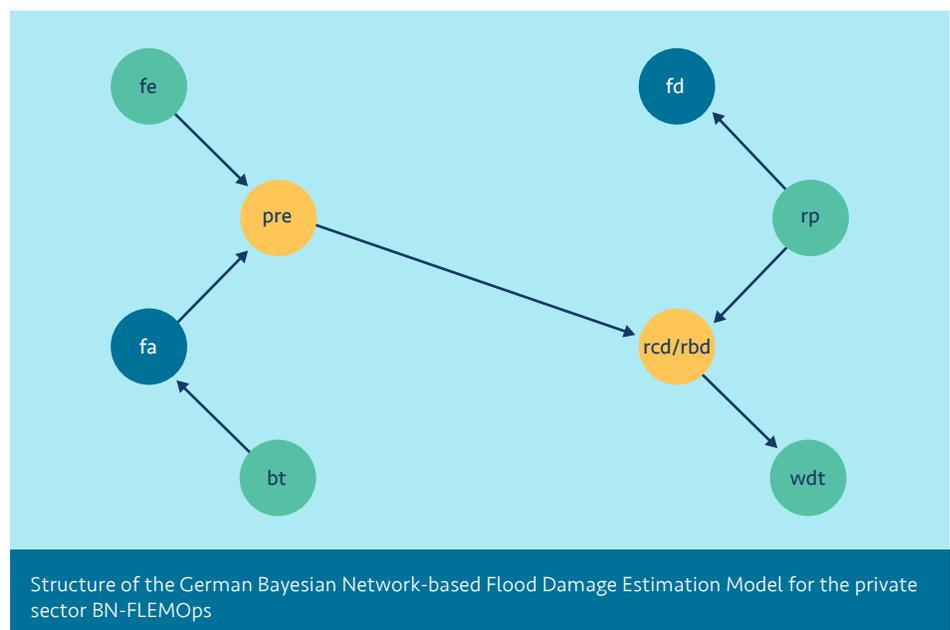
6.3 RISK MAPPING AND DAMAGE PREDICTION

6.3.1 Flood damage prediction

Many flood damage assessment models utilize water depth to calculate damage curves based on specific location and flood conditions. Applying the same curves to different situations therefore often produces unreliable results. This project researches how the inclusion of additional variables can be used to improve the transferability of flood damage prediction models.

Bayesian networks and regression random forests were constructed to relate the relative building damage or relative content damage reported by surveyed households to various input features. Results show that models which are trained using heterogeneous data (i.e., flood events with various characteristics) have a higher performance. The authors emphasize the importance of acquiring a heterogeneous training set for flood damage models, including a variety of flood events, geographical locations, and asset characteristics.

Underlying DRM goal	Flood damage assessment
Which input data were used	<ul style="list-style-type: none"> • Water depth • Building type • Building footprint area • Floor area for living • Building age • Basement • Household size • Flow velocity • Flood duration • Return period • Flood experience • Precautionary measures
Reference data	Relative building damage and relative content damage from field surveys
Unit of analysis	Tabular (survey data)
Scale of analysis	Regional (a flood event in the Netherlands in 1993 and six flood events in Germany between 2002 and 2013)
Which algorithm was used	Bayesian networks and random forests (regression)
Who completed the analysis	Deltares, GFZ German Research Centre for Geosciences
Results and lessons learned	<ul style="list-style-type: none"> • Updating an ML algorithm with data from a different country improves the model's performance on flood events from that country. • The collection of training data from various flood events and regions may be more effective than a large amount of information from a single event.
More information	Regional and Temporal Transferability of Multivariable Flood Damage Models



6.3.2 Machine learning-powered seismic resilience for San Francisco

Modelling structural damage from earthquakes (as with other hazards) is challenging due to the number of factors which influence the process. A proprietary algorithm developed by OneConcern models seismic resilience by predicting the structural damage resulting from earthquakes. It leverages various data sources such as earthquake shaking parameters, soil and seismic hazard characteristics, multiple building characteristics, and real-time field input to estimate the impact of earthquakes. Data from previous earthquakes are used to train the ML models, which are optimized using a unique performance measure to ensure a better estimation of higher damage to buildings. Techniques such as geographical hold-out, event hold-out, and randomized hold-out are used to further improve the model's performance.

