POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

An open machine learning framework for residential water consumption estimation

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Loris Panza

Advisor: Prof. Andrea Francesco Castelletti

Co-advisors: Prof. Dr. Andrea Cominola (TU Berlin), M.Sc. Wenjin Hao, M.Sc. Siling Chen (TU Berlin)

Academic year: 2021-2022

---

1. Introduction

Over 1 billion people worldwide lack access to water and a total of 2.7 billion experience water scarcity for at least one month of the year. Many of the water systems that keep ecosystems thriving and feed a growing human population have become stressed. Rivers, lakes, and aquifers are drying up or becoming too polluted to use, especially in the urban environment. Climate change is altering patterns of weather and water around the world, causing shortages and droughts in some areas and floods in others. The extent of human water requirements is increasing rapidly at the global scale and it is crucial to analyze the possible imbalance between water demands and supply under various scenarios of climate change and across various temporal and spatial scales. Due to their high population density and water-intensive activities, urban areas frequently use the most water and are susceptible to water stress. Water shortage is a problem in many regions as a result of the continued urbanization and growth of the world's population (Lambin and Meyfroidt, 2011). As a result, in recent years, the emphasis on managing urban water resources has switched from just creating infrastructure to accommodate urban growth to implementing more sustainable and all-encompassing water demand management (Diaz et al., 2016; Brown et al., 2009). Knowledge of when, where, and how people use water is essential to model water demands and inform demand management strategies. However, lack of data on residential water consumption at the household level often limits our knowledge on past and current water demands with high spatial and temporal resolution. A primary source of difficulties is the absence of water consumption monitoring infrastructure (i.e., water meters) which, when available, consists of analog sensors that do not automatically log and transmit data on water consumption. The absence of standardization in data collection and reporting is another source of issues. In many countries, data gathering on residential water consumption is sporadic and inconsistent, preventing comparison across regions or time periods. Furthermore, water consumption data are often stored by water utilities, but are hardly accessible for research purposes,

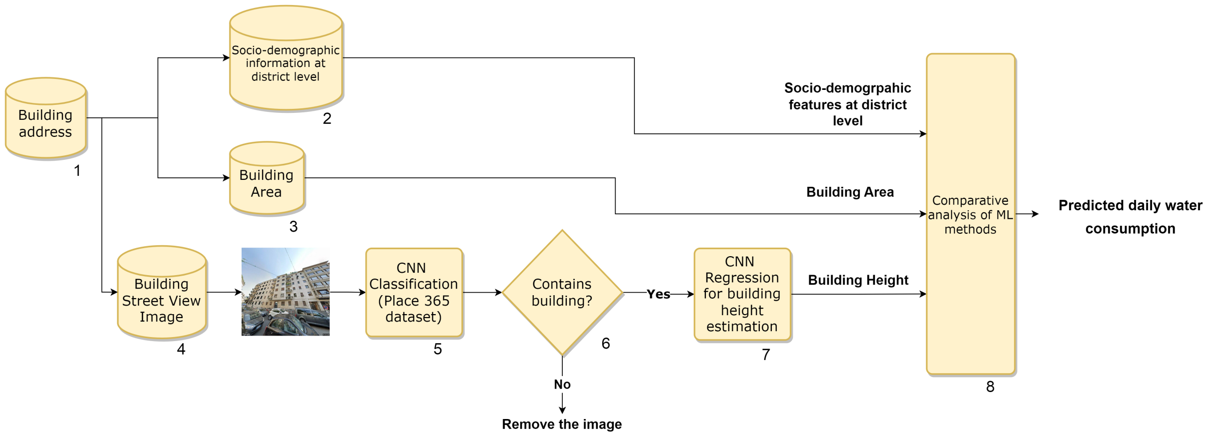

Figure 1: Methodology pipeline of our open machine learning framework for residential water consumption estimation.

when available. Without reliable data, it is difficult to identify the most significant drivers of water consumption and evaluate the effectiveness of different water-efficient technologies and water conservation practices. In this context, the thesis project consists in estimating the residential daily water consumption relying exclusively on publicly available data and machine learning techniques.

