



This is a self-archived – parallel-published version of an original article. This version may differ from the original in pagination and typographic details. When using please cite the original.

AUTHORS Sajad Shahsavari, Ashvinkumar Chaudhari, Eero Immonen, Hashem Haghbayan

TITLE An Open-Source Framework For CFD-based Digital Twins: A Case Study on Storm Water Management

YEAR 2025

DOI <https://doi.org/10.7148/2025-0209>

VERSION Publisher's PDF

CITATION

Sajad Shahsavari, Ashvinkumar Chaudhari, Eero Immonen, Hashem Haghbayan (2025). An Open-Source Framework For CFD-based Digital Twins: A Case Study on Storm Water Management, ECMS 2025, Proceedings Edited by: Marco Scarpa, Salvatore Cavalieri, Salvatore Serrano, Fabrizio De Vita, European Council for Modelling and Simulation. doi:10.7148/2025-0209

# AN OPEN-SOURCE FRAMEWORK FOR CFD-BASED DIGITAL TWINS: A CASE STUDY ON STORM WATER MANAGEMENT

Sajad Shahsavari<sup>1,2,\*</sup>, Ashvinkumar Chaudhari<sup>1,\*</sup>, Eero Immonen<sup>1</sup>, and Hashem Haghayan<sup>2</sup>

<sup>1</sup>Computational Engineering and Analysis, Turku University of Applied Sciences

<sup>2</sup>Department of Computing, University of Turku

\*corresponding authors, emails: [sajad.shahsavari@turkuamk.fi](mailto:sajad.shahsavari@turkuamk.fi) & [ashvin.chaudhari@turkuamk.fi](mailto:ashvin.chaudhari@turkuamk.fi)

## ABSTRACT

Digital Twins (DTs) are increasingly applied for optimization of operations in logistics, healthcare, smart cities, and beyond. However, implementing high-fidelity DTs remains challenging in computationally intensive domains such as Computational Fluid Dynamics (CFD). While simplified models can facilitate real-time operation, they often lack physical fidelity. This article presents an open-source scalable software framework along with a case study of CFD-based digital twinning on stormwater management. The presented framework enables online execution of CFD-based models by containerizing and integrating them into OpenShift platform, providing a two-way communication channel for simulation parameters and results. The framework is capable of dynamically scaling computing resources to run computationally-intensive CFD-models. In the case study, we present a novel CFD simulation model of a bioretention cell intended to reduce runoff volumes of urban stormwater. The simulation model, implemented in OpenFOAM, is then integrated into the presented software framework to create the DT. The framework source code, simulation model and the DT are made publicly available to promote future research.

## KEYWORDS

Computational Fluid Dynamics, Digital Twins, Stormwater Management, OpenFOAM, Django.

## 1 INTRODUCTION

Digital Twins (DT) — simulation models running alongside their physical counterparts in real-time — are increasingly being adopted in the Industry 4.0 era (Batty, 2018; Mihai et al., 2022). Initially developed for industrial manufacturing, their application has expanded to other domains, including logistics, healthcare, smart cities and beyond (Atalay et al., 2022; Kaiblinger and Woschank, 2022; Turab and Jamil, 2023; Immonen et al., 2024). The adoption of digital twins is primarily driven by their potential to enhance decision-making processes for non-expert operators by providing predictive insights into system behavior during run-time.

Despite these advantages, in practice, DTs are often based on simpler modeling methodologies, such as lumped-parameter systems (Ritto and Rochinha, 2021). While computationally efficient — model execution on a single core is often enough — such methodologies often yield limited accuracy and an incomplete representation of system dynamics. Implementing *high-fidelity* DTs to accurately and fully represent the dynamics of the physical system remains a challenge, particularly in computationally intensive applications such as Computational Fluid Dynamics (CFD). Solution of high-resolution CFD models, which can incorporate several interacting physical phenomena, requires significant computational resources, and, typically, parallel computing. This makes real-time CFD model execution alongside their physical counterparts particularly demanding (Batty, 2018). This article is concerned with execution and user interaction for CFD-based DTs requiring parallel execution.