OneConcern also focuses on using the developed model to provide real-time and on-demand situational awareness right before, during, and immediately after a seismic event. The damage predictions are made at a census block-level resolution, thus visualizing detailed localized data for seismic hazards throughout the city of San Francisco while maintaining the anonymity of the individual blocks within the city. This also enables the risks and vulnerability data to be democratized by sharing it with local communities and volunteers.

Underlying DRM goal	Earthquake structural damage modelling
Which input data were used	<ul style="list-style-type: none"> • Seismic shaking data for the earthquake of interest • Soil characteristics • Seismic hazard parameters • Building characteristics like material, number of stories, area, etc.
Reference data	Historical earthquake damage data from multiple events
Unit of analysis	Tabular (survey data)
Scale of analysis	City block-level
Which algorithm was used	Proprietary algorithm
Who completed the analysis	OneConcern, Inc.
Results and lessons learned	<ul style="list-style-type: none"> • Making use of data streams from multiple sources and at multiple resolutions can gain a higher training accuracy. • It is important to use diverse data sources to ensure generalizability of the algorithms. • The inclusion of localized data captures effects which are generally not identified through generic methods.

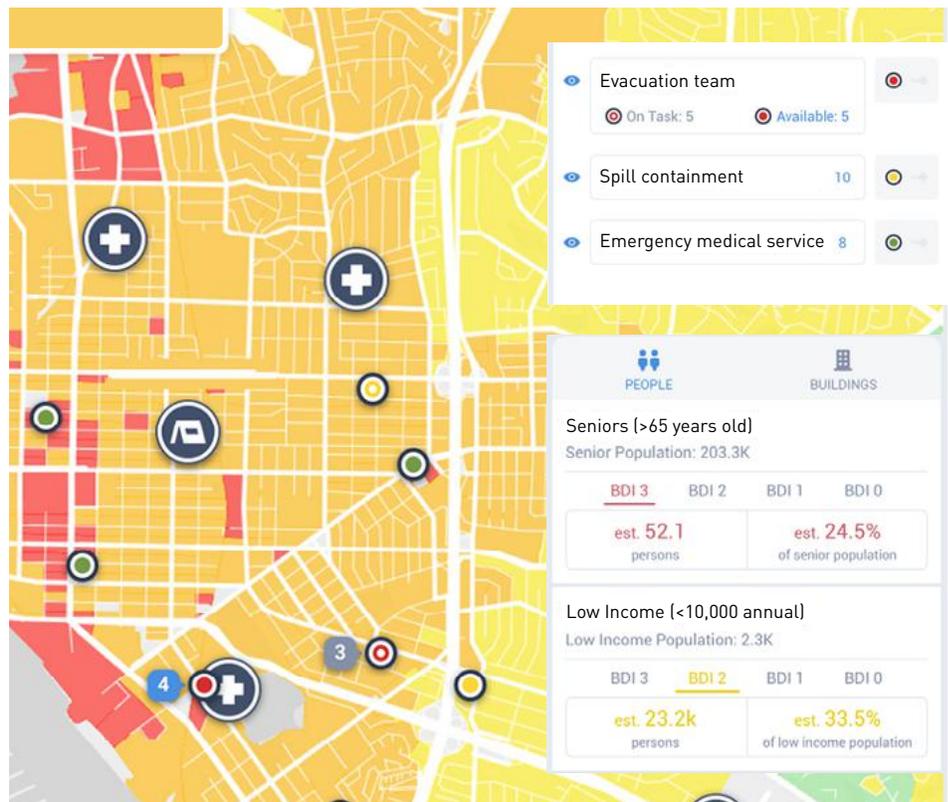
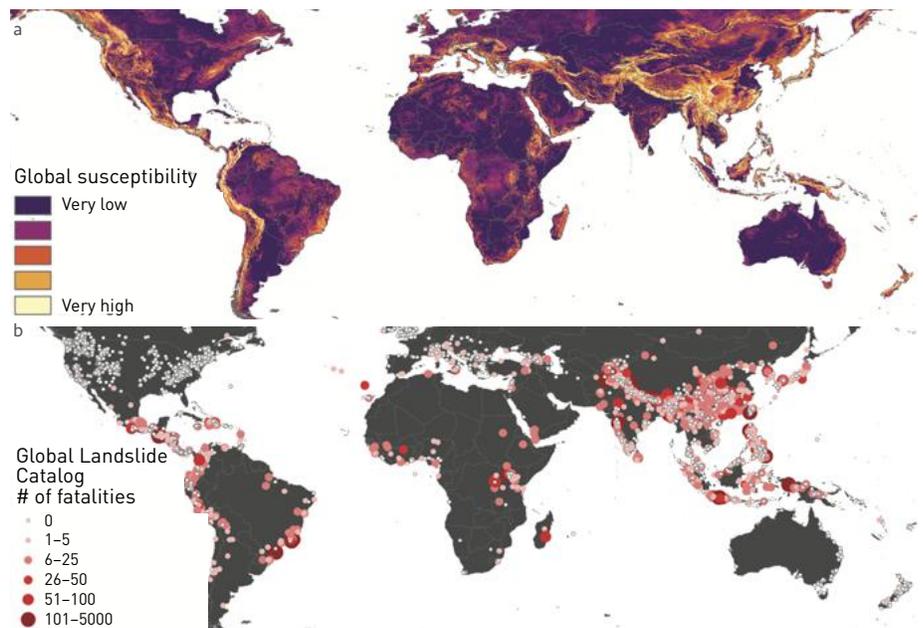