2. Methodology

The methodology implemented (Figure 1) in this study involves the use of separate models for processing Google Street View (GSV) images, socio-demographic features, and open building data based on specific building addresses. These models generate distinct sets of features, which are combined and considered as inputs to a machine learning (ML) model for estimating the daily water consumption of each building. Firstly, GSV images of building addresses (1 steps in Figure 1) are filtered using a Convolutional Neural Network (CNN) to extract only the ones showing buildings with clearly identifiable facades (steps 5, 6). Thus, the height of the building is estimated through a regression CNN applied to the building image (step 7). The relative area is retrieved from the public shapefile (step 3), as it will be explained in Section 3. Additionally, publicly available socio-demographic data, such as the population density of the district where the buildings are located, represent the last feature branch fed in the final model (step 2). Different machine learning algorithms are evaluated to predict the daily water consumption of residential buildings and their performance is compared (step 8). However, the target variable was also discretized into different consumption levels, leading to a change in the problem type from regression to classification.

2.1. GSV images acquisition and selection

Starting from a specific building address, the GSV image is extracted through the relative API. Since some street view images are unsuitable, i.e., buildings are blurred, or an image only shows streets or vegetation, a CNN classification is applied to select relevant images, i.e., those that show clearly identifiable buildings. Each available GSV image can be requested in a HTTP URL form using the GSV Image API. By defining URL parameters sent through a standard HTTP request using the GSV Image API, users can get a static image in any direction and at any angle for any point where GSV is available. After the image acquisition, a pretrained CNN was used to filter the GSV images; the objective is to remove invalid images such as the interior of buildings and those in which facades had been obscured by large vehicles (e.g., buses) or greenery. The CNN used is called Places365-VGG16 and it is a CNN pretrained on a subset of *Place dataset* a quasi-exhaustive repository of 10 million scene photographs, labeled with 434 scene semantic categories (Zhou et al., 2017). The CNN was applied to the GSV raw images and the list of all detected scenes were reported with the relative probability. After sorting the detected classes according to the probability given the GSV image, if none from the top 5 most probable predicted classes belong to the building list exceeding a specified threshold probability, the image is filtered out. Figure 2 and 3 show respectively an example of an invalid and valid image.

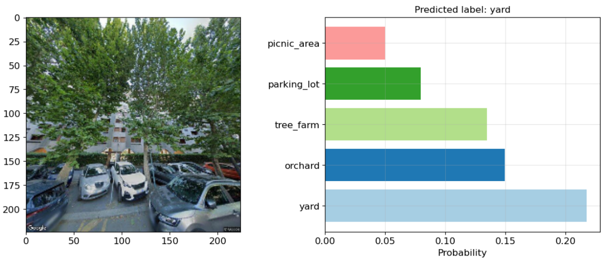

Figure 2: Places365-VGG16 predictions on GSV invalid image.

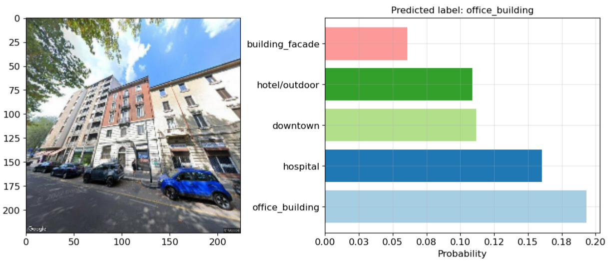

Figure 3: Places365-VGG16 predictions on GSV valid image.

2.2. Convolutional Neural Network for building height estimation

Once the valid building's GSV images are extracted, the building height associated with them was considered as a target to train a new Convolutional Neural Network. To reach the best height estimation, different kinds of architectures, input preprocess functions and training procedures were taken into consideration. The dataset contains about 4245 images. It was split into a training set of 2717 samples, a validation set of 679 samples and a test set of 849 samples.

2.2.1. Baseline model

As a baseline CNN, a handcrafted Convolutional Neural Network architecture was designed for image regression tasks. The architecture consists of several convolutional layers, each followed by a max pooling layer. The used convolutional layers contain in sequence 16, 32, 64, 128 and 256 filters, 3x3 kernel size, ReLU activation, and the same padding. The max pooling layers following them have 2x2 pool size. The last convolutional layer is followed by a flattening layer that converts the output into a 1-dimensional vector, which is then passed through two fully connected layers with dropout (Srivastava et al., 2014) regularization, and finally to an output layer with a single output unit and linear activation. The architecture is shown in Figure 4.