Modern DT solutions are predominantly offered by proprietary software vendors, leading to a reliance on closed-source implementations (Minerva et al., 2020). Open-source solutions are also available, as recently reviewed by Gil et al. (2024). However, the ones included in their review primarily focus on simulation co-execution and data connectivity, without addressing the scalability and computational challenges, such as large-scale CPU parallelization, associated with high-fidelity simulation models. There are also open-source DT frameworks that directly facilitate CFD model execution. In the MCX framework due to Shahsavari et al. (2021), the CFD model can be wrapped in a Functional Mock-up Unit (FMU) and accessed via Functional Mock-up Interface (FMI). Among the most widespread CFD codes, ANSYS Fluent and OpenFOAM (and others) support the FMI/FMU standard. The scalability of the MCX framework is not addressed by the authors, but it includes a message queue facilitating asynchronous (non-real-time) model execution. Another alternative solution, employed by Molinaro et al. (2021) is to pre-calculate the computationally expensive CFD cases and make use of supervised machine learning to reconstruct or interpolate simulation data and create physics-based data-driven models. While the computational cost of the CFD models is thus circumvented, the downside is that such models

can typically only represent simple low-parameter input-output relationships learned through the CFD case results (Immonen, 2022). Finally, there are custom DT solutions (see e.g. Shah et al. (2022)) that are designed for a specific CFD application only.

This article presents an open-source, cloud-native software framework for developing and deploying digital twins based on computationally-intensive CFD models. The framework enables users to interactively configure and execute CFD simulations online, offering a unique combination of live simulation access, computational resource management, and web-based usability. Importantly, the framework is designed for end users who lack CFD or programming expertise—such as planners, architects, or sustainability consultants—allowing them to test and analyze system designs virtually. Furthermore, in the case study application, we present a novel CFD simulation model of a bioretention cell as one of the most effective ways to sustainably manage urban stormwater by reducing runoff volumes and promoting infiltration, evaporation, and evapotranspiration (Lisenbee et al., 2021). Such management is critical since impervious surfaces have increased significantly due to urbanization, leading to elevated stormwater runoff, frequent flooding, and degraded water quality in catchment areas. A pilot-scale bioretention cell has been implemented in the city of Turku, Finland (Nyman, 2024), making this case study practically relevant. However, designing and optimizing such systems requires understanding complex flow and infiltration behavior that simplified tools like SWMM (Rossman, 2010) cannot capture reliably. This makes high-resolution CFD simulations valuable for evaluating design performance under different conditions. Yet, most professionals involved in stormwater management planning—such as urban designers or municipal officials—lack the expertise or resources to run CFD models independently. This is where our developed DT offers a significant advantage by embedding CFD simulations within an intuitive, live web interface. Thus, the DT allows non-expert users to explore how design parameters such as soil porosity, inflow rate, etc., can affect system behaviour and performance. The DT provides a robust yet accessible workflow for evaluating key outcomes such as infiltration, overflow, and hydraulic response under varying input conditions. The contributions of this work can be summarized as follows:

- An open-source software framework to execute the computationally-intensive CFD simulation models on the cloud,
- A novel CFD simulation model of stormwater dynamics in the bioretention cells,
- An online digital twin created by integration of the CFD model into the software framework.

## 2 PROPOSED DIGITAL TWIN

The proposed digital twin in this paper is composed of two main parts: 1) the general software framework for running the Computational Fluid Dynamics (CFD) simulation model online, and 2) the underlying CFD simulation

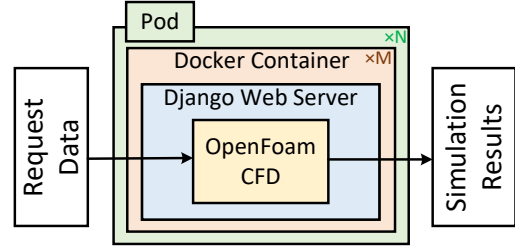


Figure 1: Structure of the running pod.

model of the stormwater dynamics in the bioretention cell. In this section, we describe the two parts in detail.

### 2.1 The Software Framework

The proposed framework is a scalable software to create online digital twins from computationally-intensive CFD models. The framework is essentially composed of 1) the placeholder of the underlying CFD model that represents the dynamics of the physical system/process, 2) a web server to provide data communication channels, 3) web interface to provide facilities for non-expert users to run the simulation model and monitor the behaviour of the system, and 4) Application Programming Interface (API) for external data sources to control/manipulate the digital environment.

