



PYSIMDEUM – AN OPEN-SOURCE STOCHASTIC WATER DEMAND END-USE MODEL

D.B. Steffelbauer¹, B. Hillebrand² and E.J.M. Blokker³

¹Hydroinformatics Group, KWB - Kompetenzzentrum Wasser Berlin, (Germany)

²KWR – Watercycle Research Institute, Nieuwegein, (the Netherlands)

³KWR – Watercycle Research Institute, Nieuwegein, (the Netherlands), Delft University of Technology, Delft, (the Netherlands)

¹ david.steffelbauer@kompetenz-wasser.de, ² bram.hillebrand@kwrwater.nl,

³ mirjam.blokker@kwrwater.nl

Abstract

Water demand is a crucial input parameter in water distribution system analysis because it can fluctuate over various temporal and spatial scales. In the past, researchers developed stochastic models that can provide realistic consumption patterns for simulations to account for those demand dynamics. Parameters for stochastic models are usually retrieved by fitting these models on smart water meter data. The stochastic demand model SIMDEUM uses an entirely different approach by generating highly realistic water demands based on (country-specific) statistical information only, without the need for measurements. While this approach makes SIMDEUM widely applicable in the water sector, its widespread usage within the community has been hindered due to its software implementation and availability.

We produced pySIMDEUM, an open-source and object-oriented implementation of the SIMDEUM software in the popular and freely available programming language Python. The pySIMDEUM software package is not only publicly available for usage within the water field — it is also intended to build the cornerstone of a widespread pySIMDEUM community of active developers. We want to use the WDSA/CCWI conference to address interested researchers or practitioners in the water sector and invite them to contribute to the software package as active part of the pySIMDEUM community. We will show SIMDEUM's history and past applications, the mathematical approach behind SIMDEUM and pySIMDEUM, where to download and install the pySIMDEUM package, the structure of the program, and a minimal example of how easily pySIMDEUM can be used to generate realistic stochastic water demand patterns from scratch. Furthermore, we will highlight possible future applications of the new pySIMDEUM tool. These applications include automatic parametrisation of pySIMDEUM parameters on smart meter data, coupling stochastic demands directly with hydraulic solvers, or how to enable city-scale stochastic demand simulations.

Keywords

Smart metering, water demand analysis, demand management, hydraulic simulations, water consumption, open-source software, Python.

1 INTRODUCTION

Water utilities use simulation models to design and operate distribution systems efficiently. In those models, water demand is a crucial input parameter that can fluctuate over various temporal and spatial scales. Historically, demand is modelled based on limited data through time-wise sparse consumption measurements (e.g., monthly, quarterly, yearly) and zonal meters with higher time resolution but are only available at a few locations. This approach resulted in inaccurate computer models, e.g., when simulating water quality. Researchers developed novel stochastic demand models that simulate water use with a high temporal and spatial resolution to overcome those inaccuracies, e.g., Poisson rectangular pulse or Neyman cluster process models [1]. Model

parameters for those novel demand models are usually estimated through fits on consumer data, e.g., from Automatic Meter Readings (AMRs) data.

The stochastic demand model SIMDEUM [2] uses an entirely different approach by simulating end-use pulses based solely on statistical information of users and appliances, all without needing expensive measurement campaigns. This different approach makes SIMDEUM highly flexible in its application, especially for zones lacking AMRs. However, the applicability of SIMDEUM has some limitations, which are partly connected to its implementation in MATLAB. First, as proprietary software, MATLAB is accompanied by high license costs, hindering the parallel execution of SIMDEUM simulations in modern multi-core computing environments such as high-performance clusters. This limitation prevents the scalability of SIMDEUM from performing simulations for large networks in a reasonable time. Second, SIMDEUM consists of MATLAB structures and direct calls to its library. The SIMDEUM source code requires the user to be aware of all the different functions offered by SIMDEUM to implement custom-tailored simulations or if the user wants to enhance SIMDEUM's capabilities.