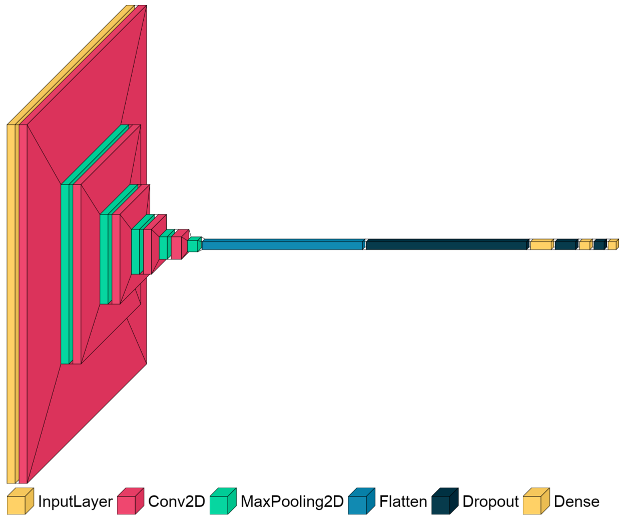

Figure 4: Baseline CNN architecture.

After an initial standard training, in the successive steps, some improvements were actuated to increase the performance both on the architecture and on the input data. In addition to considering simple rescaled images as input, also *data augmentation* (Perez and Wang, 2017) was applied to the training dataset. It is a technique to modify image characteristics by applying flips, rotation (at 90 degrees and finer angles), translation, scaling, noise addition, etc. By generating new images with different variations, the CNN is exposed to more diverse examples, which can help to improve its robustness to generalize to new, unseen images.
In addition, since the baseline architecture was designed manually and arbitrarily, KerasTuner has been leveraged to tune at least the most important parameters that compose the structure. It is an open-source hyperparameter tuning library for Keras that automates the process of hyperparameter tuning and architecture search by searching over a defined search space of hyperparameters using several search methods, the one chosen for the project is the hyperband one (Li et al., 2016).

2.2.2. Pretrained models: VGG16, ResNet50 and Places365-VGG16

One of the major challenges in developing effective CNN is the requirement for a large amount of data to enable the model to recognize underlying patterns and long-term trends. To overcome this issue and improve performance, *transfer learning* has been employed. This technique involves storing the knowledge gained from a specific task in a model that can be repurposed for a different but related task. Therefore, some models already pretrained on ImageNet were chosen: VGG16 (Simonyan and Zisserman, 2014) and ResNet50 (He et al., 2015). The original classification objective of the ImageNet CNNs was to classify images from the ImageNet dataset into one of 1,000 object categories. In addition to ImageNet, the Places dataset (Zhou et al., 2017) and its related model trained on it that has been used for the GSV image classification (Places365-VGG16) is also used for transfer learning: the CNN was used to recognize different scenes, including the characteristics of the buildings framed, and it can certainly be influential in the new task of estimating building heights. The workflow was to take these successfully pretrained CNNs, remove and design the new final fully-connected layers to match the new problem, freeze the weights of the previous layers and train the added layers in the new network using the training data. Since the target is the height, the CNN problem is a regression one; in that case, the old final layers used for the classification were removed for each CNN (ResNet50, VGG16, Places365-VGG16) and the introduced final architecture is designed to have as output layer only one neuron with a *linear activation function* to match the new regression task.
In transfer learning, the pretrained model is used as a feature extractor, where the features learned by the pretrained model are fed into a new classifier to make predictions on a new dataset. Typically, the pretrained model is kept unchanged during this process, except for possibly the final layer which is replaced with a new layer tailored to the new task. In contrast, fine tuning involves training not only the final layer but also some of the earlier layers of the pretrained model on the new dataset. It has been taken into consideration since transfer learning is a good option when the new task is similar to the task the pretrained model was trained on (ImageNet or Places), while fine-tuning can provide better performance in the case that the new task (estimating building height) is more from the original task. The results of VGG16, ResNet50 and Places365-VGG16 will be analyzed both using transfer learning and fine tuning technique. The optimization loss used during the CNNs training is the *Mean Squared Error* (MSE):

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{n=1}^{N} (y_i - \hat{y}_i)^2 \tag{1}$$

while the performance metric is the *Mean Absolute Error* (MAE):

$$MAE(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2}$$

where y represents the real target and $\hat{y}$ the estimation.