The proposed framework is built on top of *OpenShift* OKD (2024), a container orchestration platform, and enables distributed execution of multiple simulation models, performs the traffic balancing between the running nodes, and handles access to the simulation results and external data sources. The core of the framework is the running *pod* which hosts the underlying CFD models to serve external requests. Fig. 1 depicts the structure of the pod. Each simulation model is executed within a web server, enabling it to establish two-way data communication channels with external data sources and end users. The web server is then encapsulated into a Docker container Docker (2024) providing a standalone executable package including the code, model and all the required dependencies to run the application.

The pod, composed of multiple containers, is then integrated into OpenShift. Fig. 2 shows the architecture of the proposed framework. The running pods containing underlying CFD models share a storage unit Volume to be able to permanently store the processing data and/or model parameters. Since pods are supposed to be dynamic, i.e., be able to be disposed and created at any given time by the deployment/maintenance process, they may have variable IP addresses. To compensate this variability, a Service component manages access to the running pods through a unified internal IP address that does not change during the pod creation or disposal. The Route component connects the application to the Internet, providing access for external data sources and end users. External data sources are the data repositories that can be used in the CFD model such as the state of the physical system of

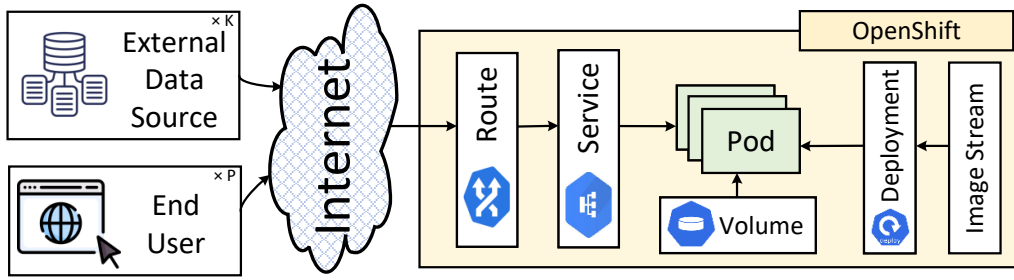


Figure 2: Overall structure of the software framework, integrating the DT running environment, external data sources from the physical entity and web access for non-expert end-users.

the DT, model parameters or arbitrary action values. End users are the non-expert users that can access the results of the simulation online via a web interface.

The Deployment component handles the pod failures and ensures that the latest version of the simulation model is being executed in multiple pods. This is performed by monitoring the state of the pods and updates of the simulation model container for new changes. The new versions of the simulation model are pushed into the Image Stream component.

### 2.1.1 Implementation

The proposed framework is implemented on Finland's national supercomputer, namely CSC—IT Center for Science. Specifically, we utilize the *Rahti 2 container cloud* system CSC (2024) which is a generic platform to create and deploy scalable and fault tolerant applications such as complex scientific software stacks and data analysis pipe lines.

## 2.2 The CFD Model of Bioretention Cell

In this section, we describe the mathematical foundation and technical details of the developed CFD model to represent the storm water dynamics in a bioretention cell. Figure 3 demonstrates a sample bioretention cell and the process of water infiltration into soil layers, recharge groundwater, and channel excess water through under-drain systems.

### 2.2.1 Overview of the Flow Equations and Solver

The CFD model is based on the Navier-Stokes equations for multiphase flows, utilizing the Volume of Fluid (VOF) approach. These equations are solved using the OpenFOAM platform, which is an open-source, C++-based computational framework designed for solving fluid flow problems (Greenshields, 2019). The presented CFD application employs the *hybridPorousInterFoam* solver, developed by Carrillo et al. (2020), specifically for simulating two-phase flows in hybrid porous media. The *hybridPorousInterFoam* solver is known to work under OpenFOAM versions 7 and 8. The solver has been extensively validated for various applications, including free-surface flow scenarios (Carrillo et al., 2020). Readers seeking

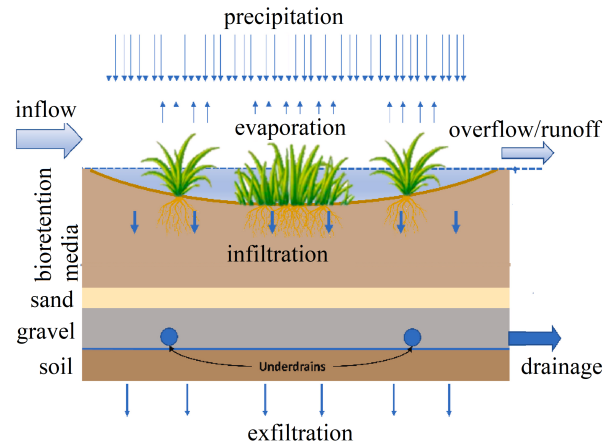


Figure 3: Schematic of a typical bioretention cell (adapted from Lisenbee et al. (2021)). Blue arrows indicate hydrologic pathways.

further details on the implementation of governing equations, VOF method, and porous media models are referred to Carrillo et al. (2020).