Consequently, the lack of an object-oriented software engineering approach hinders open-source development. To resolve all formerly mentioned limitations, we introduce pySIMDEUM, which simulates water demand using the same methodology as the MATLAB version of SIMDEUM, but is implemented in an object-oriented way in the freely available programming language Python. We chose Python because

1. it is nowadays one of the most popular programming languages, which replaces or already replaced MATLAB in the scientific community,
2. its simple syntax is easy to use (e.g., for Civil Engineering students not familiar with high-level programming),
3. it possesses a rich ecosystem of packages for data science, optimisation, and water distribution modelling (e.g., WNTR or OOPNET).

We will show where to download and install the pySIMDEUM package, the mathematical approach behind SIMDEUM and pySIMDEUM, how this translates naturally into an object-oriented program in Python, and a minimal example of how to use pySIMDEUM to generate realistic stochastic water demands from scratch. Additionally, we want to use the WDSA/CCWI presentation to show how users can contribute to the pySIMDEUM open-source initiative. Furthermore, we will highlight possible future applications of the new pySIMDEUM tool. These applications include automatic parametrisation of pySIMDEUM parameters on smart meter data, coupling stochastic demands directly with hydraulic solvers, or enabling city-scale stochastic demand simulations.

2 PURPOSE AND MATHEMATICS BEHIND SIMDEUM

SIMDEUM, abbreviated for Simulation of water Demand and End-Use Model, is a stochastic drinking water demand model with a small spatial (end-use or device level) and temporal scale (up to one second). Initially intended to be a supporting tool for water quality modelling [2], for example, to calculate maximum demands within water distribution networks and the resulting maximum velocities to design self-cleaning networks related to discolouration, SIMDEUM has found many other applications, both in research and practice. The applications range from water quantity modelling (e.g., demand pattern generation for bottom-up and top-down allocation), water quality modelling (e.g., computing residence times in distribution networks, chlorine decay calculations, water temperature and bacterial growth simulations), design of distribution systems as well as drinking water installations, up to the prediction of future demands under changes in demographics or water usage behaviour. Additionally, novel use cases led to further developments and add-ons to the SIMDEUM software package (e.g., SIMDEUM hot and cold water use package or SIMDEUM's discharge pattern simulator for computing thermal energy and

nutrients recovery potentials). A detailed overview of the many applications for SIMDEUM can be found in [3].

2.1 Mathematics behind the SIMDEUM approach

The mathematics behind SIMDEUM is relatively simple. SIMDEUM simulates rectangular water consumption pulses (B) similar to Buchberger's and Wells' Poisson Rectangular Pulse (PRP) model from 1996 [4]. Within the model, each pulse is defined by its height or intensity (I), its duration (D), and its arrival time over the day or start time (τ). The intensity, duration, and start time of each pulse are drawn from probability distributions that are different for each type of end-use (e.g., shower, toilet flush, ...) as well as the type of user (defined through the user's age, gender, and occupation), respectively. In contrast to the PRP model, where the pulse parameters are retrieved through fitting the model on data, SIMDEUM generates the pulses from more high-level statistical information retrieved from country-specific statistics and surveys and is, hence, not dependent on available measurement data.

SIMDEUM generates a water demand pattern by summing up the randomly drawn end-use pulses per end-use (k), user (j) and frequency of use (i)

$$Q(t) = \sum_{k=1}^M \sum_{j=1}^N \sum_{i=1}^{F_{jk}} B(I_{ijk}, D_{ijk}, \tau_{ijk})$$

$$B(I_{ijk}, D_{ijk}, \tau_{ijk}) = \begin{cases} I_{ijk} & \dots \tau_{ijk} < t < \tau_{ijk} + D_{ijk} \\ 0 & \dots \text{else} \end{cases}$$

M is the number of all different end-use devices, N is the number of users in the household, and F_{jk} is the frequency of a user using a specific end-use. All parameters in the formula above (I , D , τ) are described by probability density functions. Those functions are different for each end-use type and can differ for the kind of person or the household. The probability functions and statistics are taken from statistical information, in particular, from data taken from surveys and questionnaires in the Netherlands.

