Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

WaCSim: A Water and Cyber Digital Twin for Enhancing Cyber Security in Smart Water Systems

Stav Cohen[1] | Tyler Trimble[3] | Lina Sela[3] | Mashor Housh[2] | Barak Fishbain*[4]

[1]Faculty of Data and Decision Sciences, Technion - Israel Institute of Technology, Israel
[2]Faculty of Social Sciences, School of Environmental Sciences, University of Haifa, Israel
[3]Faculty of Civil, Environmental and Architectural Engineering, The University of Texas at Austin, USA
[4]Faculty of Civil and Environmental Engineering, Technion - Israel Institute of Technology, Israel

Correspondence
*Barak Fishbain. Email: fishbain@technion.ac.il

Present Address
616 Rabin Hall, Technion City, Haifa, 320003, Israel

Abstract

Smart Water Distribution Systems (WDSs) are increasingly exposed to cyber-attacks due to their growing connectivity and reliance on digital control. To support the design of resilient infrastructures, we introduce WaCSim (Water and Cyber Simulation), a cyber-physical simulation platform built to support AI integration, model development, and the testing of defensive strategies. WaCSim functions as a flexible data platform, enabling users to simulate realistic WDS configurations and analyze system behavior under both normal operations and adversarial conditions.

WaCSim supports three key control architectures: (a) a decentralized architecture, where local actuation and communication occur via networked Programmable Logic Controllers (PLCs); (b) a centralized architecture, where a Supervisory Control and Data Acquisition (SCADA) system centrally gathers data and issues commands; and (c) a hybrid architecture, combining centralized oversight with local PLC-to-PLC coordination, allowing for conflict scenarios and redundancy. This range of modes allows users to test how architectural design affects system resilience.

In addition, WaCSim provides a Python interface for integrating custom algorithms at both the SCADA and PLC levels. Users can implement anomaly detection models, optimization routines, AI-based control logic, and Conflict Resolution Algorithms (CRAs) enabling in-simulation testing of cyber defenses and adaptive responses.

Through case studies, we demonstrate how WaCSim models cyber-attacks, evaluates defense mechanisms, and supports the development of intelligent control systems. These examples illustrate WaCSim's utility as a research and development platform for advancing the robustness, security, and operational intelligence of smart water networks.

KEYWORDS:
Water Distribution System, Digital Twin, Cyber Security, Artificial Intelligence

[0]**Abbreviations:** CRA, Conflict Resolution Algorithm; DoS, Denial of Service; ICT, Information and Communication Technology; MiTM, Man-in-the-Middle; MSF, Main Simulation File; PLC, Programmable Logic Controller; SCADA, Supervisory Control and Data Acquisition; WDS, Water Distribution System

1 | INTRODUCTION

Water distribution systems (WDSs) are a fundamental part of urban infrastructure, responsible for delivering potable water. Traditionally, these systems were managed manually, requiring significant human effort and being prone to errors and inefficiencies. As demand and scale grew, planning, building, and optimizing the infrastructure of these systems became increasingly challenging, necessitating the development of computerized tools for planning and management (Bao & Mays 1990; Mala-Jetmarova, Sultanova, & Savic 2017-07; Montalvo, Izquierdo, Pérez-García, & Herrera 2014; Perelman & Fishbain 2022; Perelman, Ostfeld, & Fishbain 2023; Vairavamoorthy & Ali 2000).

The evolution of WDSs into smart water systems has integrated control technologies such as Programmable Logic Controllers (PLCs) and Supervisory Control and Data Acquisition (SCADA) systems. PLCs are specialized computers that are used to automate industrial processes by receiving inputs from sensors, processing them according to a programmed logic, and controlling outputs to actuators. Typically, a specific PLC controls a single network element based on data that is received by a limited, local, set of sensors. SCADA systems provide real-time data collection, remote monitoring, and automated or manual control over the entire WDS (Franco et al. 2019; Saravanan, Anusuya, Kumar, & Son 2018). The integration of SCADA with Information and Communication Technology (ICT) enhances water management by enabling bi-directional communication and intelligent control of water flow (Seung Won Lee & Kim 2015). Moreover, the use of advanced sensor networks and real-time monitoring significantly contributes to reducing water losses and improving the overall sustainability and efficiency of water systems (Ramos et al. 2023). Additionally, the data gathered by SCADA systems can be leveraged via AI and optimization algorithms. This, in turn, results in optimized design of the monitored networks, detect water losses, guide energy savings, forecast demand, and improve asset management (Jenny, Alonso, Wang, & Minguez 2020).

Nonetheless, the increasing complexity and interconnectivity of these smart water systems introduce significant cyber security challenges. With more ICT integration, these systems become more sophisticated, they also become more vulnerable to malicious attacks, which can threaten the reliability and safety of critical water infrastructures (Li, Yang, & Sitzenfrei 2020; Taormina, Galelli, Tippenhauer, Salomons, & Ostfeld 2017). To address these challenges, significant efforts have been made to develop effective detection mechanisms. One notable initiative is the Battle of the Attack Detection Algorithms (BATADAL) (Taormina et al. 2018), which provided a platform for comparing various algorithms designed to detect cyber-physical attacks on water distribution systems. The competition showcased the effectiveness of diverse approaches, from model-based methods leveraging hydraulic modeling (Housh & Ohar 2018), to machine learning techniques that incorporate both spatial and temporal data (Tsiami & Makropoulos 2021), and self-supervised and unsupervised learning (Mahmoud, Wu, & Gaber 2022). Recent work has further integrated detection with attack localization and severity assessment (Raza & Moazeni 2025). These collective efforts highlight the potential of anomaly detection algorithms in enhancing the security of WDSs, ensuring their resilience

against evolving cyber threats. However, with the absence of real-world records of WDS behavior under cyber attacks, all the aforementioned studies relied on benchmark networks, synthetic data, or simulated scenarios to address this gap (Fu, Jin, Sun, Yuan, & Butler 2022). Given these developments and challenges, there is a critical need for a comprehensive framework that guides the design and deployment of smart water systems, a need that is also reflected in recent calls for data-centric water engineering (Fu, Savic, & Butler 2024). This need is addressed by the concept of a Digital Twin (Boschert & Rosen 2016), which can simulate the real-world behavior of a smart water system without the necessity of building physical prototypes. Digital Twins offer numerous benefits, including process optimization, cost reduction, and reduced time to market, making them invaluable for water management. They enable the detection of leaks, testing of water assets under various constraints, and provision of autonomous and intelligent system control (Ramos et al. 2023; Zekri, Jabeur, & Gharrad 2022) .

To build effective digital twins and algorithmic approaches, there is a need for simulation frameworks that accurately model the hydraulic and water quality behavior of the WDSs. Hydraulics simulators, such as EPANET (Rossman et al. 2000) and WNTR (Klise, Murray, & Haxton 2018), were developed to focus primarily on the hydraulic and some water quality aspects of WDSs. However, these simulators did not include the cyber-network layer, which is crucial for understanding and mitigating different types of cyber-attacks on the WDS that can impact hydraulic behavior. Due to this limitation, several derivative tools have been developed to extend these capabilities. Notable examples include EPANETCPA (Taormina et al. 2019) and RISKNOUGHT (Nikolopoulos et al. 2020), which were designed to model cyber-attacks on WDSs by integrating basic cyber-physical components. Still, these tools do not provide full cyber-layer integration; they mainly dictate the anomalous hydraulic behavior of the water system under cyber-attack without modeling the detailed information traffic in the cyber-layer. That is, the output of these simulators is solely related to the water system.

Recently, Murillo et al. 2020, have introduced a simulation tool, Digital HydrAuLic SIMulator (DHALSIM), which combines EPANET-based simulations with a data network emulation environment. This integration of both hydraulic and cyber layers offers a more nuanced understanding of cyber-physical attacks. By simultaneously simulating the behavior of both layers, the tool can provide a detailed representation of reality, in which one can distinguish between different types of cyber-attacks that might otherwise appear similar when only hydraulic data is considered. For instance, Denial-of-Service (DoS) and Man-in-the-Middle (MiTM) both described in Section 2.5, can produce nearly identical hydraulic responses, making it difficult to identify the underlying cause based solely on physical observations when former simulators are used. However, by incorporating network traffic simulation these attacks can be effectively differentiated, leading to more accurate intrusion detection and better-targeted responses. The development of DHALSIM has clearly demonstrated the critical need for this type of holistic simulator. Yet, there are critical gaps that remain in the DHALSIM framework, as many important real-world features are missing. DHALSIM currently operates under the assumption of a single cyber-network configuration, where multiple PLCs are distributed across the network and connected to physical components like water pumps and storage tanks. These PLCs communicate with each other

to enforce simple control logic available in the EPANET model such as minimum volume in a tank. Although these PLCs are linked to a central SCADA system, the SCADA only passively monitors the actuators remotely, without having direct control over them.

While this setup exists in real-world systems, it is far from representing the wide range of possibilities for smart water systems configuration. This is especially true when simulating complex real-world setups that use a centralized control scheme, which often characterizes large-scale systems. Furthermore, in real systems that adopt centralized control, it is typical to maintain local schemes as a backup or for redundancy, consequently, there is a critical need for a simulation framework capable of supporting hybrid control architectures. In light of the above, we introduce a new tool, the *Water and Cyber Simulation - WaCSim*, which is built on and expands the capabilities of DHALSIM, serving as the next stepping stone towards complete synchronized simulations of cyber-water systems. Our primary contributions are twofold:

1. **Expanded Cyber-Network Configurations:** WaCSim introduces support for three control schemes, each representing a different mode of cyber-physical coordination:

    - **PLC Control:** Inherited the basis from DHALSIM and expanded, this mode supports decentralized decision-making, where PLCs operate autonomously and communicate directly with each other. The SCADA system, if present, remains passive and only observes network activity acting as a database.
    - **SCADA Control:** In this centralized scheme, the SCADA system collects data from all PLCs and issues real-time control commands using either predefined rules or user-defined algorithms. This reflects typical configurations in large-scale water systems.
    - **Hybrid Control:** This mode integrates both centralized and decentralized control logic, where PLCs process data and commands from both peer devices and the SCADA system. This configuration mirrors real-world systems that utilize centralized oversight alongside local fallback mechanisms. It enables the configuration of logic to determine the shift between these two modes and the handling of conflicting decisions, which are essential for studying system behavior under communication delays, hardware failures, or cyber-attacks.

    These configurations allow users to simulate a wide range of operational architectures and evaluate how control design influences resilience and behavior under cyber-physical conditions.

2. **Python Interface for Customizable Algorithms:** WaCSim provides a flexible Python-based interface that enables users to integrate custom algorithms at both the SCADA and PLC levels. This feature allows Users to override the default EPANET control rules with their own logic for tasks such as anomaly detection, optimization, and adaptive system control. By embedding user-defined code directly into the simulation loop, WaCSim supports the testing and evaluation of

intelligent operational strategies under a wide range of scenarios. Furthermore, this capability allows for modeling the specific countermeasures a system may take in response to detecting a cyber-attack or failure. Unlike previous simulators that primarily provide a passive observation of an attack's footprint, this interface makes it possible to simulate how the attack footprint evolves as a result of the active interventions and reactions taken by the system operator

A key use case for this interface is the implementation of Conflict Resolution Algorithms (CRAs) in Hybrid Control mode. In this mode, PLCs may receive simultaneous inputs from the SCADA and from peer PLCs. Under normal conditions, these inputs should align, but during a cyber-attack, WaCSim can simulate adversarial conditions where the two data streams are manipulated differently. For example, an attacker might tamper with SCADA-bound traffic while leaving peer-to-peer PLC communication intact (or vice versa), creating inconsistencies that the PLC must resolve. CRAs allow users to define custom decision logic for handling such conflicts.

Detailed instructions for using this interface are provided in the Methods section, as well as in the comprehensive user manual and tutorials on GitHub.

Additionally, the results and discussion section presents two case studies that demonstrate how WaCSim models cyber-attacks under varying control modes. They highlight how PLC, SCADA, and Hybrid configurations shape system resilience, and how custom algorithms such as control logic and conflict handling can mitigate disruptions caused by data manipulation.

2 | METHODS

In this section, we provide a detailed breakdown of WaCSim's components, guided by Figure 1, which presents an overview of the WaCSim architecture. The diagram is divided into several color-coded sections, each representing a major part of the simulation framework. A key visual distinction in the diagram is the use of solid versus dashed lines: solid lines represent features inherited from the original simulation framework, while dashed lines highlight new or heavily upgraded features introduced in WaCSim to support advanced cyber-physical modeling.