Figure from the original study, available at: <https://medium.com/@oneconcerninc/2018-the-dawn-of-benevolent-intelligence-263c6bd1a63>

6.3.3 Real-time global landslide hazard mapping

The Landslide Hazard Assessment for Situational Awareness (LHASA) provides landslide hazard data in real time. An algorithm was trained which links landslide susceptibility factors (slope, geology, road networks, fault zones, and forest loss) to historical landslide events. This model is applied to precipitation data from the Global Precipitation Measurement (GPM) mission at three-hour intervals. When the rainfall for a given region is extremely high for that region, the landslide susceptibility map is consulted. If a region is also classified as being highly susceptible to a landslide, a nowcast warning is issued. Thus, LHASA provides near-real-time situational awareness of landslide risk on a global scale, presented in an open-source framework.

Underlying DRM goal	Landslide hazard mapping
Which input data were used	<ul style="list-style-type: none"> • Elevation • Faults and geologic regions • Roads • Forest cover • Rainfall
Reference data	Global Landslide Catalog
Unit of analysis	0.1°
Scale of analysis	Global (between 50°N and 50°S)
Which algorithm was used	Decision tree
Who completed the analysis	NASA
Results and lessons learned	<ul style="list-style-type: none"> • The model would have issued a nowcast for historical landslide events with a false positive rate below 3% and true positive rate of up to 60%. • Lack of historical data and locational accuracy of historical landslide events make it challenging to train a good model. The Cooperative Open Online Landslide Repository was launched to obtain additional reference data through citizen science.
More information	<p>NASA landslide map estimates risk in real time</p> <p>Satellite Based Assessment of Rainfall Triggered Landslide Hazard for Situational Awareness</p>



(a) Global landslide susceptibility map computed using slope, geology, fault zones, road networks, and forest loss (Stanley and Kirschbaum, 2017); (b) Global Landslide Catalog (2007–2016) showing the distribution of landslide fatalities (Kirschbaum et al., 2015)

6.3.4 Wildfire prediction

Two high school students invented a device to predict the probability of a forest fire occurring. The device is placed in the forest and can take real-time photos which are uploaded to SensorInsight to enable real-time visualization. Deep learning algorithms are used to analyze the images and predict the amount of dead fuel present in the sensor's area. This information is combined with local weather data to predict the possibility of a fire.

This study is based on a relatively small sample size and will likely require extensive validation with more substantial reference data. Despite these factors, it is a very unique case study as it showcases a grassroots solution and how to combine ML algorithms to obtain real-time risk predictions. The Smart Wildfire Sensor they devised is being further developed and tested with Cal Fire in three counties in California.