2.2 Households, water users and end-use devices within SIMDEUM

SIMDEUM consists of three different main types of components, which are *households*, *water users*, and *end-use devices*. Three main household types are implemented: (i) *one-person* households, *two-person* households representing double-income-no-kids apartments, (iii) and *family* households consisting of two or more people that can also include children.

The inhabitants of the houses are water users. Each water user is characterised by their age, gender, and occupation. These characteristics impact the daily schedules of users and hence, their water usage. Daily schedules are generated from probability distributions linked to statistics that depend on age, gender, and profession. For example, the profession defines whether a user has a job and leaves the house for work during the day. Consequently, this user's water usage at home during work hours is zero.

SIMDEUM takes eight different end-use devices into account, i.e., *bathtub*, *bathroom tap*, *dishwasher*, *kitchen tap*, *outside tap*, *shower*, *washing machine*, and *toilet*. The type of the house and the number of people living in it influence the end-use devices present in a home. For example, a dishwasher or a bathtub is more likely to be present in a house the more people live in it. Each of the end-uses devices can possess different end-use subtypes. The subtypes are distinct from one another by the average usage duration. For example, the bathroom tap can be used to wash, shave, and brush teeth. These different usage types possess different probability distributions and

parameters connected to each end-use subtype. The frequency, duration, and intensity are drawn from those probability distributions.

The primary statistical information, i.e., specific probability distributions and their parameters (e.g., mean, standard deviation, etc.), are taken from surveys and questionnaires. For example, the frequency of use is taken from the residential water-use survey [5], while the statistics describing the daily schedules of the users (e.g., duration of sleep, time of leaving the house, ...) are taken from the time-budget survey [6]. Household statistics (e.g., number of people per household, their gender, or their age distribution) are taken from statistics Netherlands [7]. A detailed description of the household statistics and end-use devices, end-use subtypes, and the specific probability distribution per end-use subtype can be found in [2].

3 THE PYSIMDEUM PYTHON PACKAGE

3.1 Shortcomings and limitations of SIMDEUM's implementation

The variety of SIMDEUM's practical applications demonstrated its widespread utility. However, its general use and development within the water community have been hindered by the software's availability, usability, and implementation.

First, SIMDEUM has been only available through directly communicating with its developers and signing a memorandum of understanding. After that procedure, the developers would send you the software. Direct contact with the developers might already be a barrier for users who simply want to try out suitable software for their use case. Additionally, SIMDEUM is written in the proprietary language MATLAB, which comes with high license costs.

Second, SIMDEUM was developed more than ten years ago when object-oriented programming was unavailable in MATLAB. That is why SIMDEUM is based on a workaround for a proper object-oriented approach, consisting of a mixture of MATLAB structures and cells, which is not ideal for the usability (e.g., the understanding of users of the software) nor enforcing collaboration between multiple developers as in an object-oriented software consisting of various sub-packages. Furthermore, the three main components of SIMDEUM (households, water users, and end-use devices) plus the statistical data are ideally implemented in software that uses an object-oriented structure.

3.2 Python implementation and overview of the object-oriented structure

To address the shortcomings and limitations of SIMDEUM's MATLAB implementation, the decision was made to port SIMDEUM to Python and to refactor the software to enforce usability and collaboration through an object-oriented software design approach. Python is one of the most used programming languages worldwide, freely available, and includes a vast standard library that is topped up with a rich ecosystem of scientific software packages, all of them publicly available on *GitHub* (www.github.com) or the Python Package Index PyPI (<https://pypi.org>). Furthermore, Python is a multi-paradigm programming language that supports different programming approaches, including object-oriented programming. The usage of Python allows SIMDEUM to grow as an open-source software project and attract a big community of users, where each community member can contribute to the software's source code.