- **Input Files (Orange):** Represents the configuration files, including hydraulic network definitions, PLC configurations, control algorithms, and attack scenarios, all referenced in the Main Simulation File (MSF).
- **Core Simulator Engines (Light Blue):** Consists of two main components: the Hydraulic Simulator and the Cyber-Network Simulator. These two modules interact at each time step to emulate both the physical behavior of the water distribution system and the network communication between control units
- **Control Algorithms (Green):** Highlights the new Python-based interface for implementing custom Decision, Monitoring, and Conflict Resolution Algorithms (CRAs) at both the PLC and SCADA levels.

- **Output Files (Purple):** Indicates the storage of both physical hydraulic data and cyber-network traffic logs (e.g., .pcap files).

- **Simulation Modes (Blue):** Shows the three available operational modes: PLC, SCADA, and Hybrid, which determine the logic for control decisions.

Each of these components is described in detail in the following subsections.

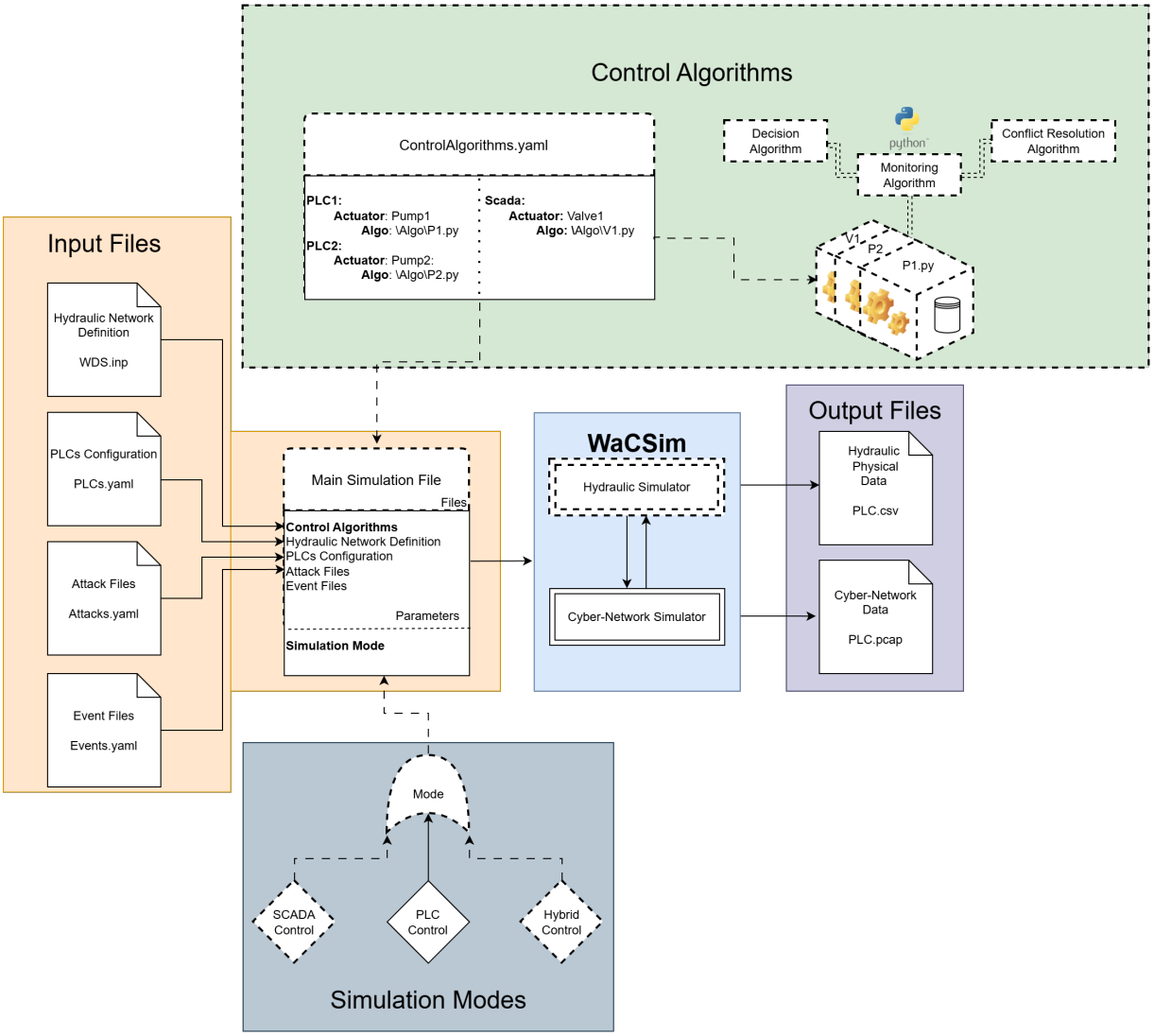

Figure 1 Architectural framework of WaCSim. Solid lines represent features inherited from the original simulation framework, while dashed lines highlight new or heavily upgraded features introduced in WaCSim to support advanced cyber-physical modeling.

2.1 | Simulator Engine

The core engine of WaCSim is built upon the integration of two simulators, which operate in tandem to simulate both the physical and cyber aspects of a water distribution system. These two components - *Hydraulic Simulator* and *Cyber-Network Simulator* - work together step-by-step to create a comprehensive model of WDS behavior under normal operations and during potential cyber-physical attacks.

2.1.1 | Hydraulic Simulator

The hydraulic component of WaCSim is primarily driven by EPANET version 2.2 (Rossman et al. 2000), an open-source software package used to simulate the hydraulic and water quality behavior of pressurized WDSs. Physical components modeled by EPANET include pipes, junctions, valves, pumps, and reservoirs. In addition to these physical components, there are several additional components such as demand patterns, pump curves, volume curves, and control rules. From all these components, EPANET can then simulate and report key parameters, such as pressure, demand, flow rate, and velocity.

While the EPANET software package includes a basic rule-based control system, WaCSim extends these capabilities by interfacing directly with the EPANET Toolkit. Compared to DHALSIM, which relies on intermediate Python wrappers such as WNTR (Klise et al. 2018), WaCSim's direct utilization of the Toolkit provides greater transparency and more granular control over simulation behavior. Furthermore, this architectural choice ensures that future enhancements are easier to implement and remain independent of the update cycles of third-party libraries. By working directly with underlying EPANET functions to override control rules and actuator states in real-time, WaCSim can accurately replicate the complex decision-making processes observed in real-world systems, as discussed in detail in Section 3.1.2.

2.1.2 | Cyber-Network Simulator

To simulate the cyber layer, WaCSim utilizes Mininet, a network emulation tool that creates realistic virtual networks on a single machine. As done in DHALSIM Murillo et al. (2020), WaCSim pairs Mininet with MiniCPS (Cyber-Physical System) Antonioli and Tippenhauer (2015), which enables the emulation of industrial control protocols such as Ethernet/IP and Modbus, commonly used in real-world SCADA systems. The cyber-network simulator is responsible for generating and managing the network traffic between the PLCs and the central SCADA server. This layer simulates the packet transfers, communication delays, and potential disruptions that can occur due to network anomalies or cyber-attacks. Through Mininet, WaCSim can emulate the interactions between the digital components and the hydraulic system, providing insights into how network performance impacts WDS operations.

To maintain synchronization between the hydraulic and cyber components, WaCSim runs in a step-by-step framework. Each simulation time step begins with EPANET calculating the physical behavior of the water network, such as flow rates, pressures,

and tank levels. This updated physical state generates sensor data, which is transmitted through the Mininet based cyber-network. The network simulation emulates how data is transmitted between devices, incorporating delays, packet loss, or manipulation as defined by the configured attack scenario.

PLCs and the SCADA system receive sensor data and control messages through the emulated network. Based on the data they obtain, each unit applies its control logic, which may be defined by simple EPANET-style rules or custom algorithms implemented by the user. The resulting control decisions such as changes to pump or valve states are then passed back to the hydraulic simulator, which updates the physical system state for the next simulation step.

2.2 | Network Files

WaCSim relies on a set of configuration files that define the structure and behavior of each simulation. These files include the hydraulic network description, the list of PLCs and their sensor - actuator mappings, user defined control algorithms, the simulation mode, and any cyber-attacks or events to be simulated. All of these are referenced by a central Main Simulation File (MSF), which serves as the entry point for running a scenario. In the following subsections, we explain the purpose and format of each file type, using a simplified example, referred to here as Eden-Town. This small network consists of two reservoirs (R1 and R2), two pumps (P1 and P2), two storage tanks (T1 and T2), and seven pipes connected through five junctions (J1, J2, J3, J4, and J5). An illustration of the network is shown in Figure 2.

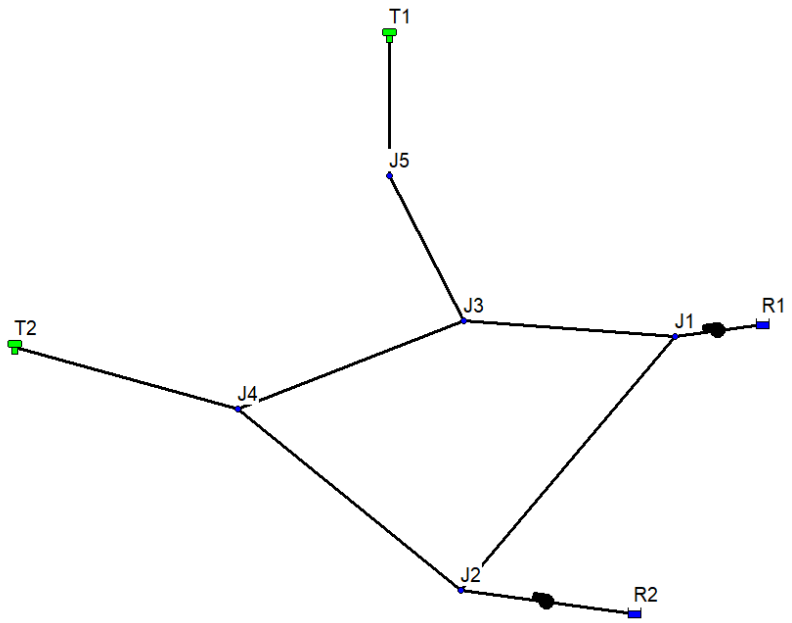

Figure 2 Eden Town layout.

2.2.1 | Main Simulation File (MSF)

The *MSF* is the core configuration required to run a WaCSim simulation. Users create this file to reference all other necessary files for the simulation. It acts as the central hub, linking all components together for the execution. Once the *MSF* is set up, it can be submitted to the Simulator Framework to run the simulation. Listing 1 shows an example of an *MSF* used for the Eden-town simulation. In the following sections, we review the files required to run the simulation, with further details on the necessary parameters provided in the GitHub manual.

```
1 - inp_file: EdenTown.inp
2 - plcs: !include EdenTown_plc.yaml
3 - mode: hybridcontrol
4 - Algo_plc: !include EdenTown_Algo_plc.yaml
5 - Algo_scada: !include EdenTown_Algo_scada.yaml
6 - attacks: !include Plc4_DoS.yaml
7 - events: !include PLC1_NetworkDelay.yaml
```

Listing 1: Main Simulation File (MSF) in *YAML* Format

2.2.2 | Hydraulic Network Definition (INP File)

The *INP file* defines the structure and operational parameters of the physical WDS, including details about network components such as pipes, pumps, valves, tanks, and junctions. The file format is the standard EPANET format for describing the hydraulic network. The *INP file* also contains important simulation settings and control rules for operating actuators, such as valves and pumps. In WaCSim, the control sections of this file, as shown in Listing 2 are particularly important as they specify the predefined control logic that governs the system. For example, in lines 2-3, Pump 2 (P2) operates based on the water level of Tank 2 (T2). These control rules can be overridden by the custom controls described in Section 3.1.2, where users can implement more sophisticated decision-making algorithms to handle complex scenarios.

```
1 [CONTROLS]
2 LINK   P2  OPEN    IF NODE T2  BELOW  3
3 LINK   P2  CLOSED  IF NODE T2  ABOVE  7
4 LINK   P1  OPEN    IF NODE T1  BELOW  6
5 LINK   P1  CLOSED  IF NODE T1  ABOVE  7
```

Listing 2: PLC Controls

2.2.3 | PLC Configuration

PLCs play a central role in WaCSim by linking the physical and cyber components of the water distribution system. Each PLC is assigned to specific sensors and actuators in the network, allowing it to monitor parameters such as tank levels, pump flows,

valve positions, and junction pressures. Sensing PLCs gather this data and transmit it across the network-either to peer PLCs or to the SCADA system-while actuator PLCs use the incoming information to control pumps, valves, and other physical elements. This division of roles enables both localized control, where decisions are made based on peer-to-peer communication, and centralized control, where decisions are coordinated by the SCADA system. The flexibility of this arrangement supports a wide range of operational strategies and control schemes in WaCSim

To configure PLCs in WaCSim, all PLCs are defined in a single YAML file declared in line 2 in Listing 1, specifying their names, the sensors they monitor, and the actuators they control. Sensors can track various system parameters, such as tank water levels, junction pressures, valve flows, and pump flows, while actuators allow PLCs to manage components like valve statuses and pump operations. For example, in the Eden-Town simulation, the *YAML* configuration shown in Listing 3 defines the PLCs along with their corresponding sensors and actuators. In this configuration, PLC-1 is responsible for controlling P2, while PLC-2 monitors the water level in Tank-1 (T1), and PLC-3 tracks T2. Furthermore, PLC-4 is tasked with controlling Pump 1 (P1).

```yaml
- name: PLC1
  actuators: P2
- name: PLC2
  sensors: T1
- name: PLC3
  sensors: T2
- name: PLC4
  actuators: P1
  sensors: P1F
```

Listing 3: PLC Configuration in *YAML* Format

2.3 | Simulation Modes

To enhance flexibility and customization within WaCSim, users are now provided with the ability to select from three distinct simulation modes: *PLC-Control*, *SCADA-Control*, and *Hybrid-Control*. This feature allows for the creation of more realistic simulations tailored to the specific WDS being designed or modified. In Listing 1, line 3, users can configure the desired simulation mode by choosing one of these options.