2.3. Machine learning models for water consumption estimation

Once the height of the building is estimated, the area and socio-demographic information associated with the neighborhood where the building is located is normally publicly available. The socio-demographic data can be divided into two categories: data on the number of residents divided by age and gender and data regarding the size of families occupying the residential buildings in the urban neighborhood. Water consumption may be related to many potential factors (determinants) with nonlinear relationships, mostly unknown. The goal is to be able to discover and express and understand this nonlinearity using a machine learning model while relying only on relevant input features. The target of the model is to estimate the average daily water consumption of a residential building (known via water meter reading data from utilities) using only public information sources. The dataset consists of 1699 samples, 80% of which will be used as the training set and the remaining portion as the test set. Several regression methods were considered: *Linear Regression, Polynomial Regression, K-Nearest Neighbors, Random Forest* and *Extreme Gradient Boost*. Before involving all the features, a baseline model was evaluated having as input only the area and height of the building while the target is daily water consumption. The goal is to have a basic model to understand how much building characteristics were influential in estimating water consumption and how estimation changes when inserting socio-demographic information. For evaluating the regression machine learning algorithms, *R-squared* (R^2) and *MAE* were calculated. The formula for R^2 is:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{3}$$

$$RSS = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{4}$$

$$TSS = \sum_{i=1}^{N} (y_i - \overline{y})^2 \tag{5}$$

where the *residual sum of squares* (RSS) represents the squared difference the sum of the differences between the predicted $\hat{y}$ and actual values y of the dependent variable. The *total sum of squares* (TSS) calculates the sum of the squared differences between the actual values and the mean value of the dependent variable. Apart from considering the water consumption estimation as a regression problem, the target was also discretized identifying three levels of building water usage as "low", "medium", or "high" water use: the division of the classes took place via percentiles. The 33rd percentile which defines the upper limit for the "low" water consumption set assumes a value of 4.19 m^3 while the 66th percentile that is the lower bound for the "high" water usage set is equal to 10.46 m^3. The water usage within those two limits identifies residential buildings belonging to the "medium" set. The ML techniques for classification evaluated the *KNN*, *Decision Tree*, *Random Forest*, *Gaussian Naive Bayes*, *Logistic Regressor*, *Ada Boost* and *Extremely Randomized Tree*. The *bagging* technique was combined with the KNN and Decision Tree as base models to increase the performances. The metrics used to evaluate the performance of the classification models in this thesis are accuracy, precision, recall, and F1 score. These metrics are calculated using the following formulas:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. Before training the regression and classification models, outliers are identified thanks to three different methods: *local outlier factor* (Breunig et al., 2000), *isolation forest* (Liu et al., 2008), and *interquartile range*. All three methods will be applied separately, and the one that yields the best result will be chosen. The same type of analysis was performed on the scaling methods. *Standard scaler*, *robust scaler*, and *minmax* were applied individually to the data, and only the best one will be taken into consideration. Additionally, to have a more robust performance analysis of the models on the chosen metric and select the best ML algorithms hyperparameters, *nested cross-validation* was applied. *Nested cross-validation* is a technique commonly used in machine learning to evaluate the performance of a model and to select its hyperparameters. It involves performing an outer loop of *k-fold cross-validation* to estimate the generalization performance of the model, and an inner loop of k-fold cross-validation to tune the hyperparameters of the model.

3. Data

In this research, data from mainly 3 sources are used: (i) time series water consumption dataset for more than 1500 buildings in the city of Milan (Italy), (ii) a shapefile containing information on building location and features, and (iii) open datasets of socio-demographic features at district level. The water consumption time series are used as targets for the ML models, while the shapefile is used to extract the building's height used as a target for the CNN and the area that will be considered as a direct input with the socio-demographic features to the final model. The time series water consumption is data shared confidentially by the water utility while the shapefile and socio-demographic features are open and constitute all needed for the model input.

3.1. Water consumption time series

The water consumption data are provided by MM Spa. MM Spa is a company created by the City of Milan in 1955 to design and build the first underground lines. The water consumption time series used in this study starts on 01-01-2019 and end on 08-03-2020 and are collected from a specific *PDR* (Punto di Riconsegna). The PDR is a numerical code that uniquely identifies the location of individual water use. The displayed value represents the water consumption of a building associated with the PDR in a single day. For each building and corresponding PDR, there are an associated address and civic number, a flag indicating whether the structure is residential or not, the NIL code and NIL name where the structure is located, and finally its latitude and longitude. A preprocessing phase was necessary to manage missing and outliers values. Figure 5 represents an example of the raw time series that will be preprocessed and then used for calculating the average daily water consumption.