Underlying DRM goal	Real-time wildfire prediction
Which input data were used	<ul style="list-style-type: none">• Weather data<ul style="list-style-type: none">◦ Humidity◦ Temperature◦ Gas◦ Carbon monoxide/dioxide◦ Wind• Images
Reference data	Approximately 100 randomly sampled images of grass and shrubs from Google Images
Unit of analysis	Point (locations of sensors placed in forests in California)
Scale of analysis	Regional (selected forests in California)
Which algorithm was used	Deep learning
Who completed the analysis	Cal Fire and Monta Vista High School
Results and lessons learned	<ul style="list-style-type: none">• Classifies images of grasses and shrubs into 14 classes indicating various forest fire risk levels with 89% accuracy.• Model will likely require more extensive validation.• Real-time, grassroots approach of using ML algorithm for DRM.
More information	Fighting fire with machine learning: two students use TensorFlow to predict wildfires



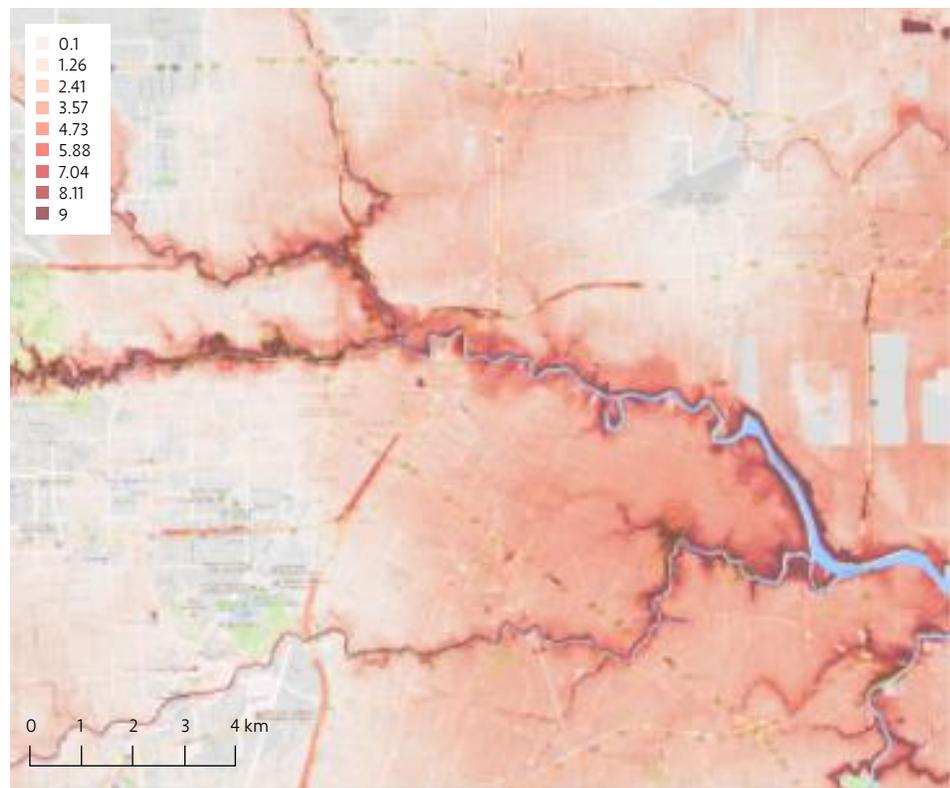
Image from the original study, available at: <https://www.blog.google/technology/ai/fighting-fire-machine-learning-two-students-use-tensorflow-predict-wildfires/>

6.4 POST-DISASTER EVENT MAPPING AND DAMAGE ASSESSMENT

6.4.1 Flood extent mapping

Orbital insight developed a project in 2017 in which they used Synthetic Aperture Radar (SAR) as an input for an image classification algorithm that allowed the categorization of at-risk areas for flooding in Houston, Texas, U.S.A. A combination of optical and SAR imagery (which is capable of “looking” through clouds) helped identify the flooding extent. Digital elevation models (DEMs) allowed natural watersheds to be delimited, and crowdsourced geotagged images were used to confirm the flood extents.