2.3.1 | PLC Control

In this mode, each configured PLC operates independently and is distributed across different locations within the network. Every PLC connected to a physical component initializes and begins collecting local sensor readings. These PLCs then transmit their local readings to other PLCs that depend on this data to operate according to the basic EPANET control rules. For example, the control rules in Listing 2 (line 2) indicate that the operation of P2 depends on the water level in T2. Since PLC-3 is physically

connected to T2 and holds the necessary sensor readings, PLC-1 (responsible for P2) must request this data from PLC-3 to execute its logic. This peer-to-peer communication allows PLC-1 to acquire the specific remote data it needs to start or stop the pump according to the system requirements.

Additionally, the PLCs transmit their local data to the main SCADA unit. It is important to note that in this mode, the SCADA unit only receives and records that transmitted data without interfering with the network's operation. While this setup reflects certain real-world systems, it does not account for scenarios where a SCADA system, situated in a central control center, actively manages the network to optimize performance and efficiency. This limitation highlights the need for simulation modes that incorporate centralized control by the SCADA system, which is addressed in the *SCADA-Control* and *Hybrid-Control* modes discussed in the following sections. Figure 3 illustrates the data flow in this mode, showing the interactions between the PLCs and the SCADA unit. In this diagram, dashed lines indicate data transfer, while solid lines represent control commands. The purple lines specifically highlight the data and control flow for PLC-1 and Pump-2, where the action is triggered by sensor data obtained by PLC-3 from Tank T2 and transferred to PLC-1.

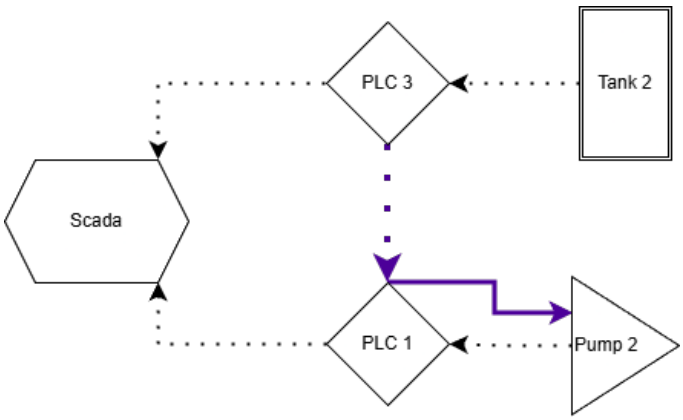

Figure 3 PLC Control diagram. Dashed lines represent data transfer, while solid lines represent control commands. Purple lines highlight the specific data and control flow between PLC-3, PLC-1, and Pump-2.

2.3.2 | SCADA Control

In this mode, the simulation transitions from decentralized to centralized control. The PLCs do not communicate with each other directly; instead, each PLC collects local sensor readings and transmits this data exclusively to the central SCADA system. The SCADA system, situated in a control center, aggregates data from all PLCs and processes the incoming information using the predefined rules from Listing 2 or custom algorithms. It then sends control instructions back to the respective PLCs that operate physical actuators.

For example, referring to Listing 2, line 2, we see that P2 depends on the water level of T2 to determine its operation. In *SCADA-Control* mode, PLC-3, which is connected to T2, sends the tank level readings directly to the SCADA system. The SCADA processes this information and decides whether P2 should start or stop. It sends the appropriate control command to PLC-1, which operates P2 accordingly. Figure 4 illustrates this process. This centralized control approach allows for more coordinated and optimized network operations. The SCADA system can make decisions based on the overall state of the network rather than isolated local data, enabling advanced control strategies and real-time adjustments to enhance performance and efficiency during both routine operations and emergencies.

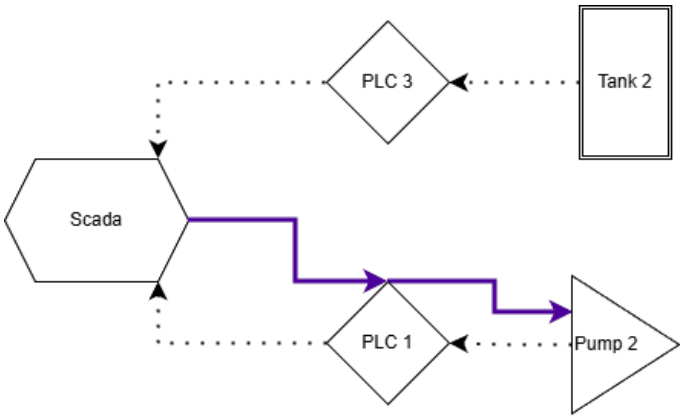

Figure 4 SCADA Control diagram. Dashed lines represent data transfer from PLCs to the central unit, while purple lines illustrate the active control commands issued by the SCADA system to the actuators.

2.3.3 | Hybrid Control

In this mode, WaCSim combines elements of both decentralized and centralized control to more closely mimic complex real-world water distribution systems. In this mode, PLCs send data not only to the SCADA system but also to other PLCs that rely on their sensor readings. Each PLC that operates an actuator receives input from dependent PLCs and simultaneously receives control commands from the SCADA system.

This dual-source input introduces the possibility of conflicting commands or data discrepancies (which may be caused by cyber-attacks, network errors or bad configuration), potentially leading to operational inconsistencies or malfunctions within the system. For instance, a PLC may receive a pump shutdown local control command based on data from another PLC, while also receiving a pump startup command from the SCADA system. To manage these conflicts, each PLC must implement a Conflict Resolution Algorithm (CRA).

The CRA enables the PLC to determine the appropriate action when faced with conflicting inputs. The resolution strategies can vary based on the system's requirements and can be configured (as described in the next section) as: *PLC Priority*: The PLC

always prioritizes commands based on inputs from other PLCs, effectively giving precedence to decentralized control. *SCADA Priority*: The PLC always follows the SCADA's commands, emphasizing centralized control. *Custom Control Algorithms*: Users can develop advanced CRAs using Python to define more sophisticated decision-making processes. Figure 5 illustrates the data flow in Hybrid Control mode.

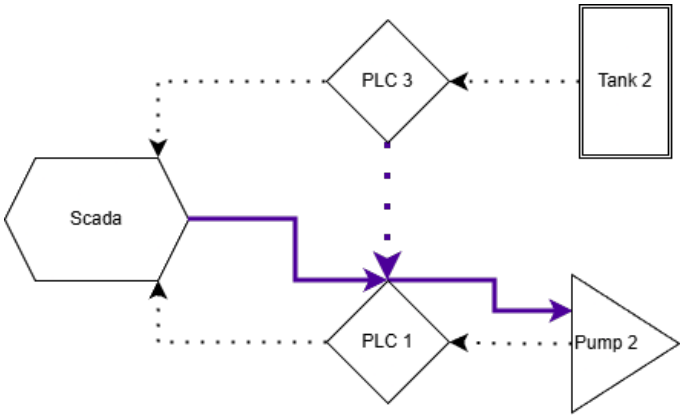

Figure 5 Hybrid Control diagram. Dashed lines represent data transfer between PLCs and the SCADA system, while purple lines illustrate the combined data and control flows that enable both local coordination and centralized oversight.

2.4 | Control Algorithms

Until now, existing simulators (e.g. DHALSIM) have only been capable of executing basic control rules, as illustrated in Listing 2. While these rules provide a functional baseline for simple operations, they represent an elementary decision-making process that fails to capture the multi-layered complexity and dynamic requirements of real-world industrial control systems. However, WaCSim expands this capability by enabling the integration of custom algorithms that can be deployed on both individual PLCs and on the SCADA system. Because custom algorithms override the default INP rules, users do not need to define control rules in the INP file for rules that will be controlled by the custom algorithm. This enhancement allows for more complex, data-driven decision-making, for both the local control scheme and centralized control where the SCADA system can access data from all distributed PLCs to provide a broader system perspective.

2.4.1 | PLC Algorithms

Unlike existing simulators that can only simulate basic PLC control rules which are defined in EPANET, one of the main features of WaCSim is its ability to integrate custom algorithms for local control at the PLC level. To implement custom algorithms, users need to create a *YAML* file that specifies how each PLC and actuator operates based on the data they receive. In Listing 1, line 4, we configure the algorithms for the PLCs, and Listing 4 presents an example of a PLC algorithm file.

In this example, we are running the simulation in Hybrid Control mode as defined in Listing 1. In lines 1-4 of Listing 4, PLC-1 operates P2 based solely on commands from the SCADA system. This configuration implements the SCADA priority mode, emphasizing a reliance on the central system's broader operational oversight. In this setup, the SCADA system can execute advanced, data-driven algorithms to generate and issue these control directives

Conversely, as shown in lines 5–8 of Listing 4, PLC-4 manages P1 using a custom Python algorithm. This control logic integrates inputs from three primary sources: (1) remote sensor data, such as the J1 pressure specified via the *dependents* field (see Listing 4, line 9); (2) local sensor data, such as P1 Flow, which is accessed directly without requiring network communication; and (3) centralized control commands and sensor values issued by the SCADA system, which are based on a system-wide perspective and may themselves be generated by a custom SCADA-level algorithm. These inputs allow PLCs to make more informed decisions, balancing decentralized sensor data with centralized insights from the SCADA system. Moreover, the control algorithm can be fully customized as a CRA. Users have the flexibility to implement their own models to handle conflicts between the PLC and SCADA commands. These custom algorithms can incorporate any desired logic, from simple decision rules to advanced machine learning models, depending on the complexity of the system and the research objectives.

The implemented algorithm can return 'OPEN' or 'CLOSE' to directly instruct the actuator to perform a specific action based on the custom logic. For pumps, WaCSim also supports speed control for variable speed pumps. By returning a numerical value between 0.0 and 2.0, the algorithm can set the pump speed from 0% (closed) up to 200%. This flexibility allows for the implementation of complex control logic that requires precise adjustments rather than binary on and off states.

```
1 - name: PLC1
2   actuators:
3     - name: P2
4       decision_maker: scada
5 - name: PLC4
6   actuators:
7     - name: P1
8       decision_maker: examples/EdenTown/PLC_Algo/PLC4_P1_Algo.py
9       dependents: [J1, T1]
```

Listing 4: PLCs Algorithms Configuration in *YAML* Format

2.4.2 | SCADA Algorithms

WaCSim supports the integration of advanced control strategies at the SCADA level, enabling centralized decision-making based on data collected from across the entire network. While PLCs are responsible for managing local components or specific dependencies, the SCADA system collects and aggregates sensor data from all distributed PLCs. This gives the SCADA a comprehensive, system-wide view of the WDS, allowing it to perform optimizations and make strategic decisions that individual

PLCs cannot achieve due to their localized scope and processing limitations. As a result, the SCADA system can coordinate multiple components more effectively, improving overall network performance, efficiency, and stability.

Implementing custom SCADA algorithms follows a similar process as configuring PLC algorithms. The user creates a *YAML* file that specifies which PLCs control active actuators, and a path to a custom algorithm with the appropriate Python file. An example of this configuration is provided in Listing 5.