The main components of SIMDEUM (households, users, devices, and end-uses) contain different information and statistics depending on their type. Each of them influences the other and translates naturally into an object-oriented structure. Slight differences or variations in similar objects can be implemented via inheritance. For example, a toilet with a water-saving flush option can be the child of a more general standard toilet. Additionally, pySIMDEUM contains a statistics object to store underlying (country-specific) statistical information that influences people's water usage.

Figure 1 shows the overall structure of the pySIMDEUM package. The software flow is the following: First, a house is constructed based on underlying household statistics. Depending on the house type (one-person, two-person or family homes), the house is populated with water users.

Users are uniquely defined by age, gender, and job. The age defines the type of user. Four different age options exist, namely, (i) child, (ii) teen, (iii) adult, and (iv) senior. Each of them has different underlying statistics that define their daily schedules and water usage behaviours. For example, Dutch water use statistics show that teenagers shower more often than seniors. There exist two different options for the gender property (either male or female) and two different options for the job property, which have a strong influence on their daily water use schedule (either the user has a job where he leaves the house during the day [*True*], or the user stays at home [*False*]). The user's presence in the house and daily water use schedule are computed based on the user's properties and probability distributions defined in the (country-specific) statistics object.

Once the house is populated with users, the water use devices are generated by furnishing the house. This is again based on information contained in the statistics object. The users and devices present in a house define the possible end-uses with their statistical properties (i.e., underlying probability functions and parameters).

The house instance serves as a collector for storing different users and device objects (depicted as square brackets in Figure 1) and contains methods to simulate the water consumption of the entire household once users, devices and end-uses are defined.

The consumption is stored in a multi-dimensional labelled array, using the Python package *xarray* [10], with users, devices, samples and time as dimensions. The format of the output allows to access individual consumptions per user or device, compute statistics over all dimensions (e.g., mean, standard deviations, ...), including moving averages over the time axis, algorithms to plot water consumption, and routines to write the results to different file formats (e.g., Excel, text files, or to netCDF). Additionally, wrappers exist to write consumption patterns to EPANET input files, allowing for directly connecting pySIMDEUM and hydraulic simulations.

It has to be noted that pySIMDEUM is highly flexible: All statistical information, household, user and device properties can be changed in runtime, either by directly changing the properties (of houses, users, devices, or end-uses), changing (default) values within the construction methods, or changing the values stored in the statistics object directly. Additionally, the user can introduce new objects (e.g., novel end-use devices) by inheriting from similar devices and adding the novel device simply to the method that furnishes the houses. This gives pySIMDEUM users total control over the software to tailor the software and statistics based on their needs or use-case.