### 2.2.2 CFD Model Configurations

The configurations of the implemented bioretention cell are shown in Figure 4 with dimensions and boundary conditions (BCs). The length ( $L$ ) and height ( $H$ ) of the computational domain is set to 10 m and 3 m, respectively. The diameter of the inlet and outlet pipes is 10 cm. The lowest 0.53 m of the domain height (i.e., below the inlet pipe) is filled with 3 soil layers. The CFD grid resolution is fixed to 5 cm along the horizontal direction, while the vertical resolution varies from 2 to 15 cm.

The simulation accounts for two fluid phases: water and air. Water enters the cell only through the inlet pipe, located on the left side (blue arrow in Figure 4), and users can specify water flow rate. Air enter or leave the domain through those 3 open boundaries (red boundaries in Figure 4) depending on the pressure solution. The lower and side boundaries (below the height of 0.6 m; indicated by black color in Figure 4) are considered solid walls (no-slip BCs). Therefore, no water or air flow is allowed to pass through these boundaries. Consequently, water is forced to exit the domain either through the outlet pipe

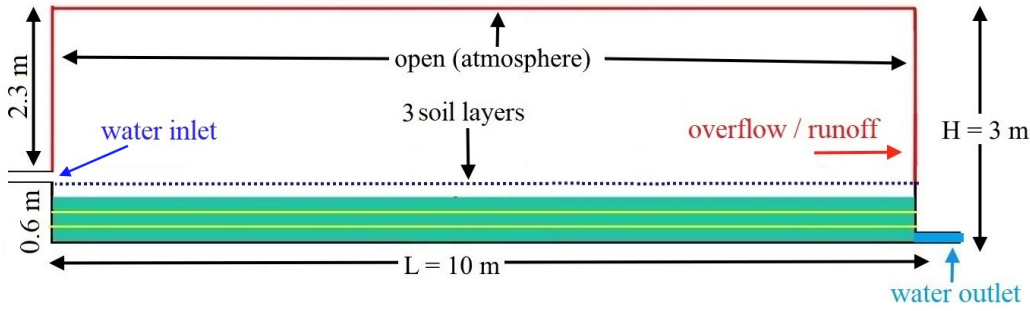


Figure 4: CFD model of the simulated bioretention cell with dimensions and the locations of inlet and outlet pipes.

(i.e., infiltration) or via surface overflow (i.e., runoff) on the right-side open boundary.

### 2.2.3 Model's Input Parameters and User Control

It is essential to assign the required BCs and other model inputs in a correct way to ensure meaningful results. To minimize user-input-oriented errors and simplify the process for non-expert users, the present application allows control of only a limited set of critical parameters. Although default parameter values are pre-configured, the user can adjust specific inputs for a set of parameters, such as water inflow rate and soil-layer porosity, within pre-defined ranges. Table 1 lists the most critical parameters with their default values and a clarification on whether a user can control it or not. The table also shows the range of parameter in case if a user is able to change that parameter. This design ensures the model remains user-friendly and error-free. By providing control over key parameters, users can optimize the design and performance of bioretention systems according to their specific requirements.

## 3 RESULTS

The developed DT is hosted on CSC's Rahti server and is accessible at <https://lknbs.rahtiapp.fi/>. Through the web server, users can input their desired parameters and submit simulation jobs. The server then executes the operations outlined in Figure 2, including running the simulation and displaying intermediate results on the web interface every 10 s of flow calculation time. The running CFD model generates data for various flow quantities over the course of a simulation. For demonstration purposes, the web server displays the most critical results including: 1) the water volume fraction across the domain, 2) the time series of water flows at a few locations: inlet, outlet and runoff. These outputs are essential for evaluating the cell's performance. The water volume fraction, a dimensionless quantity ranging from 0 to 1, indicates the proportion of water present in the cell, with 1 representing 100% water and 0 indicating the absence of water (i.e., air only).