The custom Python algorithm used by the SCADA system has access to all sensor data from across the network, along with the status of all actuators. This allows the SCADA to make more informed decisions by analyzing the complete state of the system in real time. The custom algorithm processes this data and is responsible for generating a command for the specific actuator managed by the PLC. The output of the SCADA algorithm is either a binary instruction ('OPEN' or 'CLOSE') or, in the case of variable speed pumps, a continuous control value between 0.0 and 2.0 that sets the pump speed.

In Hybrid Control mode, WaCSim extends this interaction by allowing the SCADA algorithm to transmit selected state variables to PLCs alongside the actuator command. This functionality is configured using the *hybrid_values_to_send* field in the SCADA algorithm configuration file. Through this option, users explicitly define which sensor readings or internal state variables computed at the SCADA level are forwarded to a specific PLC in addition to the control instruction. These values are represented using the original sensor name with an appended *S* suffix (e.g., T1S, J1S), allowing the PLC custom algorithm to distinguish between values received from peer PLCs (e.g., T1) and those transmitted by the SCADA system (e.g., T1S).

This selective data transfer enables controlled information sharing between centralized and local logic, allowing PLC-level algorithms and CRA to incorporate a curated subset of SCADA-derived information without exposing the full centralized system state. For example, a SCADA algorithm may issue a command to a pump while simultaneously transmitting selected tank levels or junction pressures that the PLC can use to validate, contextualize, or override the command under adversarial or uncertain conditions. When this field is not specified, the PLC receives only the actuator command.

It is important to note that the execution of these logic layers depends strictly on the selected simulation mode. In PLC Control mode, actuators are governed exclusively by local PLC algorithms, and control algorithms can therefore be integrated only at the PLC level. In SCADA Control mode, the centralized SCADA decision is the sole determinant of system behavior, with algorithm integration occurring at the SCADA level. In Hybrid Control mode, algorithms may be implemented at both the PLC and SCADA levels. However, the custom PLC algorithm acts as the final arbiter of the actuator state. In this configuration, the PLC processes the SCADA command and the transmitted hybrid values as inputs and performs the final decision-making logic before execution at the physical layer. As summarized in Table 1, this dual-level integration in Hybrid Control mode may introduce conflicts, which motivates the need for conflict resolution algorithms implemented at the PLC level.

```
- name: PLC4
  actuators:
    - name: P1
```

```
    decision_maker: examples/EdenTown/Scada_Case/ScadaAlgos/Scada_P1_Algo.py
    hybrid_values_to_send: [T1,J1]
```

Listing 5: SCADA Algorithms Configuration in *YAML* Format

Table 1 Control Modes and Custom Algorithms

	PLCs Algorithm	SCADA Algorithm	Conflict Resolution Algorithm
PLC Control	V	X	X
SCADA Control	X	V	X
Hybrid Control	V	V	V

2.5 | Attacks and Events

WaCSim uses the original attacks and events from DHALSIM and builds upon them. In our manual we provide detailed instructions on how to use them and describe their effects. In this section we will shortly explain the two attack types used to demonstrate the tool's capabilities in our results.

2.5.1 | Denial of Service

A Denial of Service (DoS) attack is a type of cyber assault that aims to disrupt the normal operation of a targeted computational unit, service, or network by overwhelming it with excessive or illegitimate traffic (Schuba et al. 1997). This saturation of resources can lead to significant slowdowns, complete unresponsiveness, or total service denial for legitimate users. In the context of water distribution systems, such an attack may target key control elements like PLCs or SCADA units, preventing them from receiving updated sensor readings or issuing control commands. As a result, these components are forced to operate based on outdated information, potentially causing actuators such as pumps or valves to remain in their last known state. This behavior can lead to unintended system outcomes, including the overfilling or depletion of storage tanks, pressure imbalances, and inefficient energy usage.

2.5.2 | Man In the Middle

A Man-in-the-Middle (MitM) Conti, Dragoni, and Lesyk (2016) attack involves an adversary intercepting and modifying network communication between devices, such as PLCs and SCADA systems, without the knowledge of either party. In WaCSim, MitM attacks are implemented using ARP poisoning to insert the attacker between communicating nodes, after which the attacker inspects and alters selected packet contents. In the standard MitM attack, users specify tags to be altered using a fixed value,

an offset, or a custom time-varying sequence provided via a CSV file (path). This allows precise control over how data is manipulated during the attack. For progressively changing attacks, the *seq_mitm* variant applies a scaling parameter (*scaleParam*) at regular time intervals (*scaleTime*), enabling simulation of attacks that gradually drift further from ground truth over time

2.6 | Output Data

Upon the completion of a WaCSim simulation, an output directory is generated, containing all the files associated with the simulation. This directory is organized into two main categories: **physical simulation results** and **network activity logs**. The simulation results files capture the physical behavior of the WDS as modeled by the EPANET engine. These files, formatted in CSV, include two key data sets: the actual physical data (ground truth) and the SCADA-reported data. The ground truth reflects the precise outputs from the hydraulic simulation, representing the undistorted conditions of the system. In contrast, the SCADA-reported data consists of the information polled by the SCADA system from the PLCs. In addition, each PLC also records its polled values from the network, allowing for better debugging and monitoring of the data received by the PLC. The network activity logs are stored as packet capture (.pcap) files, documenting all communication exchanges observed in the Mininet emulated network. These files provide a detailed record of the data packets transmitted between devices, offering insights into the network dynamics during both standard operations and under attack conditions.

These output files are essential for analyzing WDS behavior under both normal and attack scenarios. By comparing ground truth data with SCADA-reported values, users can detect discrepancies that may indicate cyber threats or malfunctions. The collected data can also be used to train AI models to recognize patterns and identify anomalies, enabling real-time monitoring and early threat detection. It is noteworthy that these exact monitoring algorithms can be directly embedded in the simulator using the custom algorithm capability at both the PLC and SCADA levels.

3 | RESULTS AND DISCUSSION

We evaluate WaCSim's functionalities across different control configurations using two case studies. The Eden-Town network is used to demonstrate the SCADA Control and PLC Control modes, providing a clear view of centralized and decentralized behavior in a compact system. The CTown network (Kapelan, Savic, & Mahmoud 2017) is used to explore the Hybrid Control mode, and introduce a scenario with multiple simulated attacks.

3.1 | Eden-Town Case Study

This subsection presents results from simulations using the Eden-Town network under SCADA Control and PLC Control modes. The compact scale of the network allows for direct comparison of communication structures and decision logic. We also demonstrate how custom algorithms can be integrated into each mode to influence decisions and enhance responsiveness during both normal operations and simulated attack scenarios.

Figure 6 provides a 3D diagram of Eden-Town, highlighting the operational and communication flows across WaCSim's three-layer architecture: the SCADA layer (top), cyber layer (middle), and physical layer (bottom). The diagram compares the structure and behavior of each control mode: PLC Control (left column), SCADA Control (center), and Hybrid Control (right).

In the SCADA layer, the SCADA may act as either a passive observer or an active controller, depending on the configuration. Under PLC Control, the SCADA only collects data; in SCADA Control and Hybrid Control, it also issues commands to the PLCs. These interactions are visualized by red dotted arrows (control commands) and black dashed lines for the transfer of sensing data to the central unit.

The cyber layer illustrates how PLCs communicate to execute control logic. Under PLC Control, PLCs exchange the specific sensor data required for their control operations directly via purple dotted arrows, which enables decentralized decision-making. SCADA Control routes all communication through the SCADA unit, while Hybrid Control combines both approaches by allowing PLC-to-PLC coordination as well as SCADA-driven oversight.

At the physical layer, components such as tanks, pumps, and valves are connected to PLCs via sensors and actuators. These connections enable real-time monitoring and actuation, depicted by double lines (component-to-component pipes) and solid lines (component-to-PLC wired connection).

3.1.1 | SCADA Control Mode

In this simulation, we demonstrate how the SCADA Control mode, combined with a custom algorithm, can be utilized to model a security mechanism capable of detecting and mitigating cyber attacks. Specifically, we focus on a DoS attack scenario and evaluate the SCADA Control mode's ability to maintain operational integrity under compromised conditions. Initially, we simulate the Eden-Town network in SCADA Control mode and maintain the same control logic as the baseline PLC rules shown in Listing 2. In this configuration, however, the point of execution shifts: the logic is performed by the SCADA system rather than the local PLCs. These rules, centrally enforced by the SCADA system, represent the baseline operation as illustrated in the middle diagram of Figure 6.

In this case, we model a malicious actor who successfully compromises the communication line between PLC 2, PLC 3, and the SCADA system. The attacker launches a DoS attack when the water level in T1 drops below 7. During this attack, the SCADA system is unable to receive updated readings from PLC 2 and PLC 3, leaving it with outdated information. As a result, during

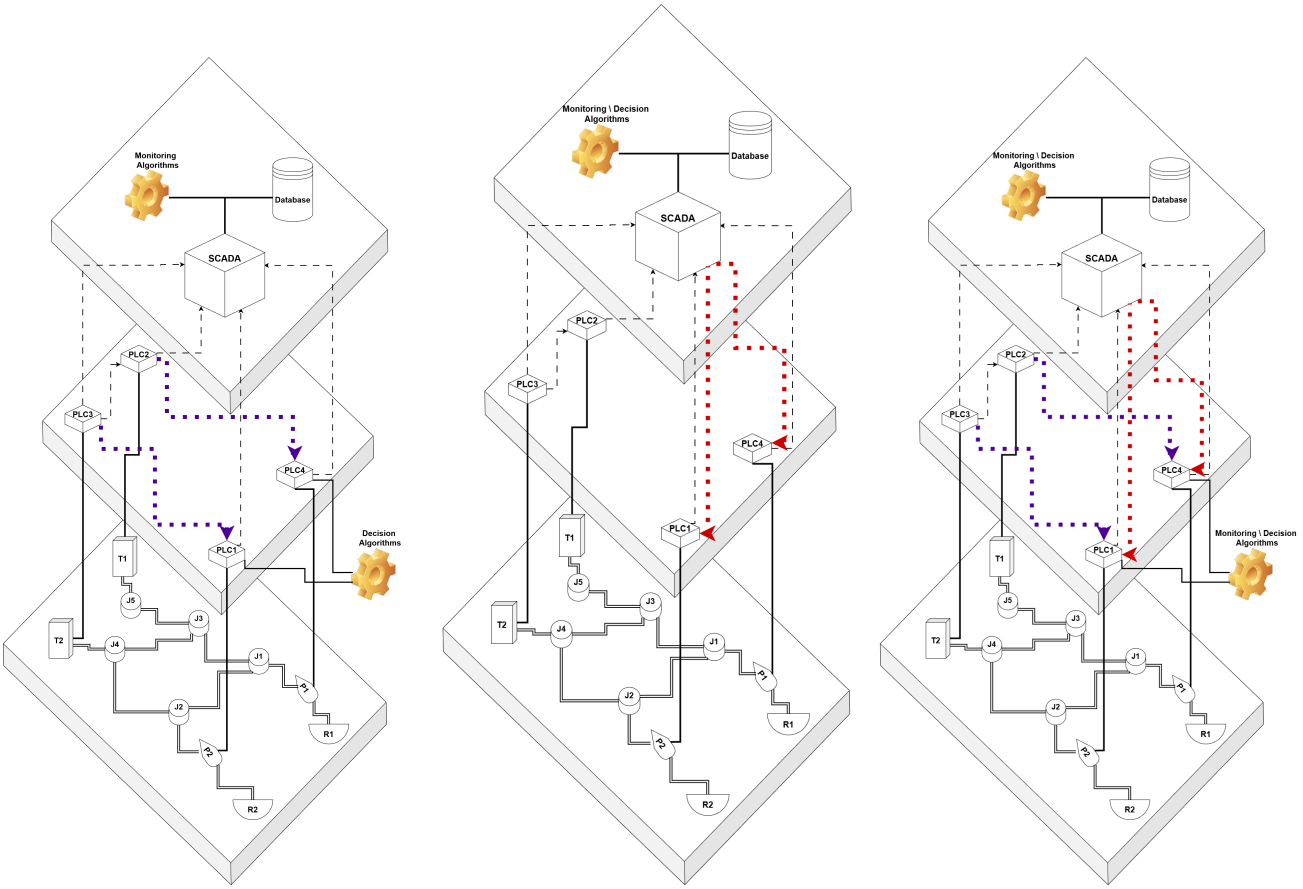

Figure 6 Multi-Layer Architecture of Eden-Town in WaCSim. The diagram illustrates the three operational modes: PLC Control, SCADA Control, and Hybrid Control. Double lines represent physical pipe connections, while solid black lines indicate wired component-to-PLC links. Black dashed lines show sensing data transfer to the SCADA, red dotted arrows represent centralized control commands, and purple dotted arrows indicate decentralized PLC-to-PLC data exchange for control operation

this period P1 continues operating while P2 remains idle. The SCADA system keeps instructing PLC 4 to run P1, following the control logic defined in Listing 2, lines 4–5, even though the actual tank level no longer justifies it. This causes T1 and T2 to keep filling beyond their target threshold, eventually reaching an overflow condition. Such behavior introduces inefficiencies and may lead to operational disruptions if not corrected promptly.