Underlying DRM goal	Flood extent mapping
Which input data were used	<ul style="list-style-type: none"> • Optical satellite imagery • SAR imagery (through clouds) • Digital elevation models (DEMs)
Reference data	Crowdsourced, geotagged images
Unit of analysis	Pixel
Scale of analysis	Hurricane Harvey flood event
Which algorithm was used	Deep learning
Who completed the analysis	Orbital insight
Results and lessons learned	<ul style="list-style-type: none"> • Combining various types of large-scale spatial data helped estimate flood extent. • Crowdsourced, geotagged imagery can help verify flooding in accuracy analysis.
More information	<p>Understanding the Extent of Flooding in Houston from Hurricane Harvey</p> <p>How Orbital Insight Measured Hurricane Harvey’s Flooding through the Clouds</p>



Actual flood maps after applying Orbital Insight’s geospatial interpolation across observation points and DEM (Source: Orbital Insight, Google Street Map)

6.4.2 Cyclone damage assessment

The World Bank and UAViators collected UAV images after Cyclone Pam hit Vanuatu in 2015. High detail and the ability to collect data under cloud cover were advantages of using imagery from UAVs rather than satellites. At the time, volunteers from Humanitarian Open Street Map (HOT) and the Digital Humanitarian Network annotated the damage in the images.

Since then, the images and reference data have also been used to develop ML algorithms. Artificial Intelligence for Digital Response (AIDR) is an open platform combining crowdsourcing and ML to interpret social media data in disaster situations. A similar pipeline was developed for Cyclone Pam data when MicroMappers organized volunteers to identify and demarcate various levels of damage to buildings. These were used to train a deep learning algorithm (Nazr-CNN) to first recognize buildings and then identify damage levels. The study indicates a need for additional training samples in order to improve the transferability of the model.

Underlying DRM goal	Damage assessment
Which input data were used	UAV optical imagery
Reference data	Crowdsourced annotation of images
Unit of analysis	Pixel
Scale of analysis	Regional (Cyclone Pam, Vanuatu, 2015)
Which algorithm was used	Deep learning
Who completed the analysis	Artificial Intelligence for Digital Response (AIDR), Qatar Computing Research Institute, MicroMappers; World Bank and UAViators acquired the imagery
Results and lessons learned	<ul style="list-style-type: none"> • A pipeline was developed for combining crowdsourced damage annotation and deep learning with 63% accuracy. • Tests on a damage event in the Philippines were 41% accurate, demonstrating a need for more training data to improve model predictions.
More information	<p>Lessons from Mapping Geeks: How Aerial Technology Is Helping Pacific Island Countries Recover from Natural Disasters</p> <p>Nazr-CNN: Fine-Grained Classification of UAV Imagery for Damage Assessment</p>

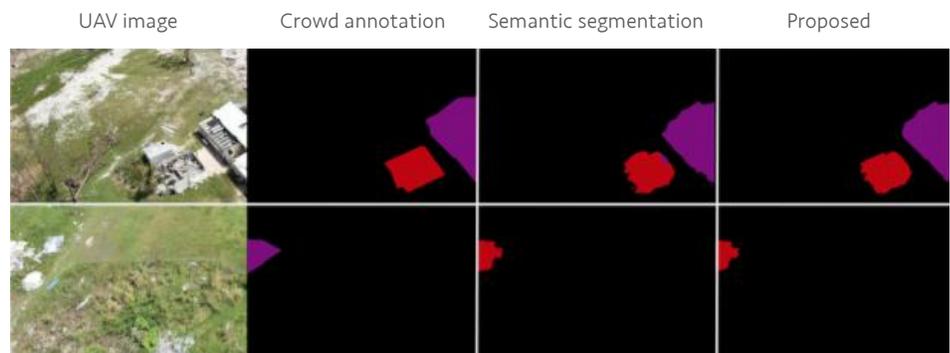


Figure from the original study, available here: <https://arxiv.org/pdf/1611.06474.pdf>

7. GLOSSARY

AGI: Artificial General Intelligence—the artificial intelligence that does not exist yet, where computers have learned the ability to be self-aware and tackle all different types of generalized problems in a way that’s indistinguishable from human intelligence

AI: Artificial intelligence—a term used to describe all types of computer machine learning

CAPTCHA: Completely Automated Public Turing test to tell Computers and Humans Apart—a tool ubiquitously used in web pages to discern humans from machines in an attempt to protect online resources from malicious software

Commons: Resources and information that are freely available to all members of a community, e.g., wikis and open-source software

Crowdsourcing: A method of creating data that leverages the communal work of a team or community (crowd), using software often made that allows the communal effort to be properly saved, validated, and analyzed to then become a common asset