To illustrate the application of the DT, Figure 5 presents the CFD results for two simulation scenarios, detailed in Table 2. The two scenarios represent contrasting condi-

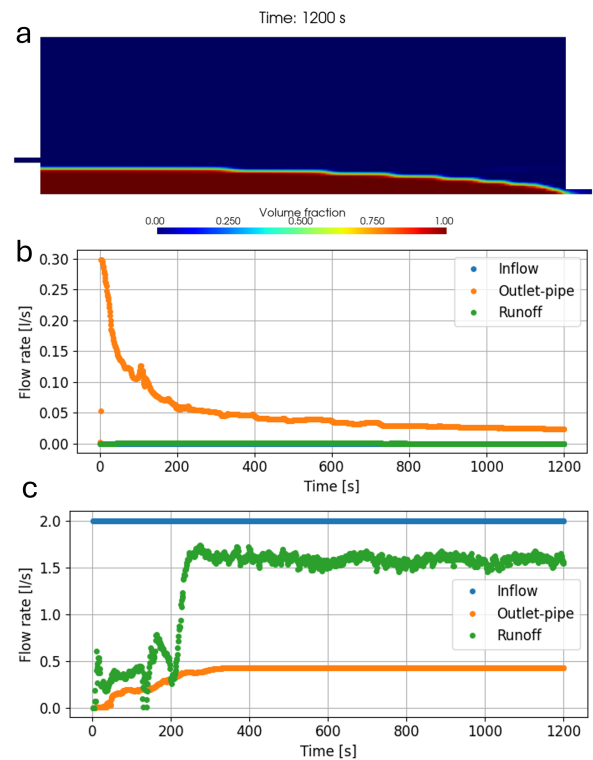


Figure 5: CFD model results for the the simulated Scenarios 1 (a and b) and 2 (c). The flow rate unit is liter per second (l/s).

tions. In Scenario 1, the cell is initially fully saturated with water (water volume fraction = 1), and no additional water is introduced (water inlet is off). In this case, gravity causes the water to drain through the outlet pipe over time, leading to the gradual emptying of the cell. Figures 5(a) and 5(b) show the results for Scenario 1 at a simulation time of 1200 seconds. In Scenario 2, water is introduced into the cell at a flow rate of  $2 \text{ m}^3/\text{s}$ , with an initial water fraction of 1% (indicating that the cell is 99% filled with air). This setup allows water to flow over the soil layers, creating small ponding and wave patterns, while slowly infiltrating into the soil. As a result, as shown in Figure 5(c), both runoff and outlet-flow rates remain low approximately for the first 250 s. Once the cell becomes fully saturated, the runoff rate increases sharply to approx-

Table 1: A list of some necessary parameters for the CFD model application and the web server (DT). *Switch for the water inlet*: A binary flag (1: on, 0: off) to control whether water enters through the inlet.

Parameter	Given or default values	Can a user change it?	Range of the input (if allowed)
Domain size ( $L \times H$ )	$10 \times 3 \text{ m}^2$	No	-
CFD model resolution	5 cm & 2-15 cm	No	-
Simulation time-step	automatic (CFL condition)	No	-
Water inflow rate	$0.001 \text{ m}^3/\text{s}$	yes	0 – 0.002
Switch for the water inlet	1 (yes)	yes	either 0 (no) or 1 (yes)
Porosity of the top soil layer	0.40	yes	0.05 – 0.99
Porosity of the middle soil layer	0.34	yes	0.05 – 0.99
Porosity of the bottom soil layer	0.43	yes	0.05 – 0.99
Initial volume fraction of water	0.5 (= 50%)	yes	0.01 – 0.99
Simulation time	300 s	yes	1 – 3600

Table 2: Input values for the simulated Scenarios 1 and 2.

	Scenario 1	Scenario 2
Water inflow rate ( $\text{m}^3/\text{s}$ )	0.000	0.002
Switch for the water inlet	0	1
Porosity of the top layer	0.4	0.4
Porosity of the middle layer	0.34	0.34
Porosity of the bottom layer	0.43	0.43
Initial water fraction	0.99	0.01
Simulation time	1200 s	1200 s

imately 1.6 l/s, with minor fluctuations due to ponding. Meanwhile, the outlet flow rate stabilizes at around 0.4 l/s. Together, these rates add up to the inflow rate of 2 l/s, maintaining mass balance.