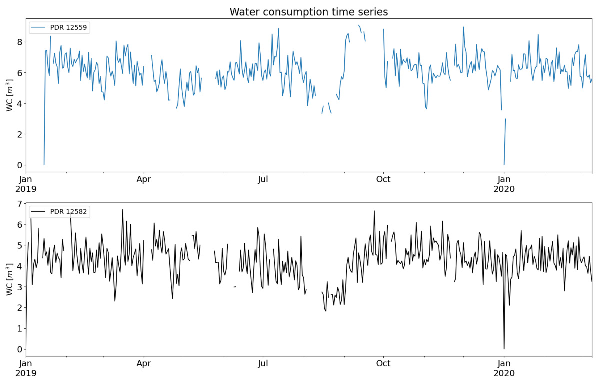

Figure 5: Examples of two water consumption time series associated to two different PDRs.

3.2. Building shapefile

A *shapefile* is a popular geospatial vector data format for storing and sharing geographical data. A city shapefile contains geographic information on the boundaries and features of a particular city, typically represented as a collection of interconnected *points*, *lines* and *polygons*. In the official Milan geoportal website (MilanGeoportal, 2012), it is possible to find, as public data, shapefiles related to streets, buildings, waterways, greenery, and other features that make up the urban landscape of a city, providing valuable information for researchers studying various aspects of city life and infrastructure. The one related to the residential buildings is analysed for its information regarding the building's area and height. The aim was to map the building addresses in the water consumption dataset into the shapefile to associate the area of the building with its corresponding daily water usage. The height extracted and building GSV image will be part of the dataset for the dedicated CNN. The point distribution of building addresses in the water consumption dataset is shown in Figure 6.

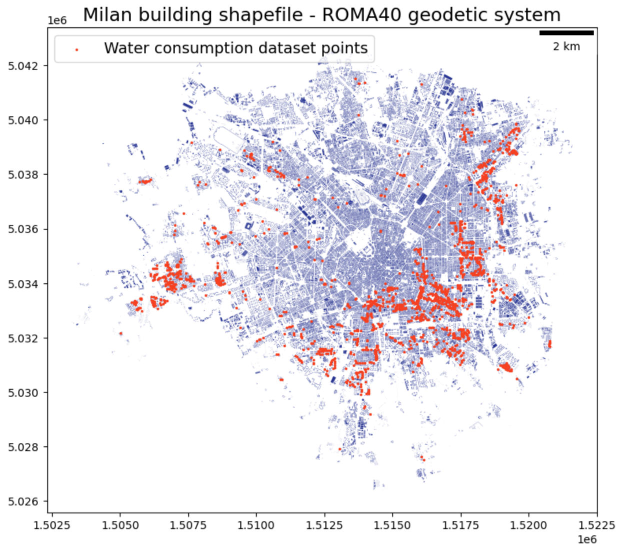

Figure 6: Red points represent the addresses associated with the water time series available in the dataset. The blue polygons describe the Milan building footprints.

3.3. Socio-demographic information

The geoportal of the Milan municipality contains various information on the socio-demographic characteristics of its inhabitants. The goal is to include useful features for the ML model. Due to privacy concerns, the surveys are at NIL level.

According to Milan Government Plan of the Territory (*PGT*: Piano di Governo del Territorio), Milan city area is divided into 88 NILs. *NIL* stands for "Nuclei di Identificazione Locale" which translates to "Local Identification Nuclei": they are administrative subdivisions

	Test Set		Validation Set	
Model	**MAE [m]**	**MSE [m^2]**	**MAE [m]**	**MSE [m^2]**
Baseline CNN	4.09	28.84	4.08	27.19
VGG16	4.18	30.49	4.11	28.02
ResNet50	**4.05**	**27.70**	**3.87**	**25.70**
Places365-VGG16	4.20	30.74	4.15	28.44

Table 1: CNNs building height estimation performances.

that are used for a variety of purposes such as census data collection, urban planning, and public services. For each NIL, both the population amount and the size of the neighborhood are indicated. Using this information, it is possible to calculate the population density in that area of the city. Additionally, the population composition, i.e., the numbers of males, females, minors and people over 65 years old are public and used in the final models developed in this thesis. Based on the number of members composing them, the families are divided into three categories: (i) single families composed of only 1 person, (ii) multi families composed of 2-4 people, and (iii) large families with more than 4 people. For each NIL, the number of these three kinds of family was also used as socio-demographic feature for the model.