Deep learning: A term that references the architecture of neural network algorithms, where there are hidden layers between the inputs and outputs that connect with each other in a way similar to neurons in the brain, albeit with many fewer connections

DRM: Disaster risk management

ESA: European Space Agency

Forests (of Decision Trees): A common supervised ML algorithm, where the term “forests” refers not

to the biological ecosystem, but to the fact that it uses many decision “trees”—decision structures where a yes/no decision is made at every fork, creating a “tree”

GANs: Generative adversarial networks
<https://skymind.ai/images/wiki/GANdancers.png>
<https://skymind.ai/wiki/generative-adversarial-network-gan>

GDAL: Geospatial Data Library

GEOSS: Global Earth Observation Systems of Systems

GFDRR: Global Facility for Disaster Reduction and Recovery of the World Bank

GOST: Geospatial Operations Support Team of the World Bank

GRASS: Geographic Resources Analysis Support System

HDX: Humanitarian Data Exchange

HOT: Humanitarian OpenStreetMap

ISET: Informal settlement

LiDAR: Light Imaging Detection and Ranging

MLA: Machine learning algorithm

OBIA: Object-Based Image Analysis

OpenAerialMap: Online platform for sharing openly satellite, aerial, and drone imagery

OpenStreetCam: An open data version of Street View, with street-level imagery collected from the ground
<https://openstreetcam.org/>

Optical Imagery: Imagery that is obtained via an optical sensor, whether in the visible Red-Green-Blue bands or in other wavelengths of the electromagnetic spectrum

OSM: OpenStreetMap, a global crowd-sourced map of roads, buildings, and other physical features. OSM is an open, collaborative, crowdsourced version of other common maps, such as Google Maps or Bing Maps

QGIS: Quantum GIS, an open-source GIS software

Radar/SAR: Synthetic-aperture radar, a type of sensor used in earth observation

RMSE: Root-mean-square error, a type of statistical analysis used to assess the accuracy of MLA results

Supervision: Human training of ML algorithms to learn to classify data according to set target parameters

UAV: Unmanned Aerial Vehicle

8. REFERENCES AND RESOURCES

There are a number of online resources available, and for the user that wants to go more in depth, there are indeed courses as well as many academic papers and textbooks that can be referenced. The following is a curated list of these references and resources

8.1 ONLINE RESOURCES

- One of the most thorough and up-to-date courses on machine learning is from **TechChange: Artificial Intelligence for International Development**
<https://course.tc/301-1/c>
- **Educational resources on AI from Google**
<https://ai.google/education/>
- **Crash course on Machine Learning from Google**
<https://developers.google.com/machine-learning/crash-course/>
- **A Machine Learning online course on Coursera, from Stanford University**
<https://www.coursera.org/learn/machine-learning>
- **Along with the above Coursera course, the following is specifically about Unsupervised Learning**
<https://www.coursera.org/learn/machine-learning/lecture/olRZo/unsupervised-learning>
- **Another resource for learning statistical methods is Datacamp**
<https://www.datacamp.com/>

8.2 VIDEOS AND TALKS

- **PBS Machine Learning and Artificial Intelligence: Crash Course Computer Science #34**
https://www.youtube.com/watch?time_continue=687&v=z-EtmaFjieY
- **Deep learning in medical imaging**
https://www.youtube.com/watch?v=2_Jv1VpOF4&feature=youtu.be&t=4m7s

8.3 INFOGRAPHICS AND INTERACTIVE RESOURCES

- **Understanding Machine Learning**
<https://futurism.com/images/understanding-machine-learning-infographic/>
- **Machine Learning 101**
<http://usblogs.pwc.com/emerging-technology/machine-learning-101/>
- **A Beginner's Guide to Machine Learning Algorithms**
<http://dataconomy.com/2017/03/beginners-guide-machine-learning/>
- **A Visual Introduction to Machine Learning**
<http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>
- **The mostly complete chart of neural networks, explained**
<https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>