The model is restricted to a two-dimensional (2D) representation to significantly reduce computational cost, making near-real-time interaction feasible on cloud infrastructure. While 2D models simplify complex hydrodynamic interactions, they still capture key design insights, especially for conceptual evaluation and education. Several assumptions are made to further streamline the analysis, such as excluding water losses due to evaporation and evapotranspiration. These losses typically occur over extended periods, ranging from days to weeks, depending on factors like solar radiation and soil properties (Heck et al., 2020).

The current implementation does not include formal model validation against field data from the Turku bioretention cell due to the unavailability of data. However, the *hybridPorousInterFoam* solver used in this study has been validated in prior studies for similar subsurface flow conditions (Carrillo et al., 2020). The goal of this prototype is not rigorous scientific validation but to offer an operational and trusted simulation interface for non-expert usage. Future work will focus on comparative validation studies to enhance confidence in predictive capabilities.

#### 4 CONCLUSION AND FUTURE DIRECTION

This paper presents an open-source, cloud-native framework for executing CFD-based digital twins along with

a novel CFD model to represent the water dynamics in a bioretention cell. The system combines the OpenFOAM solver for the CFD modeling with Django, Docker, and OpenShift software platform to support scalable, on-demand simulation execution in an accessible digital environment. This work uniquely integrates a live simulation engine with a user interface and a flexible backend for scaling. It bridges the gap between high-fidelity CFD modeling and real-world decision-making by offering a working proof-of-concept platform that the research community can replicate or adapt to their domains. Furthermore, the DT is designed particularly to serve non-expert users—such as stormwater planners or municipal consultants—who typically lack the skills or resources to run CFD models independently. The framework, simulation model and developed DT were all made accessible online to inspire further development of similar DT tools in industrial and environmental fields where professionals need simulation insights but lack access to complex modeling environments. Note that even though the case study of stormwater management particularly focused on a bioretention cell simulation demonstrating the practical relevance and usability of the framework, the proposed architecture for the software framework is generally applicable in other complex applications as well.

Future work will focus on several directions: integrating automatic actuators for closed-loop physical control, extending support for 3D and multi-domain simulations, adding performance benchmarking for scalability assessment, and incorporating field data for validation. Additional use cases across other domains will also be explored, further demonstrating the framework’s versatility beyond stormwater applications.

#### Acknowledgement

This work was supported by the Ministry of Education and Culture of Finland, funded project ISMO – Intelligent and sustainable stormwater management. We gratefully acknowledge the Center for Scientific Computing (CSC) for providing computational resources and hosting support through the Rahti server.

## Source Code and Model

The source code of the proposed framework is publicly accessible via <https://github.com/chaudhariashvin/OpenFoamDjango>. The Docker image for the OpenFoam simulation model can be found in [https://hub.docker.com/r/ashvinc9/of\\_lknbs](https://hub.docker.com/r/ashvinc9/of_lknbs)