4. Results

4.1. CNNs performances for building height estimation

Table 1 shows the best results for each type of CNN. Regarding the baseline model, the best performance on the test set was achieved by training with the parameters selected by KerasTuner. The VGG16 architecture performed better using transfer learning technique and thus utilizing the weights learned from ImageNet as a basis for a new training for the new task of predicting building heights. However, the result was not sufficient to outperform the simpler baseline model with tuned hyperparameters on the test set: it achieved a MAE of 4.09 m while the VGG16 achieved 4.18 m. ResNet50, on the other hand, achieved the best results on both the test (4.05 m) and validation set (27.70 m^2) through a retraining of all its layers, which brought the greatest benefits. The final pretrained Places365-VGG16 did not reach the same performances as the other CNN architectures, demonstrating that the ImageNet dataset was more related than Places365 to the final objective.

The CNNs estimations are slightly imprecise for a main reason: despite the fact that invalid images were filtered out by the CNN Places365-VGG16, there are many GSV images that capture multiple buildings of different heights, or they are quite zoomed in and cannot capture the entire height of the building. Surely this can be a limitation in the final calculation of the target. Overall, regardless of the architecture, results are always very similar and consistent, suggesting that all models are able to provide fairly good estimations. The mean MAE value on the test set is about $4m$ and so an uncertainty a little bit higher than 1 floor.

Figure 7 shows two GSV images fed in input to the ResNet50 architecture.

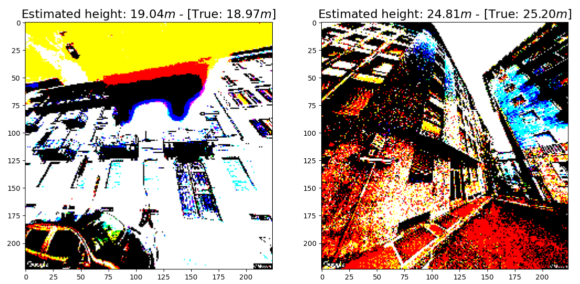

Figure 7: ResNet50 buildings height prediction on GSV images. On the left of the image title, it is reported the height estimated by the RseNet50 model while on the right it is showed the true height.

All the GSV images are passed through a specific preprocess function before being fed into the ResNet50 CNN. That function will convert the input images from RGB to BGR, then will zero-center each color channel with respect to the ImageNet dataset, without scaling. For that reason the image color is different from the original one.

4.2. ML performances for daily water consumption estimation

The baseline ML model has only two features, i.e., building height (m) and area (m^2), while the target variable is the daily water consumption (m^3). The performances shown in Table 2 suggest that polynomial regression achieves the best performance. Yet, all baseline models do not provide very accurate estimations of daily water consumption in regression mode. We can explain this as due to the following reasons: first of all, trying to predict the water consumption of a building considering only its physical characteristics might be limited. Water usage is closely related to human behavior and for this reason, it is complex to calculate it even knowing the smallest details.

	Test Set	
Model	**MAE [m³]**	**R2**
Linear Regression	5.08	0.32
Polynomial Regression	4.90	**0.38**
KNN	**4.78**	0.34
Random Forest	5.09	0.29
XGBoost	4.70	0.32

Table 2: Baseline models performances

Apart from considering the building shape information about area and height, the final ML models are built considering also socio-demographic features at NIL level in the input set. Looking at Table 3, the introduction of new features improves the overall performance of the models, with decision tree-based algorithms being the ones that benefited the most.

	Test Set	
Model	**MAE [m³]**	**R2**
Linear Regression	4.66	0.28
Polynomial Regression	4.70	0.25
KNN	4.68	0.27
Random Forest	4.49	**0.39**
XGBoost	**4.19**	0.38

Table 3: Final models performances

On the other hand, KNN did not improve due to the curse of dimensionality: it refers to the phenomenon where the performance of the K-nearest neighbors (KNN) algorithm deteriorates rapidly as the number of features or dimensions in the data increases. The improvement due to the new features is not so crucial, but it demonstrates that by integrating demographic and social information, a better result can be achieved and so it assesses that kind of information is influential in water consumption estimation. The hypothesis is confirmed looking a the Figure 8 that describe the feature importance analysis performed on the ML algorithms based on tree structure: Random Forest and XBGoost. XGBoost algorithm seems to balance the relevance of each feature even if there is still a certain difference between the height and area from the other ones. It should be noticed that the importance coefficient of building characteristics is small and it is not so far from the values of the socio-demographic factors differently from the Random Forest case.