Figure 7 provides a detailed analysis of network traffic during the attack, categorized by ARP (blue), TCP DATA (green), and TCP SYN (red) packets. In the panels for PLC2 and PLC3, both of which are targeted, the start of the attack is clearly marked by a sudden surge in ARP packets used to redirect the communication path. This redirection causes an immediate collapse in TCP DATA throughput for these units as the sensor readings requests are blocked from reaching their destination.

The impact of the attacker blocking both units is mirrored in the SCADA panel, which shows a drastic fall in received TCP DATA during the attack window. Because the SCADA system is unable to receive the requested sensor data from the targeted PLCs, it subsequently stops sending out corresponding TCP DATA updates to the rest of the network. This explains why PLC1 and PLC4, though not directly targeted by the redirection, also exhibit a noticeable drop in TCP DATA traffic during this period.

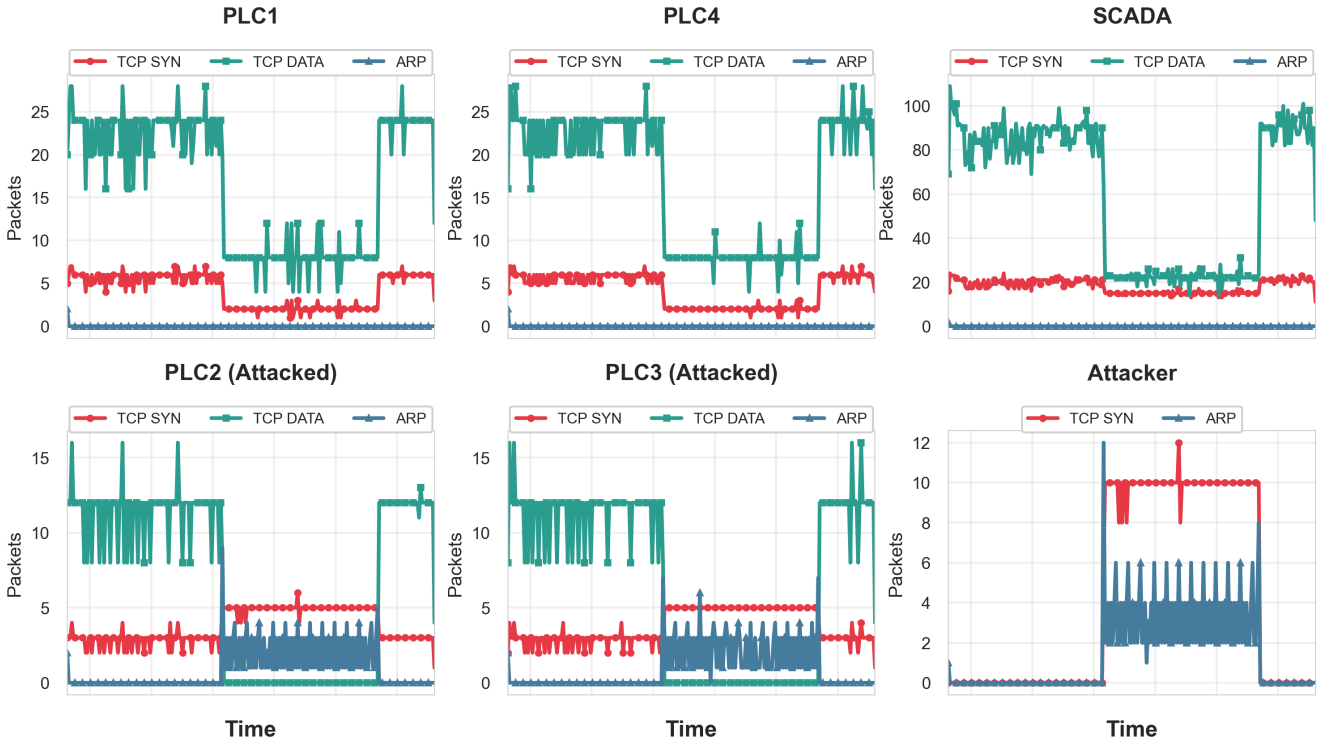

Figure 7 Cyber-layer packet activity during the DoS attack in SCADA Control mode. TCP DATA, TCP SYN, and ARP packet counts are shown for all PLCs, the SCADA, and the attacker. The attack targets the communication paths of PLC2 and PLC3, leading to a collapse of TCP DATA traffic and reduced data visibility at the SCADA.

Meanwhile, the Attacker panel reveals a high volume of TCP SYN and ARP packets initiated strictly during the attack. These observations clearly highlight two key indicators of the attack: the surge in ARP packets directed at PLC 2 and PLC 3, which signals a redirection attempt by the attacker, and the corresponding drop in TCP packets received by the SCADA, indicating disrupted communication with the compromised PLCs. While these indicators help identify the attack and pinpoint the affected components, they fail to provide the SCADA with the necessary data to determine the appropriate operational response for P1 and P2.

To further analyze the implications of the attack, we shift our focus to the physical layer of the Eden-Town network. The impact of the DoS attack on hydraulic behavior is most evident when comparing the actual physical state of the network, or ground truth, against the values reported to the SCADA system. Figure 8 illustrates these discrepancies by contrasting the real-time hydraulic behavior modeled in EPANET with the distorted data processed by the central controller.

In the left panel of Figure 8, the ground truth values represent the accurate sensor readings within the system. In contrast, the middle panel presents the SCADA-reported values, which reflect only the information successfully received and processed by the SCADA system. The stark differences between these two panels highlight the effectiveness of the DoS attack, which successfully isolated the SCADA system from the real-time data provided by the local PLCs. This lack of visibility forces the

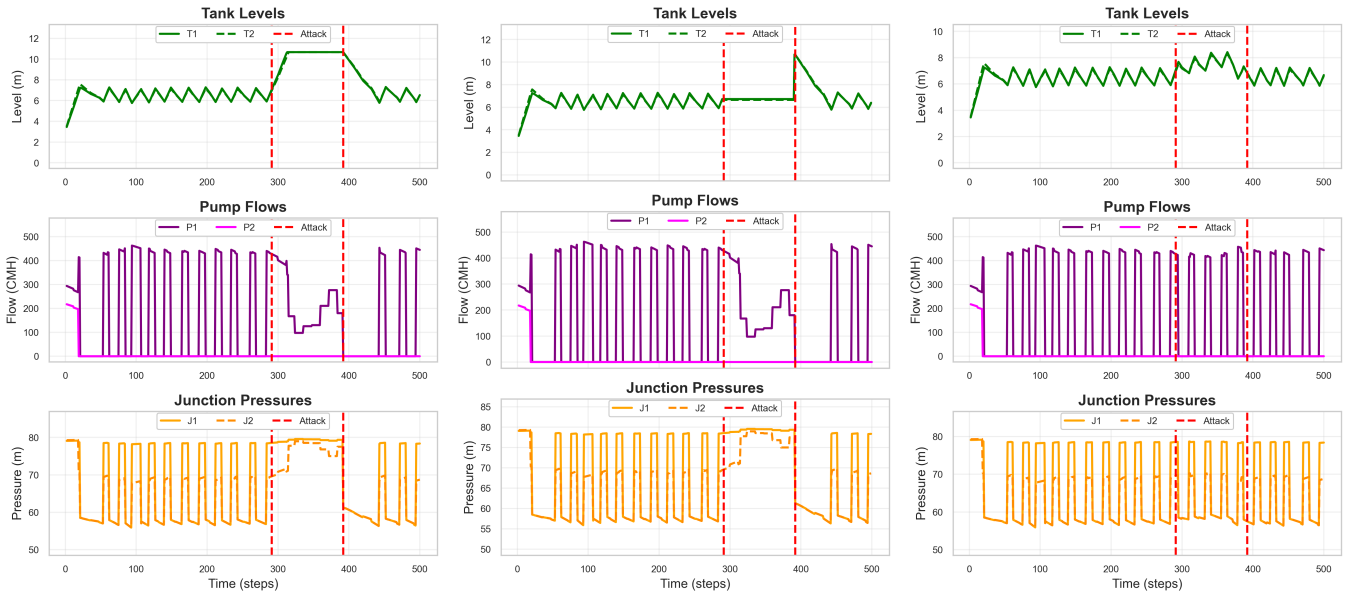

Figure 8 Physical Layer Ground truth (Left) Physical Layer SCADA Values (Middle), Physical Layer Ground truth with Custom Algorithm (Right)

system to operate based on frozen or outdated information, leading to the hydraulic inefficiencies discussed in the following sections.

The first row presents the water levels in Tank T1 and Tank T2. The ground truth data shows that both tanks continue to rise until they reach their maximum capacity (10.6m). At this stage, the EPANET engine automatically closes the inlet links for both tanks to prevent them from exceeding their defined limits. However, because of the DoS attack, the SCADA-reported values for both T1 and T2 remain frozen at the levels recorded before the communication was disrupted. As a result, the SCADA system continues to issue commands based on this outdated state: Pump P1 remains ON while Pump P2 remains OFF. The SCADA system never receives the updated data, which would normally trigger a change in pump operations. This results in P1 attempting to push water into a system with closed tank inlets, leading to constant high pressure observed in the third row of Figure 8.

The second row displays the flow rates of Pump P1 and Pump P2. In the left panel, the system initially operates as expected, with both pumps opening and closing according to the predefined rules. However, during the attack period, P1 remains open, and its flow gradually decreases to exactly meet the water demand in the network, keeping the water level in the tanks at maximum capacity.

These results highlight how the DoS attack obstructs the SCADA system from receiving critical updates, leading to poor operational decisions. However, enhancing the SCADA's data stream with additional sensors, such as flow readings and junction pressure measurements, can provide alternative insights into system behavior. This enriched observability enables the development of more robust defense algorithms capable of detecting and mitigating attacks by leveraging multiple data sources.

Custom Control Defense Algorithm

To mitigate the effects of a DoS attack, we designed and implemented a simple algorithm that can be integrated at the SCADA level to enhance the system's resilience against attacks that cause data loss or outdated sensor readings. We present this algorithm as an initial example, demonstrating how users can create and integrate a custom algorithm within the simulator framework. This serves as a foundation for developing more complex algorithms in future implementations.

At the SCADA level, the algorithm first creates a CSV file that acts as a simple database, storing all the required hydraulic data recorded by the SCADA throughout the simulation. Using this dataset, the algorithm tracks the median runtime of Pump P1 by counting how many consecutive time steps it remains active, as well as how long it remains inactive. It then calculates median ON and OFF durations based on all previous cycles from the clean historical data (before any attack), providing an estimate of how long P1 should remain active and inactive under normal conditions. Building on this, the algorithm monitors for data loss scenarios where the SCADA is unable to receive updated readings from T1. Specifically, if the sensor readings show negligible variation over a sliding window (indicating frozen or stale data), the algorithm activates a fail-safe routine. In this mode, P1 first completes its current cycle based on the previously computed median duration, then continues alternating between active and inactive states using these median values. This routine continues until the SCADA successfully receives updated readings showing normal sensor variation, at which point normal operation resumes based on the sensor-driven control thresholds.

Figure 8 right panel presents the results for the same simulation configuration under the same DoS attack but shows how an implementation of the custom algorithm described above at the SCADA level, can protect the system from the adverse effects of data loss. Unlike the baseline scenario where the SCADA remains blind to the rising water level in T1, the algorithm ensures that P1 follows an operational pattern that mimics normal behavior, preventing excessive water accumulation and reducing the risk of overflow.

By comparing the baseline ground truth in the left panel with the ground truth after applying the algorithm in the right panel, it becomes evident that the system maintains more stable operations despite the attack. The periodic activation and deactivation of P1, guided by the estimated runtime, prevents uncontrolled water flow and helps balance system pressure. Additionally, the algorithm's fail-safe approach ensures that SCADA-dependent operations do not remain indefinitely active in response to missing sensor data, mitigating the impact of cyber-induced disruptions.

While this implementation serves as a simple yet effective safeguard, this approach is however limited in scope, leaving opportunities for future enhancements that could significantly expand its robustness and adaptability. Refinements could incorporate additional sensor data, such as pressure readings and flow rates from junctions and valves, to improve inference accuracy. Moreover, integrating machine learning techniques could enable the algorithm to adapt dynamically to varying system conditions, further enhancing resilience against sophisticated cyber attacks.