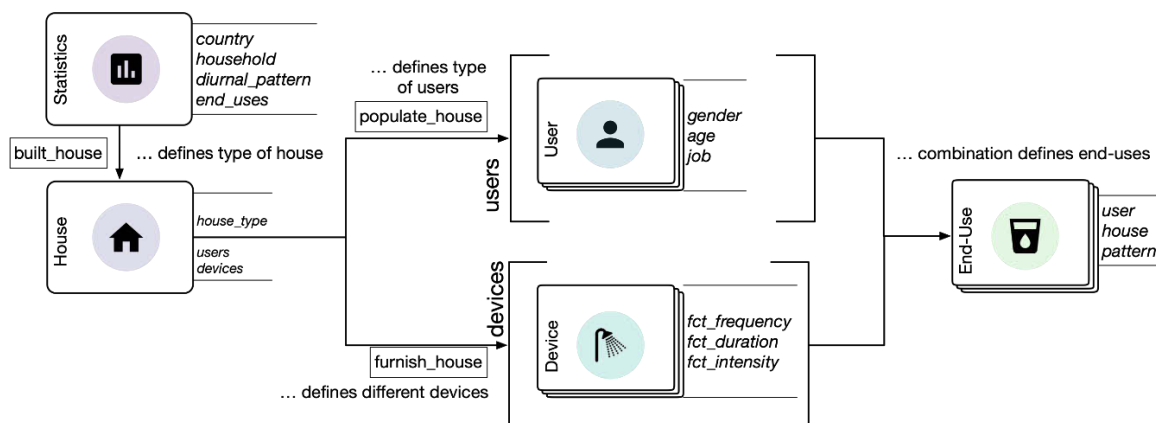


Figure 1: Object-oriented structure of pySIMDEUM

3.3 Source code repository, installation and dependencies

The pySIMDEUM software is available under the European Union Public License 1.2 on GitHub (from the conference's start date) under the following URL: <https://github.com/KWR-Water/pySIMDEUM>.

GitHub is a source-code hosting service that offers distributed version control and source code management based on the version control system Git. Additionally, it provides access control and advanced collaboration features, such as bug tracking, feature requests, task management, continuous integration, and wiki pages. The pySIMDEUM GitHub repository contains not only pySIMDEUM's source code but also a short README and a minimal usage example in the form of a Jupyter notebook, plus the complete documentation of the software. Furthermore, a link is available on the repository to the online documentation of pySIMDEUM on *Read the Docs* (<https://readthedocs.org>). Collaborators have to create a GitHub user account to contribute to pySIMDEUM, but anyone worldwide can download the repository.

Additionally, pySIMDEUM is hosted on PyPI. Hence, pySIMDEUM can also be installed via the command line by typing '`pip install pysimdeum`', which Python users are familiar with.

pySIMDEUM is available for Python version 3.7 or newer. The reason for this is that the objects in the software are implemented as Python data classes. This is a new feature for classes mainly containing data, which allows type hinting and saves a lot of boilerplate code (e.g., class initialisation and representation functions), making the source code more readable, understandable, and easier to maintain for pySIMDEUM developers. Further dependencies of the pySIMDEUM package include standard Python packages such as *numpy*, *scipy*, *matplotlib*, *pandas*, *xarray*, and *wntr* [8] for writing pySIMDEUM water usage patterns to EPANET input files.