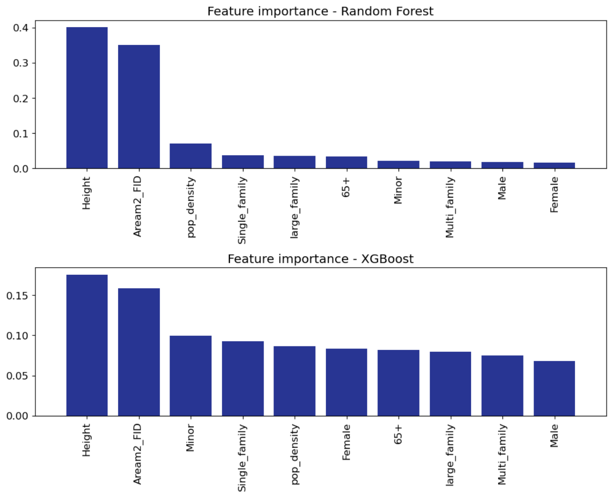

Figure 8: Features importance for the Random Forest and XGBoost algorithms.

The daily water consumption was discretized into three intervals representing low, intermediate or high water consumption for that particular building: the division of the classes took place via percentiles as explained in the methodology Section 2. Figure 9 shows the accuracy calculated on the test set. Based on the results, the Bagging(Tree) algorithm achieved the highest accuracy on the test set, followed closely by KNN and Bagging(KNN). Random Forest also achieved decent performance, but the Extremely Randomized Trees and Ada Boost algorithms did not perform equally well. The Bagging(Tree) model has the highest F1 score of 0.651, followed closely by Bagging(KNN) and

KNN with F1 scores of 0.647 and 0.639, respectively.

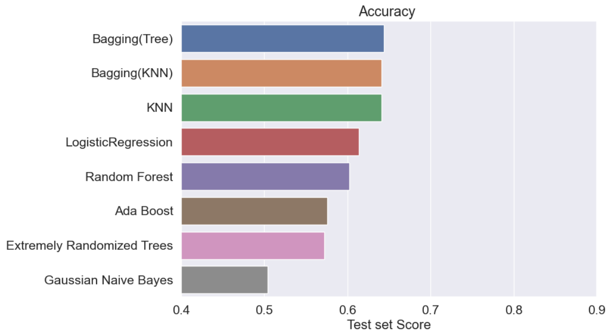

Figure 9: Comparison of models accuracy.

4.3. Water consumption approximate formula: a comparison

In order to see if the effort spent in developing and implementing deep learning and ML techniques was justified in terms of improved accuracy, the performances of the previously described models are compared to the most basic water consumption estimation formula based on approximation regarding the area, number of stories, population per square meter and water consumption per capita per day. ISTAT (Italian National Institute of Statistics) is the main statistical institute in Italy. It is responsible for collecting, processing, and disseminating official statistics on the Italian economy, society, and environment. The most recent data on water consumption in Italy published by Istat is for the year 2020, which shows that the average daily water consumption per capita in Italy was 236 liters (ISTAT, 2022). However, as already explained, this is just an average and the actual water consumption can vary depending on a range of factors, including the region, the time of year, and the type of building but for an approximative comparative formula is enough. The basic formula we consider is expressed as a multiplication of several factors:

$$b\_dwc = per\_capita\_dwc * b\_area * b\_stories * NIL\_pop\_density \tag{10}$$

The final outcome of the formula (b_dwc) represents the building daily water consumption expressed in liter per day. b_area and $b\_stories$ are the dimensional features of the building expressed respectively in m^2 and unit. per_capita_dwc is the daily water usage per single person estimate by ISTAT while $NIL\_pop\_density$ determines the number of people per square meters for each single NIL. Since the building daily water consumption estimated by the ML models is expressed in m^3, the formula outcome is converted from liters to the target unit measure by dividing it by 1000. With all the information at our disposal, the calculation of the average water consumption was applied to the entire dataset of 1699 samples. Obviously, since it is a simple formula, it does not require a training procedure but is simply applied to each row of the dataframe. The estimated final quantity variable was eventually discretized into three fundamental classes according to the limits previously discussed. To make the comparison between the approximate formula and the best classification algorithm discovered (bagging with Decision Tree) more reliable, the portion of the dataset used as the test set in the algorithm training will be used for evaluation. It contains 337 samples.
The results show (Table 4) that the bagging tree model has higher accuracy, precision, recall, and F1 score compared to the approximate model. This indicates that the machine learning approach is providing better results than the approximate approach. The difference in accuracy between the two models is quite significant. Similarly, the difference in precision is 0.135, recall is 0.157, and F1 score is 0.213.