This case study demonstrates that even a basic predictive algorithm can significantly enhance the robustness of smart water systems. Moreover, integrating such algorithms within the simulator not only improves resilience but also alters network behavior, enabling the modeling of operational scenarios that previous simulators could not capture. By allowing dynamic responses to cyber-induced disruptions, this approach provides new insights into system performance under attack, offering a more comprehensive framework for evaluating mitigation strategies.

3.1.2 | PLC Control Mode

While malicious attacks can disrupt system operations, as demonstrated in the previous example, real-world WDSs incorporate physical defense mechanisms within the hydraulic network to mitigate and prevent critical failures. These mechanisms aim to reduce the impact of events such as DoS attacks that could cause a tank overflow, potentially leading to further network damage due to excessive water pressure. A significant operational risk exists in these scenarios if a pump continues to remain active against a closed system. For instance, if safety valves close to prevent overflow while the pump remains operational, the resulting pressure surge can lead to pipe bursts or mechanical damage to the pump itself. However, current simulators lack the ability to model and integrate such real-world protective strategies. In this scenario, we demonstrate how a defense mechanism can be modeled and integrated using the custom algorithm interface. Specifically, we simulate the Eden-Town network with the addition of three valves-V1, V4, and V5-placed between the following junctions: J4 to T2, J5 to T1, and J1 to J3.

Figure 9 shows the modified network layout. For this simulation, the maximum level of both tanks T1 and T2 is set to 10.6 m and the control scheme is set to PLC control adhering to the rules listed in Listing 2.

Here we model a malicious actor with the same capabilities and objectives as in the previous example, compromising the communication line between PLC-2 and PLC-4 and initiating a DoS attack aimed at keeping Pump P1 running and overflowing the network's tanks. To achieve this, the attacker triggers the DoS attack when P1 is already running.

Custom Control Defense Algorithm

To develop a realistic defense mechanism against such attacks, we introduce a safeguard designed to prevent network tanks from exceeding a critical threshold. Specifically, we aim to cap tank levels at 10m, keeping them below the maximum capacity of 10.6m in Eden-Town. The core concept behind the control algorithms is to establish a physical (i.e., that does not depend on the cyber layer) safeguard mechanism that operates effectively under uncertain conditions, where network communication is compromised. By relying solely on local sensor data read directly by the PLCs, these algorithms enable autonomous decision-making, ensuring continued system stability even in the absence of network communication.

To achieve this, we integrate two controllable valves that regulate water flow to and from T1 and T2, controlled by PLC-2 and PLC-3, respectively. Additionally, we incorporate pressure sensors at Junction 5 and Junction 4 to improve network monitoring,

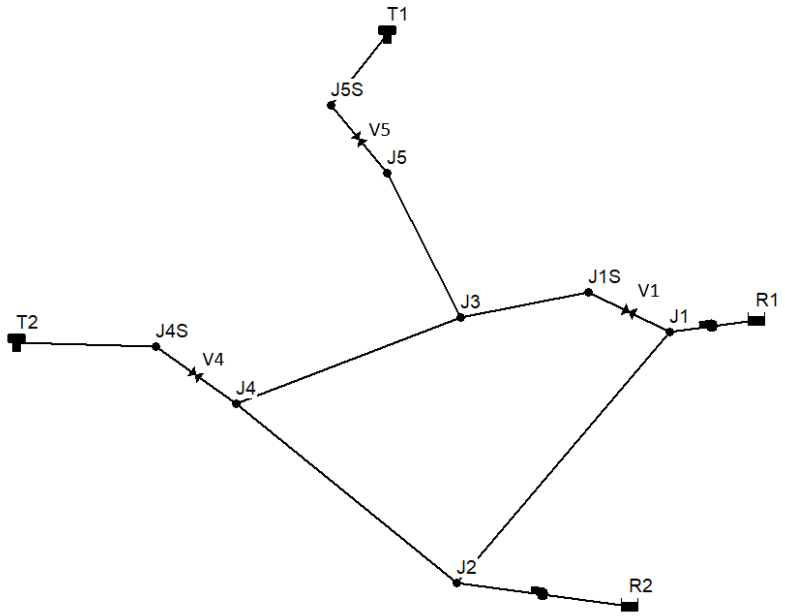

Figure 9 Eden-Town With added Valves

along with an additional pressure sensor at Junction 1, connected to PLC-4. All these components are physically linked to their respective PLCs, enabling real-time control and response and immunity to remote cyber-attacks. While the objective is clear, the main challenge lies in determining the optimal timing for closing and reopening the valves to prevent overflow while attempting to maintain the system stability.

To address this, we implement two custom algorithms at the PLC level. The first algorithm, deployed on PLC-2 (and similarly on PLC-3), continuously monitors the water level in the connected tank. If the tank reaches a level of 10, the algorithm closes the valve, preventing further water flow. Under normal conditions, based on the simple EPANET rules outlined in Listing 2, the tanks should not reach this level. However, an overworked pump-potentially due to a cyberattack-could continue pushing excess water into the system, disregarding the predefined rules.

Once the valve is closed to prevent overflow, the tank PLC must identify a safe window to reopen it without causing an immediate recurrence of the high-level condition. To achieve this, the PLC monitors the pressure at the upstream junction. While the valve is closed and the pump is active, the pressure at this junction remains elevated due to the resistance of the closed system. A calibrated decrease in this pressure serves as a hydraulic signal that the pumping activity has subsided or the network demand has changed. By detecting this drop, the tank PLC can infer that the inflow pressure has returned to a baseline level, allowing the valve to reopen and allow for tank draining.

In parallel, we implemented a custom algorithm in PLC-4 to control P1 while still adhering to the EPANET rules. This algorithm continuously monitors the flow of P1 and the pressure at Junction 1. When it detects a significant and sudden drop in

pump flow, it activates its protection mechanism, immediately shutting off the pump. In the following time steps, the algorithm continues monitoring the pressure at Junction 1. Once the pressure decreases below a predefined threshold, the protection mechanism is deactivated, and normal pump operation resumes according to the EPANET rules. However, if the attack remains active, the tanks will eventually fill again, causing the safeguard mechanism to restart the entire control cycle.

We ran the simulation with the implemented defense mechanism against the DoS attack. Figure 10 presents the ground truth results from the simulator, showing how the system responded under attack while operating with the safeguard in place.

Initially, as the DoS attack is launched, Pump P1 remains active and continuously pushes water into the system. As a result, Tank 1 and Tank 2 begin filling and almost reach their capacity, which is indicated by the black dash-dot line in the Tank Levels (top left) panel of Figure 10. However, once the levels reach 10 meters, the custom algorithms running on PLC-2 and PLC-3 take action by closing Valves V4 and V5 to prevent further inflow. This intervention successfully mitigates the overflow risk despite the attack's attempt to keep P1 running, as shown in the Valve Flows (bottom right) panel where the flow drops to zero around time step 200.

At this stage, with the valves closed, PLC-4 detects a sharp drop in the flow of P1, as illustrated in the Pump Flows (bottom left) panel of Figure 10. This drop occurs because closing Valves V4 and V5 restricts the available pathways for water distribution. Consequently, the pressure at Junctions J5, J4, and J1 increases, as seen in the Junction Pressures (top right) panel, leading to higher system resistance and a decline in P1's output. Recognizing this change, the custom algorithm on PLC-4 activates its protection mechanism and immediately shuts down P1.

With P1 shut down, the network pressure drops drastically, indicating the system is no longer experiencing excessive inflow. In response, the algorithms on PLC-2 and PLC-3 reopen the valves to restore normal hydraulic operations. As seen in the top left panel, the tanks begin to deplete even while the attack is still active. This depletion leads to a pressure decrease at J1, prompting the custom algorithm on PLC-4 to deactivate the defense mechanism and resume operation under the EPANET rules. However, because the attack persists and the reported status of P1 remains outdated, the pump activates again and continues pumping until it hits the same trigger point around time step 300. This cycle repeats until the attack ends at time step 350, after which the network returns to routine operation.

This example demonstrates how a realistic defense mechanism can be designed, integrated, and tested in WaCSim, effectively countering the DoS attack, preventing tank overflow, and attempting to maintain system stability. It also paves the way for developing more localized defense algorithms that leverage real-time sensor data while adapting to disruptions originating from the broader network.

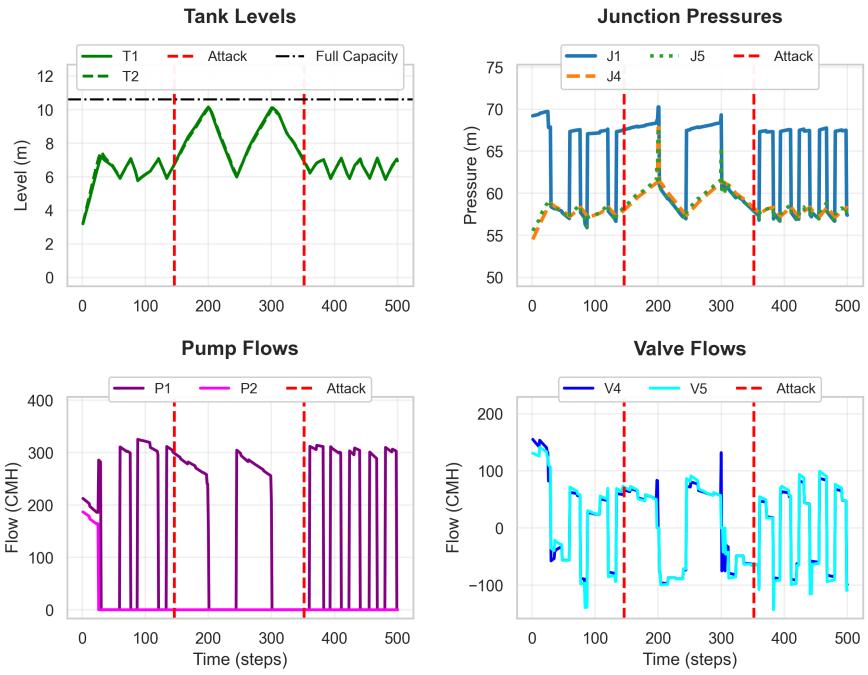

Figure 10 The Groundtruth results of the simulator.

3.2 | CTown Case Study - Hybrid Control Mode

In this section, we simulate a larger network, CTown (Kapelan et al. 2017), operating in Hybrid Control mode. We focus on a specific part of the network, examining the effects of three different cyber attacks designed to disrupt operations by manipulating the tank level values sent from a PLC to the SCADA, to another PLC, and to both simultaneously. Through this analysis, we evaluate the hydraulic signatures and data anomalies associated with such attacks.

To begin, we examine **PLC-5** within the CTown network. Figure 11 presents the topology of CTown on the right and the perimeter of PLC-5 on the left, and Table 2 presents the data received by PLC-5, categorized into three types:

- **Local Sensors**: Sensors physically connected to the PLC, providing direct real-time data. This includes the flow rate at the pumps, denoted by the suffix "F" (e.g., P10F represents the flow rate at Pump 10).
- **Controlled Actuators**: Actuators that PLC-5 can control, either via predefined rules or custom algorithms.
- **Received Readings**: Data transmitted over the network from other components (This includes tank levels (T5, T7) from other PLCs and values from the SCADA (T5S, T7S). Additionally, this category includes SCADA Command signals (e.g., SCADACmd_P10), which are the specific binary control commands issued by the central SCADA system to the PLC.

In CTown, at the PLC level, P10 is controlled by the predefined rules as defined in Listing 6.

```
[CONTROLS]
LINK PU10 OPEN IF NODE T7 BELOW 1.5
LINK PU10 CLOSED IF NODE T7 ABOVE 4.8
```

Listing 6: PLC-5 T7 Controls

Given the control logic defined above, the received readings for T7, T7S, and SCADACmd_P10 are of particular importance, as they directly dictate the operational state of P10. These specific variables are highlighted in Table 2 to illustrate their role in the hybrid architecture. In this mode, PLC-5 must process local tank level readings while simultaneously evaluating centralized directives from the SCADA system. This dual-source input creates a scenario where the SCADACmd might contradict the peer-to-peer data, necessitating the use of a CRA to resolve inconsistencies and maintain system stability.

All the data introduced above is recorded in PLC-5's output during the simulation and is also available for use in developing custom algorithms. This setup allows us to investigate how cyber attacks targeting specific communication links impact network behavior, system stability, and decision-making at both the local PLC level and the centralized SCADA level.

Table 2 PLC 5 Data: Critical variables for P10 are shaded.