3.4 Minimal usage example

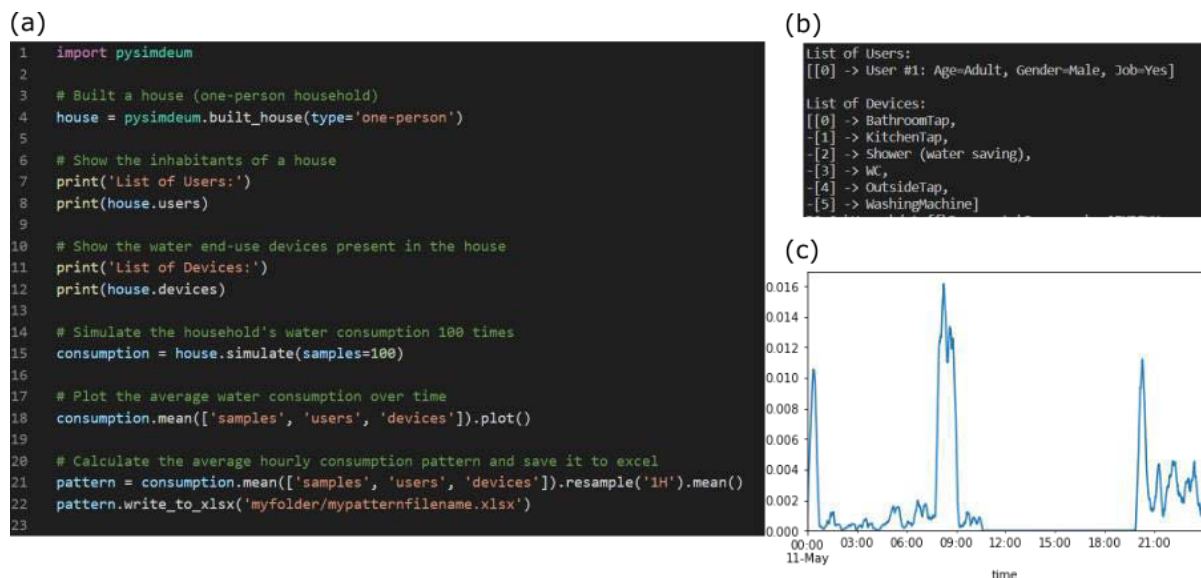


Figure 2: Minimal usage example for running a pySIMDEUM simulation: (a) code of the program, (b) Python console output, (c) graphical output of the simulation.

Figure 2 (a) shows the Python source code for simulating a water consumption pattern for a one-person household. First, the pySIMDEUM package is imported into Python (Line 1), enabling the user access to pySIMDEUM's application programming interface (API). Then, a house is generated

through the 'built_house' routine. The user can either specify a specific house type (i.e., a *one-person* household in this case) or if the type is not specified, pySIMDEUM will generate a random household (*one-person*, *two-person* or *family*) based on the underlying pySIMDEUM statistics. Furthermore, the number of inhabitants and the appliances within the house can be specified through the function's keywords 'num_users' or 'devices', respectively. Additionally, instead of providing pySIMDEUM with the number of users, a list containing specific users can be pre-defined and given to the house constructor via the 'users' keyword. All information that is not particularly given to the function will be drawn from random distributions defined within pySIMDEUM's underlying statistical information.

The house inhabitants can be depicted by printing the 'users' property of the generated 'house' object to the Python console (see Line 8 in the source code and the output in Figure 2 (b)). The printed information contains a list of users with the user's index in the list (0 in this case), and the essential properties of the user, i.e., their age, gender, and profession. The generated user in the example is a male adult with a job so the user will leave the house during working days. The same print function can be used to depict the end-use devices present in the household by printing the 'devices' property of the house (Line 12 in Figure 2 (a)). In this case, six water end-use devices are present in the house, a bathroom, a kitchen and an outside tap, a shower, a toilet, and a washing machine (see the output in Figure 2 (b)). Standard devices within pySIMDEUM are printed without additional information, while special devices are printed with other information in brackets. For example, the shower present in the one-person household is a water-saving shower, so the water usage intensity is less when a user takes a shower. Consequently, the water usage for shower events is lower than for houses having a standard shower.

In Line 15, the water consumption of the household is simulated. In particular, 100 samples are generated for the stochastic water usage of this specific house. The results are stored in a multi-dimensional labelled array using the *xarray* package. The array contains four dimensions: time, users, devices, and samples. Hence, each end-use device, user, sample, or point in time can be extracted by the user.

Furthermore, *xarray* allows building statistics over time over each dimension or combinations of dimensions. For example, the average consumption over time of the house can be retrieved by calculating the mean over the samples, users, and device dimensions (see Figure 2 (a) Line 18). Additionally, the statistics can be plotted (see Figure 2 (c)). It can be seen that the user's water consumption shows high peaks, one in the morning and one in the evening when the user returns from work. During the day, the user's consumption in the house is zero since he is not at home. Consumption during the night-time is low since the user is generally asleep. However, pySIMDEUM assumes that approximately 1.5 % of the water consumption happens during this time since users can get up during the night and use water, for example, when using the toilet. Of course, this parameter can be changed in pySIMDEUM as well. Finally, Line 21 in Figure 2 (a) shows how to generate water usage patterns by building averages over time intervals (i.e., one hour). Line 22 shows how to store the pattern subsequently in an MS Excel file with the 'write_to_xlsx' method.