	Bagging (Tree)	Formula
Accuracy	0.641	0.484
Precision	0.644	0.509
Recall	0.641	0.484
F1	0.642	0.429

Table 4: Comparison between best ML classifier developed and approximate formula.

These differences are meaningful and suggest that the bagging tree model is able to provide more accurate and precise predictions than the approximate model.

5. Conclusion and future research

This thesis contributed a data-driven machine learning framework to predicting water consumption using publicly available data. Different machine learning methods were comparatively assessed to evaluate their suitability to predict building-level water consumption. While the regression and even more classification algorithms achieved decent performances explaining the weight of each feature in the final prediction, our results also suggest that there is room for improvement of those algorithms that did not perform as well. Future research could focus on identifying ways to optimize these algorithms or developing new methods that can outperform the current state of the art regarding both the ML and deep learning fields. The analysis presented in this thesis demonstrates the value of using machine learning and deep learning techniques for water consumption prediction and highlights the potential of publicly available data sources for this type of study. The CNN that takes as input GSV images allows the extraction of the height dimension of the building and that technique is quite reproducible considering different locations, provided that certain conditions exist; the presence of the GSV images must be ensured, and the buildings exterior structure of the new targets should be similar to Milan's residential buildings as them represents the training set for the CNN. One possible improvement could be to expand the dataset of GSV images to include different types of buildings and make the model more generalized. Regarding the footprint of the structures, during the development of the project, it was considered as input data thanks to the Milan shapefile. But for future improvement, satellite images could be used to extract the area of interest since the shapefiles may not be publicly available or may not exist for certain areas. Even if shapefiles do exist, they may not have accurate or up-to-date information on the boundaries of the study area or the locations of individual buildings within it. In this way, the data related to the characteristics of the building would be dependent only to GSV and satellite images, two kinds of images that could be easily extracted on internet. As described in the results chapter, CNNs for building height estimation showed more satisfactory results than the final model. To increase the performance of both models, obtaining higher-quality data is certainly useful. Public data may not provide detailed socio-demographic and building dimension information and most of them have low and aggregated spatial resolutions, which can limit the effectiveness of water consumption estimation. The public socio-demographic features were collected at NIL level and increasing the resolution of data can provide more accuracy. The same type of problem is encountered in the shapefile used for the physical characteristics of the buildings: some structures are merged with others in the same polygon reducing the resolution of the information.

The comparison evaluated between the developed final model and the most basic approximate water consumption estimation formula supports more complex approaches based on machine learning and data-driven techniques but at the same time, they can be expensive in terms of computational resources and development effort.

References

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000. ISSN 0163-5808. doi: 10.1145/335191.335388. URL `https://doi.org/10.1145/335191.335388`.

R. Brown, N. Keath, and T. Wong. Urban water management in cities: Historical, current and future regimes. *Water science and technology : a journal of the International Association on Water Pollution Research*, 59:847–55, 02 2009. doi: 10.2166/wst.2009.029.

P. Diaz, P. Stanek, N. Frantzeskaki, and D. Yeh. Shifting paradigms, changing waters: Transitioning to integrated urban water management in the coastal city of dunedin, usa. *Sustainable Cities and Society*, 26, 04 2016. doi: 10.1016/j.scs.2016.03.016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

ISTAT. *ISTAT WATER STATIS-*

TICS / YEARS 2019-2021, 2022. https://www.istat.it/it/files//2022/04/Report_ISTAT-WATER-STATISTICS.pdf [Accessed: 24/03/2023].

E. F. Lambin and P. Meyfroidt. Global land use change, economic globalization, and the looming land scarcity. *Proceedings of the National Academy of Sciences*, 108(9):3465–3472, 2011. doi: 10.1073/pnas.1100480108. URL https://www.pnas.org/doi/abs/10.1073/pnas.1100480108.

L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. 2016. doi: 10.48550/ARXIV.1603.06560. URL https://arxiv.org/abs/1603.06560.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008. doi: 10.1109/ICDM.2008.17.

MilanGeoportal. *Territory shapefiles*, 2012. https://geoportale.comune.milano.it/sit/tematiche/territorio/ [Accessed: 03/03/2023].

L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning, 2017. URL https://arxiv.org/abs/1712.04621.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. URL https://arxiv.org/abs/1409.1556.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.