## References

- Murat Atalay, Ugur Murat, Busra Oksuz, Ayse Merve Parlaktuna, Erhan Pisirir, and Murat Caner Testik. Digital twins in manufacturing: systematic literature review for physical–digital layer categorization and future research directions. *International Journal of Computer Integrated Manufacturing*, 35(7): 679–705, 2022.
- Michael Batty. Digital twins, 2018.
- Francisco J. Carrillo, Ian C. Bourg, and Cyprien Soulaïne. Multiphase flow modeling in multiscale porous media: An open-source micro-continuum approach. *Journal of Computational Physics: X*, 8:100073, 2020. ISSN 2590-0552.
- CSC. Rahti 2 - docs csc. <https://docs.csc.fi/cloud/rahti2/>, 2024. Accessed: 2024-03-12.
- Docker. What is a container? — docker. <https://www.docker.com/resources/what-container>, 2024. Accessed: 2024-03-12.
- Santiago Gil, Peter H. Mikkelsen, Cláudio Gomes, and Peter G. Larsen. Survey on open-source digital twin frameworks—a case study approach. *Software: Practice and Experience*, 54(6):929–960, 2024. doi: <https://doi.org/10.1002/spe.3305>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3305>.
- Christopher Greenshields. *OpenFOAM v7 User Guide*. The OpenFOAM Foundation, London, UK, 2019.
- K. Heck, E. Coltman, J. Schneider, and R. Helmig. Influence of radiation on evaporation rates: A numerical analysis. *Water Resources Research*, 56(10): e2020WR027332, 2020.
- Eero Immonen. Optimal design for disc golf by computational fluid dynamics and machine learning. *Structural and Multidisciplinary Optimization*, 65(1):12, 2022.
- Eero Immonen, Tero Villman, Michael Lindholm, and Jari Kaivo-oja. Modular smart city digital twins: A survey of key technologies. In Cezary Biele, Janusz Kacprzyk, Wiesław Kopeć, Jakub Możaryn, Jan W. Owsiański, Andrzej Romanowski, and Marcin Sikorski, editors, *Digital Interaction and Machine Intelligence*, pages 316–326, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-66594-3.
- Alexander Kaiblinger and Manuel Woschank. State of the art and future directions of digital twins for production logistics: a systematic literature review. *Applied Sciences*, 12(2):669, 2022.
- Whitney A. Lisenbee, Jon M. Hathaway, Matthew J. Burns, and Tim D. Fletcher. Modeling bioretention stormwater systems: Current models and future research needs. *Environmental Modelling & Software*, 144:105146, 2021. ISSN 1364-8152.
- Stefan Mihai, Mahnoor Yaqoob, Dang V Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V Prasad, et al. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys & Tutorials*, 24(4):2255–2291, 2022.
- Roberto Minerva, Gyu Myoung Lee, and Noel Crespi. Digital twin in the iot context: A survey on technical features, scenarios, and architectural models. *Proceedings of the IEEE*, 108(10):1785–1824, 2020.
- Roberto Molinaro, Joel-Steven Singh, Sotiris Catsoulis, Chidambaram Narayanan, and Djamel Lakehal. Embedding data analytics and cfd into the digital twin concept. *Computers & Fluids*, 214:104759, 2021.
- Emil Nyman, editor. *Intelligent and Sustainable Stormwater Solutions: Results from the ISMO project 2022–2024*, volume 304 of *Turun ammattikorkeakoulun raportteja*. Turku University of Applied Sciences, Turku, Finland, 2024. ISBN 978-952-216-882-5. URL <https://urn.fi/URN:ISBN:978-952-216-882-5>.
- OKD. Home — okd documentation. <https://docs.okd.io/>, 2024. Accessed: 2024-03-12.
- TG Ritto and FA Rochinha. Digital twin, physics-based model, and machine learning applied to damage detection in structures. *Mechanical Systems and Signal Processing*, 155:107614, 2021.
- Lewis A Rossman. Modeling low impact development alternatives with swmm. *Journal of Water Management Modeling*, 2010.
- Karan Shah, T. V. Prabhakar, Sarweshkumar C. R., Abhishek S. V., and Vasanth Kumar T. Construction of a digital twin framework using free and open-source software programs. *IEEE Internet Computing*, 26(5):50–59, 2022. doi: 10.1109/MIC.2021.3051798.
- Sajad Shahsavari, Eero Immonen, Mohammed Rabah, Mohammad Hashem Haghbayan, and Juha Plosila. Mcx— an open-source framework for digital twins. In *ECMS*, pages 119–124, 2021.

Muhammad Turab and Sonain Jamil. A comprehensive survey of digital twins in healthcare in the era of metaverse. *BioMedInformatics*, 3(3):563–584, 2023.

**Sajad Shahsavari** works as Researcher at Turku University of Applied Sciences, and is a PhD candidate in Department of Computing, University of Turku, Finland.

**Ashvin Chaudhari**, D.Sc.(Tech.), is working as a Senior Research Advisor at the COMEA research group, Turku University of Applied Sciences (Finland). He studied Fluid Dynamics and Applied Mathematics and received his doctoral degree in Technology from LUT University (Finland) in 2014. Email: [Ashvin.Chaudhari@turkuamk.fi](mailto:Ashvin.Chaudhari@turkuamk.fi)

**Eero Immonen** is an Adjunct Professor at Department of Mathematics at University of Turku, Finland, and works as Principal Lecturer at Turku University of Applied Sciences, Finland.

**Hashem Haghbayan** (Member, IEEE) is an Adjunct Professor (docent) in embedded intelligent systems at the Department of Computing, Faculty of Technology, University of Turku (UTU), Finland. He received his M.Sc. in computer architecture from the University of Tehran, Iran, and Ph.D. with honors from the University of Turku, Finland. From 2018 to 2021, he has been a Postdoctoral Researcher with the Department of Computing, Faculty of Technology, University of Turku.