4 CONCLUSION

SIMDEUM is a stochastic water demand end-use software that generates highly realistic drinking water demand patterns from statistical information without demand measurement data. While SIMDEUM has been used for many applications in the water sector, its widespread usage by researchers and practitioners in the water distribution system analysis field has been hindered mainly by its MATLAB implementation and availability. This paper presented pySIMDEUM, an open-source and object-oriented implementation of the SIMDEUM software in the popular and

freely available programming language Python. The pySIMDEUM software package is publicly available for usage within the water field and is intended to build the cornerstone of a widespread pySIMDEUM community of active developers. That is why we hosted the software on GitHub to enforce the joint development of pySIMDEUM. Every interested researcher or practitioner in the water sector is invited to contribute to the software package actively.

Currently, pySIMDEUM produces water demand patterns in the same way as the original SIMDEUM software. But in future, we expect basic SIMDEUM functionality to be extended, allowing to simulate, for example, nutrient loads or realistic dry weather flows in sewers, test the effect of novel water end-use devices on the drinking water supply system, simulate the potential of peak demand load shifting and, minimise the energy consumption of drinking water systems, or in modelling premise plumbing systems, where water usage on small spatial and temporal scale plays a crucial role.

Furthermore, the implementation of the software in the non-proprietary programming language Python allows the parallel execution of SIMDEUM simulations on high-performance computers (HPCs). Due to the high licensing costs coming with MATLAB, this was impossible in the past. Consequently, massive pySIMDEUM simulations on HPCs can be used to calculate realistic water consumption on a high temporal and spatial resolution on the city scale. This will also allow to couple pySIMDEUM simulations with smart meter data: Instead of tediously retrieving country-specific statistics necessary for the SIMDEUM approach, pySIMDEUM parameters might be retrieved automatically from smart meter data and machine learning algorithms [9]. The underlying information and statistics of SIMDEUM might be learned to produce even more realistic water usage patterns tailored to specific case study areas.

5 REFERENCES

- [1] E. Creaco, P. Kossieris, L. Vamvakieridou-Lyroudia, C. Makropoulos, Z. Kapelan, and D. Savic, "Parameterizing residential water demand pulse models through smart meter readings," *Environmental Modelling & Software*, vol. 80, pp. 33–40, 2016.
- [2] E. J. M. Blokker, J. H. G. Vreeburg, and J. C. van Dijk, "Simulating Residential Water Demand with a Stochastic End-Use Model," *Journal of Water Resources Planning and Management*, vol. 136, no. 1, pp. 19–26, 2010.
- [3] E. J. M. Blokker, C. Agudelo-Vera, A. Moerman, P. van Thienen, and I. Pieterse-Quirijns, "Review of applications for SIMDEUM, a stochastic drinking water demand model with a small temporal and spatial scale," *Drinking Water Engineering and Science*, vol. 10, Nr. 1, S. 1–12, 2017.
- [4] S. G. Buchberger und G. J. Wells, "Intensity, Duration, and Frequency of Residential Water Demands," *ASCE Journal of Water Resources Planning and Management*, vol. 122, Nr. 1, S. 11–19, Jan. 1996.
- [5] Foekema, H., and Engelsma, O. "A changing consumption pattern (water use at home 2001)", TNS NIPO, Amsterdam in Dutch, 2001.
- [6] Van der Broek, A., and Breedveld, K. "Time-budget survey 1995"- TBO'95 computer file, Sociaal en Cultureel planbureau, Amsterdam in Dutch, 1995.
- [7] CBS. 2007. <http://statline.cbs.nl> accessed on Dec. 1, 2007.
- [8] Klise, K.A., Bynum, M., Moriarty, D., Murray, R. "A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study", *Environmental Modelling and Software*, 95, 420-431, 2017.
- [9] Steffelbauer, D. B., Blokker, E. J. M., Buchberger, S. G., Knobbe, A., and Abraham, E. "Dynamic time warping clustering to discover socioeconomic characteristics in smart water meter data." *ASCE Journal of Water Resources Planning and Management*, 147(6), 04021026. 2021.