8.4 ARTICLES AND BLOGS

- **A Tour of the Top 10 Algorithms for Machine Learning Newbies**
<https://www.kdnuggets.com/2018/02/tour-top-10-algorithms-machine-learning-newbies.html>
- **Top 10 Machine Learning Algorithms for Beginners**
<https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/>
- **Experts, Crowds, Machines—Who Will Build the Maps of the Future?**
<https://blog.mapillary.com/update/2017/12/21/who-will-build-the-maps-of-the-future.html>
- **Updating Google Maps with Deep Learning and Street View**
<https://research.googleblog.com/2017/05/updates-google-maps-with-deep-learning.html>
- **Introduction to GBDX**
<https://platform.digitalglobe.com/gbdx/>
- **GBDX Overview**
<https://gbdxdocs.digitalglobe.com/docs/gbdx-overview-1>
- **Machine Learning and ethics—Toward ethical, transparent and fair AI/ML: a critical reading list**
<https://medium.com/@eirinimalliaraki/toward-ethical-transparent-and-fair-ai-ml-a-critical-reading-list-d950e70a70ea>
- **The Building Blocks of Interpretability**
<https://distill.pub/2018/building-blocks/>

8.5 CONFERENCES AND MEETINGS

- **Computer Vision conferences: ECCV, ICCV, CVPR, etc.**
- **GFDRR Understanding Risk**
<https://understandrisk.org/>
- **AI for Good Global Summit 2018**
<https://www.itu.int/en/ITU-T/AI/2018/Pages/default.aspx>
- **Mapbox—Locate**
<https://www.mapbox.com/locate>

8.6 CHALLENGES AND COMPETITIONS

Challenges are an effective approach to get multiple people to try and tune models to the best of their ability in order to get the most accurate results.

- **We Robotics Open AI Challenge**
<https://blog.werobotics.org/2018/05/16/announcing-winners-open-ai-challenge/>
- **DeepGlobe**
<http://deepglobe.org/>
- **SpaceNet**
<http://explore.digitalglobe.com/spacenet>
- **DSTL Satellite Imagery Feature Detection**
<https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection/>
- **Functional Map of the World Challenge**
<https://www.iarpa.gov/challenges/fmow.html>
- **DIUx xView 2018 Detection Challenge**
<http://www.xviewdataset.org/>

8.7 OTHER REFERENCES, ARTICLES, AND TEXTBOOKS

- Engstrom et al. 2016.
<http://pubdocs.worldbank.org/en/60741466181743796/Poverty-in-HD-draft-v2-75.pdf>
- Geo-diversity for better, fairer machine learning
<https://developmentseed.org/blog/2018/03/19/geo-diversity/>
- Gevaert C M, Persello C, Sliuzas R, and Vosselman G. 2017. Informal settlement classification using point-cloud and image-based features from UAV data *ISPRS Journal of Photogrammetry and Remote Sensing* Complete 225–36.
- Graesser J B, Cheriadat A M, Vatsavai R, Chandola V, and Bright E A. 2012. Image Based Characterisation of Formal and Informal Neighborhoods in an Urban Landscape *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5. <https://www.osti.gov/scitech/biblio/1050316>
- James G, Witten D, Hastie T, and Tibshirani R. 2013. An Introduction to Statistical Learning vol 103 New York, NY: Springer New York.
<http://link.springer.com/10.1007/978-1-4614-7138-7>
- Kirschbaum, D. B., T. Stanley, and J. Simmons (2015), A dynamic landslide hazard assessment system for Central America and Hispaniola, *Nat. Hazards Earth Syst. Sci.*, 15(10), 2257–2272, doi:10.5194/nhess-15-2257-2015.
- Kirschbaum D and Stanley T 2018 Satellite-Based Assessment of Rainfall-Triggered Landslide Hazard for Situational Awareness *Earth's Future* 6 505–23
- Machine Learning Applications for Earth Observation.
https://link.springer.com/chapter/10.1007/978-3-319-65633-5_8
- Mather P M, and Koch M. 2011. Computer Processing of Remotely-Sensed Images: An Introduction John Wiley and Sons.
- Mathieu P-P, and Aubrecht C. 2017. Earth observation open science and innovation New York, NY: Springer Science+Business Media.
- Sethi I K. 1990. Entropy nets: from decision trees to neural networks. *Proceedings of the IEEE* 78, 1605–13.
<https://ieeexplore.ieee.org/document/58346/figures>
- Shankar et al. 2017. No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World. <https://arxiv.org/pdf/1711.08536.pdf>



GFDRR

Global Facility for Disaster Reduction and Recovery

Photo Credit: World Bank