Local Sensors	P8F	P10F	P11F	J302	J306	J307	J317
Controlled Actuators	P8	P10	P11	-	-	-	-
Received Readings	T5	T7	T5S	T7S	SCADACmd_P8	SCADACmd_P10	SCADACmd_P11

We introduce three attacks that utilize the Scaled Man-in-the-Middle method to disrupt network operations by altering data exchanged between PLCs and the SCADA. Each attack manipulates the reported tank level readings of **T7**, affecting how the system perceives the water levels and influencing control decisions accordingly.

1. **Attack 1: Direct Manipulation of PLC-5** The attacker directly targets PLC-5 by modifying the packets labeled T7 that it receives. The attack begins with an offset of 0.2, and with each simulation iteration, the offset increases by an additional 0.2. This manipulation gradually makes it appear as though the tank is continuously filling with more water.

2. **Attack 2: Compromising PLC-9** Instead of targeting PLC-5 directly, the attacker compromises PLC-9, which is physically connected to T7. The attacker differentiates between the values PLC-9 transmits to the SCADA and those sent to PLC-5:

    - The values sent to the SCADA (T7S) are offset by -0.2, with an additional -0.2 applied after every iteration, making it seem as though the tank is depleting.

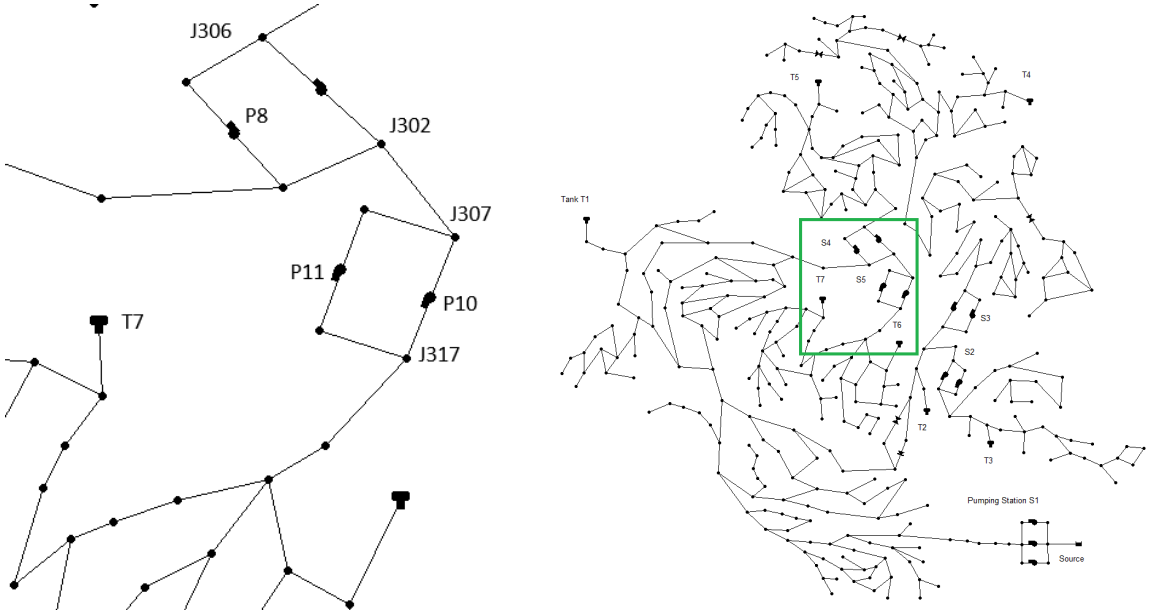

Figure 11 The CTown network layout: PLC 5 perimeter (left) and the full system network (right), with the PLC 5 region highlighted by a green square in the zoomed view.

- The values sent to PLC-5 (T7) are offset by +0.2, increasing by an additional +0.2 per iteration, creating the illusion that the tank is filling up.

3. **Attack 3: Manipulation of SCADA values of PLC-5** The attacker directly manipulates the values that PLC-5 receives from the SCADA. The T7S readings are offset by -0.2, with an additional -0.2 applied per iteration, making it appear as though the SCADA perceives the tank as draining faster than it actually is.

In summary, each of these three attacks targeted different communication links, creating distinct cyber signatures and resulting in varied hydraulic responses as perceived by PLC-5. By analyzing the data received during these attacks, we can observe how these manipulations impact system behavior.

For this simulation, we did not implement a complex CRA. Instead, PLC-5 resolved conflicts between the SCADA and PLC-reported values by giving the PLC decision a priority in case of a conflict. This approach assumes that the communication between PLC-5 and PLC-9 remains secure and trusted, highlighting the importance of safeguarding key data channels from potential cyber threats.

Data Analysis

Figure 12 presents the simulation output across three plots, each capturing a different aspect of the system's response under cyberattack conditions.

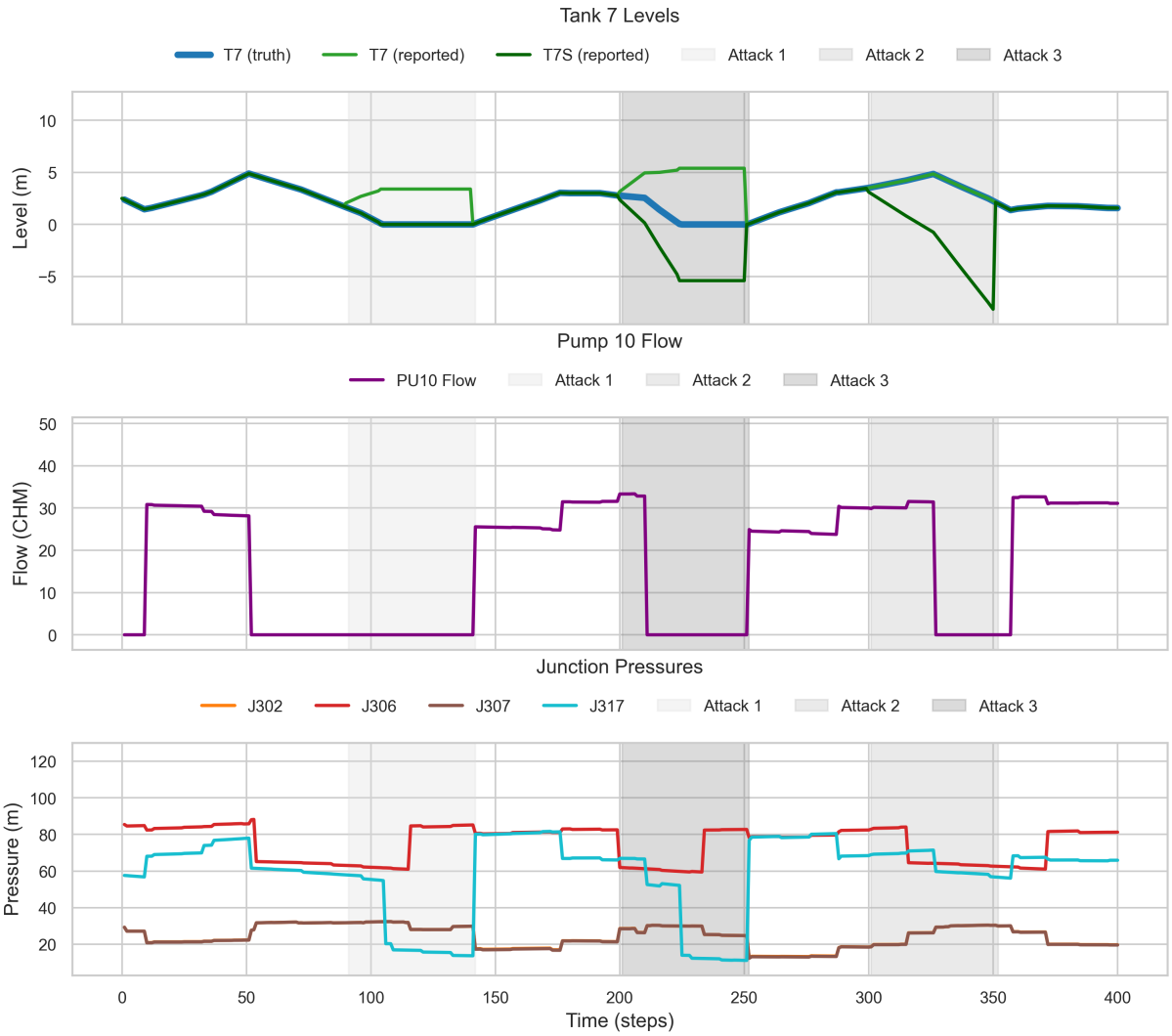

Figure 12 Simulation results under three malicious attacks. The suffix "reported" refers to data packets received by the PLC or SCADA after potential network manipulation.

Top Plot, Tank Level Readings, displays the tank level values, including T7 (received by PLC-5), T7S (the SCADA-reported tank level), and the SCADA Command for P10. Additionally, the ground truth T7 level is included to highlight discrepancies caused by the attacks. Middle Plot, Pump Flow, tracks P10 Flow, showing how the pump's operation changes in response to manipulated tank level readings. Bottom Plot, Junction Pressure Readings, monitors the pressure values at J302, J306, J307, and J317, revealing the hydraulic impact of each attack on the physical sensors connected to PLC-5. In addition, three vertical markers indicate the start and end of each attack.

Attack 1: Direct Manipulation of PLC-5 Before the attack begins, the top plot shows that T7 is naturally decreasing, indicating that the tank is draining while Pump 10 is closed, as seen in the middle plot.

As the attack starts, we observe a shift in T7's trend, where the value received by PLC-5 begins increasing, falsely suggesting that the tank is filling. Meanwhile, the ground truth T7 level and T7S (the SCADA-reported value) continue to decrease, unaffected by the attack.

In response, the SCADA issues a command to open P10 to counteract the perceived rising tank level. However, because the CRA in this simulation is configured to prioritize decisions based on local PLC rules and peer-to-peer data, it disregards the centralized SCADA command and keeps P10 shut down. This results in the actual tank continuing to deplete, leading to a pressure drop at J317 in the bottom plot.

This pressure drop contradicts the manipulated T7 value received by PLC-5, presenting an inconsistency in the data. Unlike our naive CRA implemented here, this scenario demonstrates that a well-designed CRA could incorporate junction pressure readings to detect communication inconsistencies between PLC-9 and PLC-5, potentially flagging and mitigating a cyberattack in real time.

Attack 2: Compromising PLC-9 In this attack, the attacker manipulates both values received by PLC-5, effectively controlling all the tank level information it reads from the network.

In the top plot, we observe that T7 (received by PLC-5) increases, while T7S (sent to SCADA) decreases, creating a clear divergence. Meanwhile, the ground truth data reveals that the tank is actually completely depleting, contradicting both manipulated values.

This extreme discrepancy highlights a critical vulnerability: a well-designed CRA must be prepared for scenarios where both communication lines, meaning those between PLCs and the SCADA system, are simultaneously compromised. In such instances, the CRA should shift to a fail-safe mode that relies on local sensor data or heuristic-based decision-making, such as time-based control, to maintain pump operation. Implementing these autonomous strategies ensures that the system can still prevent critical failures, like tank overflow or total depletion, even when it is isolated from the rest of the network.

Attack 3: Manipulation of SCADA values to PLC-5 In this attack, the attacker manipulates T7S, decreasing its value over time. However, T7 (received from PLC-9) and SCADACommand P10 remain unaltered, providing PLC-5 with an additional reference point.

As long as T7 remains unaltered, the attack does not impact the pump operation or the overall simulation behavior. However, this scenario highlights an opportunity for implementing anomaly detection within the control logic. Since WaCSim allows for a custom CRA, users can embed specific detection methods that flag the T7S communication line as compromised when its trend deviates significantly from the unaltered T7 values. This reinforces the importance of monitoring data consistency across multiple communication channels to detect, localize and mitigate cyber threats.

4 | CONCLUSIONS AND FUTURE WORK

The WaCSim framework, presented here, establishes a new foundation for cyber-physical research in smart water systems by unifying realistic hydraulic modeling, detailed cyber-network emulation, and fully customizable control logic within a single simulation environment. By enabling researchers and practitioners to design, test, and refine resilient operational strategies under dynamic and adversarial conditions, WaCSim paves the way for more secure, intelligent, and adaptive next-generation water distribution networks.

The results presented in this paper demonstrate the effectiveness of WaCSim for evaluating cyber-physical interactions in smart water systems. By integrating different control architectures, including PLC Control, SCADA Control, and Hybrid Control, and allowing for the embedding of customizable Python-based algorithms within the simulation, we provide capabilities that extend beyond those of state-of-the-art simulators. This framework enabled us to assess the resilience of water distribution networks against cyber attacks and explore sophisticated mitigation strategies that were previously difficult to model.

