A Formal Model for Reliable Data Acquisition and Control in Legacy Critical Infrastructures

Jose Miguel Blanco *, Jose M. Del Alamo, Juan C. Dueñas, Felix Cuadrado

Posted Date: 26 March 2024

doi: 10.20944/preprints202403.1611.v1

Keywords: Critical Infrastructure; Water Distribution Network; Formal Model; Digital Transformation; Data Management; Data Security; Data Acquisition

 doi:10.20944/preprints202403.1611.v1

Article

A Formal Model for Reliable Data Acquisition and Control in Legacy Critical Infrastructures

José Miguel Blanco *, Jose M. Del Alamo, Juan C. Dueñas and Felix Cuadrado

ETSI Telecomunicación, Universidad Politécnica de Madrid;
jm.delalamo@upm.es (J.M.D.A.); juancarlos.duenas@upm.es (J.C.D.); felix.cuadrado@upm.es (F.C.)
* Correspondence: josemiguel.blanco@upm.es

Abstract: The digital transformation of critical infrastructures, such as energy or water distribution systems, is essential for their smart management. Faster issue identification and smoother services enable better adaptation to consumers' evolving demands. However, these large-scale infrastructures are often outdated. Their digital transformation is crucial to enable them supporting societies. This process must be carefully planned, providing guidance that ensures that the data they rely on is dependable and that the system remains fully operational during the transition. This paper presents a formal model that supports reliable data acquisition in legacy critical infrastructures, facilitating their evolution towards a data-driven smart system. Our model provides the foundation for a flexible transformation process while generating dependable data for system management. We demonstrate the model's applicability in a use case within the water distribution domain and discuss its benefits.

Keywords: critical infrastructure; water distribution network; formal model; digital transformation; data management; data security; data acquisition

1. Introduction

The development of fully functional Smart Grids in recent years [1] has led to the digital transformation of formerly analog critical infrastructures. This transformation has become an important topic of research [2]. To achieve this transformation, acquiring reliable data is essential as it provides the raw material necessary for data-driven decision-making processes. By collecting and analyzing data from various sources, organizations can gain insights into customer behavior and infrastructure operation. This, in turn, enables them to make informed decisions that drive proper management and operation of these critical systems.

One domain that seems to be gaining research attention lately is Water Distribution Networks (WDNs) [3]. WDNs pose some similarities with the electrical network, e.g., their distribution over a large territory. Yet, there are also significant differences aside from the physical nature and properties of the elements provided by the services. For example, WDNs are primarily underground, leading to increased difficulty in acquiring and validating detailed data from specific elements. As a result, while there are lessons to be learned from the evolution of Smart Grids, a distinct approach is required to tackle the unique challenges of WDNs.

In particular, the digital transformation of WDNs from legacy analog systems into data-driven ones requires some guidance that offers flexibility and ensures reliability. Flexibility is required to support an incremental approach during the transformation process so as to introduce more data acquisition nodes as needed. Reliability is a must to ensure the proper verification of dependable data, thus providing a solid framework for detecting the relatively higher number of failures compared to other critical infrastructures [4], even with daily maintenance issues [5]. Furthermore, having dependable data paves the way for implementing a new Supervisory Control And Data Acquisition (SCADA) system [6] and a set of decision support tools to transform the current legacy system into a data-driven digital one.

Thus, the main goal of this paper is to create a formal model that would ensure a flexible and correct deployment of a SCADA system over a legacy WDN. Furthermore, the formal model will also ensure the validity of the data generated and inferred by the SCADA system and its nodes. In this

 doi:10.20944/preprints202403.1611.v1

sense, the formal model supports and guides the deployment of a SCADA system and its decision support tools when gearing towards a transition to a data-driven digital system in WDNs. Going into detail, the model validates the data through node and system-level statements, thus providing the reliability required for issue detection and decision-making in critical infrastructures. In addition, the model establishes the required topology for deploying the nodes of the SCADA system to ensure its correct functioning. To this end, the model is based on ternary relational semantics, also known as Routley-Meyer semantics [7]. Initially introduced to address specific technical challenges of relevance logics, these logics are lately being applied to distributed systems such as smart grids [8]. Far from a purely theoretical exercise, the model has been validated by creating a software artifact to demonstrate its applicability in an industrially relevant environment.