One of the key contributions of WaCSim is its ability to enable the design and testing of network configurations that resemble real-world water distribution systems. With WaCSim, users can model a network that mirrors the actual infrastructure they plan to implement with greater flexibility, including the placement of PLCs, sensors, and actuators.

By leveraging WaCSim's flexible architecture, users can evaluate different control modes to determine the optimal operational strategy for their specific network. Additionally, users can experiment with securing communication lines by identifying which network paths must be trusted and which could be vulnerable to cyber threats. This allows for the strategic placement of additional PLCs, pressure sensors, and flow meters, creating a more robust data platform that enhances operational efficiency and security.

Beyond cyber security considerations, the introduction of custom algorithms within WaCSim allows users to both model existing physical guardrails and design new ones. These guardrails, such as emergency shutoff valves or pressure relief systems, play a crucial role in maintaining system stability under unexpected conditions. By integrating these into the simulation, users can simulate the specific response and behavior of the decision maker when anomalies are detected. Unlike existing simulators that are passive and rely on predetermined conditions, WaCSim provides the unique capability to mimic real-time responses and evaluate their impact on the system as an event unfolds. This allows users to test the effectiveness of protective mechanisms, fine-tune operational parameters, and develop adaptive strategies that significantly improve network resilience.

As demonstrated in the PLC Control results, custom algorithms can autonomously regulate actuators like valves and pumps based on local sensor data, even when network communication is compromised. This functionality ensures that the network can maintain stability independent of centralized control, closely resembling real-world hydraulic safeguards that prevent overflows, pressure surges, and other critical failures.

Moreover, by considering real-world budget constraints, engineers can balance the trade-off between security and cost-effectiveness, testing various configurations before committing to infrastructure investments. This approach ensures that the final deployed network is not only functional and resilient but also optimized to balance cost and security.

The architecture of WaCSim supports a development cycle that moves beyond the typical "one-off" simulation. Because the tool allows for real-time intervention by the control logic, it creates an environment where a researcher can treat the simulator as a testing laboratory for evolving code. This process follows five stages:

1. **Attack Profiling**: The user begins by executing specific threat scenarios to characterize how a cyber-disruption, such as a Man-in-the-Middle attack, manifests across the hydraulic system. This stage defines the baseline impact on pressure, flow, and tank levels.

2. **Logic Design**: Using the data from the attack profile, the user develops a custom Python-based response. This logic can be structured as a CRA, an anomaly detection model, or an adaptive control strategy.

3. **In-Loop Integration**: The new logic is embedded directly into the simulation loop. Unlike passive tools, WaCSim allows the algorithm to interact with the system in real time, influencing pump behavior and valve states as the attack progresses.

4. **Robustness Testing**: The proposed logic is subjected to varying network configurations and evolving attack signatures to identify edge cases where the defensive strategy might fail.

5. **Operational Refinement**: The developer iterates on the algorithm until the system maintains stability under interference. This results in a control model that has been verified against dynamic failures before any real-world implementation is considered.

By closing the loop between the attack, the response logic, and the resulting hydraulic behavior, WaCSim shifts the focus from simple observation to active engineering. This framework enables the development of proactive, self-healing systems in a way that traditional simulators cannot.

In summary, WaCSim empowers Users with a new methodology for network design and cyber security development. By allowing them to build, test, and refine both physical and cyber-network configurations in a realistic setting, it serves as a critical tool for advancing the security and efficiency of next-generation water distribution systems.

Our future work for this simulator is divided into two main paths: (i) Enhancing the Simulator and Expanding Data Accessibility; and (ii) Leveraging Simulation Data for AI-Powered SCADA Systems. For the former, we aim to enrich the simulator's data platform by enabling the SCADA and PLCs to request additional data on demand from both the hydraulic and cyber layers. This would allow the transmission of critical information over the network, facilitating the development of more complex algorithms and enabling two-way communication between different network components. For example, a SCADA or PLC could

request a specific PLC to provide a summary of recorded cyber-layer packets or retrieve additional sensor values not typically used in routine operations. Additionally, we plan to introduce more sophisticated cyber-attack models with varied cyber-layer signatures. This includes integrating a custom attacker algorithm interface, allowing users to program advanced attack strategies in Python. This feature would enable Users to design and test adaptive, AI-driven cyber threats, simulating more realistic attack scenarios within the simulation environment.

For developing AI-Powered SCADA Systems, the simulator generates a rich dataset that can be leveraged to develop a generative AI model operating at the SCADA level. This AI-driven system would oversee network readings, detect anomalous behavior, and provide an interactive interface for human operators using natural language. Our next step is to integrate this AI model into the simulator's SCADA system, evaluating its performance in detecting cyber threats, predicting system behavior, and facilitating real-time human interaction. This will allow us to test its predictive capabilities, response mechanisms, and decision-making support under various attack and operational scenarios.

Ultimately, WaCSim lays the groundwork for a new class of cyber-physical simulation environments, ones that are not only capable of representing real-world infrastructure, but also serve as extensible platforms for AI model development, secure-by-design testing, and adaptive control research.

References

Antonioli, D., & Tippenhauer, N. O. (2015). *Minicps: A toolkit for security research on cps networks.* Retrieved from `https://arxiv.org/abs/1507.04860`

Bao, Y., & Mays, L. W. (1990). Model for water distribution system reliability. *Journal of Hydraulic Engineering*, *116*(9), 1119-1137. Retrieved from `https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-9429%281990%29116%3A9%281119%29` doi: 10.1061/(ASCE)0733-9429(1990)116:9(1119)

Boschert, S., & Rosen, R. (2016). Mechatronic futures: Challenges and solutions for mechatronic systems and their designers,. In D. B. Peter Hehenberger (Ed.), (p. 59-74). Cham: Springer International Publishing.

Conti, M., Dragoni, N., & Lesyk, V. (2016). A survey of man in the middle attacks. *IEEE communications surveys & tutorials*, *18*(3), 2027–2051.

Franco, D. J., Muhammed, A. B., Subramaniam, S. K., Abdullah, A., Silva, R. M., & Akram, O. K. (2019). A review on current and old scada networks applied to water distribution systems. In *2019 first international conference of intelligent computing and engineering (icoice)* (pp. 1–11).

Fu, G., Jin, Y., Sun, S., Yuan, Z., & Butler, D. (2022). The role of deep learning in urban water management: A critical review. *Water Research*, *223*, 118973. Retrieved from `https://www.sciencedirect.com/science/article/pii/`

S0043135422005018 doi: 10.1016/j.watres.2022.118973

Fu, G., Savic, D., & Butler, D. (2024). Making waves: Towards data-centric water engineering. *Water Research*, *256*, 121585. Retrieved from https://www.sciencedirect.com/science/article/pii/S0043135424002298 doi: 10.1016/j.watres.2024.121585

Housh, M., & Ohar, Z. (2018). Model-based approach for cyber-physical attack detection in water distribution systems. *Water Research*, *139*, 132-143. Retrieved from https://www.sciencedirect.com/science/article/pii/S0043135418302355 doi: https://doi.org/10.1016/j.watres.2018.03.039

Jenny, H., Alonso, E. G., Wang, Y., & Minguez, R. (2020). Using artificial intelligence for smart water management systems.

Kapelan, Z., Savic, D., & Mahmoud, H. (2017, 12). A response methodology for reducing impacts of failure events in water distribution networks. *Procedia Engineering*, *186*, 218-227. doi: 10.1016/j.proeng.2017.03.231

Klise, K. A., Murray, R., & Haxton, T. (2018). An overview of the water network tool for resilience (wntr).

Li, J., Yang, X., & Sitzenfrei, R. (2020). Rethinking the framework of smart water system: A review. *Water*, *12*(2). Retrieved from https://www.mdpi.com/2073-4441/12/2/412 doi: 10.3390/w12020412

Mahmoud, H., Wu, W., & Gaber, M. M. (2022). A time-series self-supervised learning approach to detection of cyber-physical attacks in water distribution systems. *Energies*, *15*(3). Retrieved from https://www.mdpi.com/1996-1073/15/3/914 doi: 10.3390/en15030914

Mala-Jetmarova, H., Sultanova, N., & Savic, D. (2017-07). Lost in optimisation of water distribution systems? a literature review of system operation. *Environmental modelling & software : with environment data news.*, *93*.

Montalvo, I., Izquierdo, J., Pérez-García, R., & Herrera, M. (2014). Water distribution system computer-aided design by agent swarm optimization. *Computer-Aided Civil and Infrastructure Engineering*, *29*(6), 433-448. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12062 doi: https://doi.org/10.1111/mice.12062

Murillo, A., Taormina, R., Tippenhauer, N., & Galelli, S. (2020). Co-simulating physical processes and network data for high-fidelity cyber-security experiments. In *Sixth annual industrial control system security (icss) workshop* (pp. 13–20).

Nikolopoulos, D., Moraitis, G., Bouziotas, D., Lykou, A., Karavokiros, G., & Makropoulos, C. (2020). Cyber-physical stress-testing platform for water distribution networks. *Journal of Environmental Engineering*, *146*(7), 04020061.

Perelman, G., & Fishbain, B. (2022). Critical elements analysis of water supply systems to improve energy efficiency in failure scenarios. *Water Resources Management*, *36*(10), 3797–3811.

Perelman, G., Ostfeld, A., & Fishbain, B. (2023). Robust optimal operation of water distribution systems. *Water*, *15*(5), 963.

Ramos, H. M., Kuriqi, A., Besharat, M., Creaco, E., Tasca, E., Coronado-Hernández, O. E., … Iglesias-Rey, P. (2023). Smart water grids and digital twin for the management of system efficiency in water distribution networks. *Water*, *15*(6). Retrieved from https://www.mdpi.com/2073-4441/15/6/1129 doi: 10.3390/w15061129

Raza, N., & Moazeni, F. (2025). Optimal cybersecurity framework for smart water system: Detection, localization and severity assessment. *Water Research*, *281*, 123517. Retrieved from https://www.sciencedirect.com/science/article/pii/S0043135425004300 doi: 10.1016/j.watres.2025.123517

Rossman, L. A., et al. (2000). Epanet 2: users manual.

Saravanan, K., Anusuya, E., Kumar, R., & Son, L. H. (2018). Real-time water quality monitoring using internet of things in scada. *Environmental monitoring and assessment*, *190*(9), 556.

Schuba, C. L., Krsul, I. V., Kuhn, M. G., Spafford, E. H., Sundaram, A., & Zamboni, D. (1997). Analysis of a denial of service attack on tcp. In *Proceedings. 1997 ieee symposium on security and privacy (cat. no. 97cb36097)* (pp. 208–223).

Seung Won Lee, D. J. J., Sarper Sarp, & Kim, J. H. (2015). Smart water grid: the future water management platform. *Desalination and Water Treatment*, *55*(2), 339–346. Retrieved from https://doi.org/10.1080/19443994.2014.917887 doi: 10.1080/19443994.2014.917887

Taormina, R., Galelli, S., Douglas, H. C., Tippenhauer, N. O., Salomons, E., & Ostfeld, A. (2019). A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems. *Environmental modelling & software*, *112*, 46–51.

Taormina, R., Galelli, S., Tippenhauer, N. O., Salomons, E., & Ostfeld, A. (2017). Characterizing cyber-physical attacks on water distribution systems. *Journal of Water Resources Planning and Management*, *143*(5), 04017009.

Taormina, R., Galelli, S., Tippenhauer, N. O., Salomons, E., Ostfeld, A., Eliades, D. G., … Ohar, Z. (2018). Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. *Journal of Water Resources Planning and Management*, *144*(8), 04018048. Retrieved from https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29WR.1943-5452.0000969 doi: 10.1061/(ASCE)WR.1943-5452.0000969

Tsiami, L., & Makropoulos, C. (2021). Cyber—physical attack detection in water distribution systems with temporal graph convolutional neural networks. *Water*, *13*(9). Retrieved from https://www.mdpi.com/2073-4441/13/9/1247 doi: 10.3390/w13091247

Vairavamoorthy, K., & Ali, M. (2000). Optimal design of water distribution systems using genetic algorithms. *Computer-Aided Civil and Infrastructure Engineering*, *15*(5), 374-382. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/0885-9507.00201 doi: https://doi.org/10.1111/0885-9507.00201

Zekri, S., Jabeur, N., & Gharrad, H. (2022). Smart water management using intelligent digital twins. *Computing and Informatics*, *41*(1), 135–153.

How to cite this article: Williams K., B. Hoskins, R. Lee, G. Masato, and T. Woollings (2016), A regime analysis of Atlantic winter jet variability applied to evaluate HadGEM3-GC2, *Q.J.R. Meteorol. Soc.*, *2017;00:1–6*.