The paper outline is as follows. Section 2 provides some background on WDNs and their data acquisition system. Section 3 analyses the related works. Section 4 describes the formal model, and Section 5 details its implementation into a software artifact. Section 6 demonstrates the model's applicability and discusses its benefits. Finally, Section 7 concludes the paper.

2. Data acquisition in Water Distribution Networks

Although WDNs can present different topologies [9], Figure 1 introduces the main elements to be found in any WDN. Water can be extracted from various sources, such as rivers, lakes, reservoirs, underground wells, or even from the ocean through desalination processes. The water source used depends on several factors, such as geographical location, availability of water resources, and existing supply infrastructure. Several nodes are linked to the water intake, such as pumps and drivers, where the water is transferred to specific points, such as treatment plants, where it undergoes further processing. The treatment plants carry out processes to remove solid particles; filtration to eliminate fine particles, organic matter, microorganisms, and dissolved chemical compounds; disinfection to eliminate bacteria, viruses, and other microorganisms; and pH adjustment and demineralization. Once treated, the drinking water is stored in tanks and then distributed through the pipeline network for domestic, industrial, and commercial use.

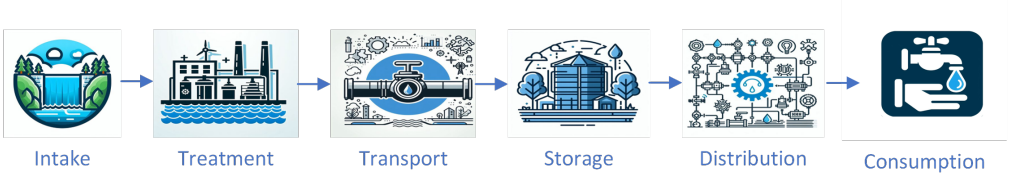


Figure 1. Stages of water processing lifecycle in a Water Distribution Network: Intake, Treatment, Transport, Storage, Distribution, and Consumption.

SCADA systems for WDN monitoring and control typically consist of, among other components, data acquisition units, automation and control units in the field, oftentimes with capabilities to execute complex logic operations such as Programmable Logic Controllers (PLCs) supervisory controllers, a supervisory system responsible for gathering data and sending control commands to the field devices, the communication elements between them, and the user interface that allows visualization of collected data and performing control operations. The SCADA components are distributed at different points of the WDN, forming a sensing (data acquisition) and control (actuators) overlay. The SCADA system includes multiple nodes distributed throughout the water intake, treatment, transport, storage, and distribution elements, such as water quality and temperature sensors, pressure and flow meters, etc. At the end of the WDN, we find smart water meters comparable to those in the electricity network [10].

 doi:10.20944/preprints202403.1611.v1

3. State of the Art

This paper addresses the digital transformation of legacy critical infrastructures, particularly WDNs, by proposing a formal model to guide the deployment of SCADA nodes and ensure the validity of the data they acquire. In this section, we analyze the previous works that have dealt with this topic.

The state of the art includes some previous works dealing with the WDNs' digital transformation [11–19]. Some initial works [11] focused on leveraging historical data to identify critical areas in the distribution network for pipe burst location and prioritizing pipes for rehabilitation. With the advent of digital twins, different authors [12–15] have proposed frameworks for leveraging real-time data to assist technicians and water utilities in addressing other issues. For example, in Zekri et al. [14] a digital twin supports detecting pipe bursts or unauthorized water usage. Recent works [16] have coined the term Water 4.0, in explicit parallelism with the Industry 4.0 concept [20], to highlight the smartification, data-driven orientation of modern WDNs. Under the Water 4.0 paradigm, various technologies such as cyber-physical systems, internet of things, big data analytics, artificial intelligence as well as cloud computing are combined to solve different management and operation problems in WDNs such as water routing [17], optimal pump control [18] and leakage detection [19]. While all these works exemplify the value of WDNs' digital transformation to support the smart operation and maintenance of WDNs, they focus on delivering support services once the data gathered from the infrastructure is available. In contrast, our work focuses on guiding the deployment of the sensors that acquire the data and validating the statements they produce.

SCADA systems [6] are widely used in industrial processes to acquire monitoring data, and previous research has already reported their application in critical infrastructures [21] such as oil refineries [22] and WDNs [23]. The main approaches to ensure the monitoring data's validity include applying statistical methods, machine learning, and data fusion techniques. Statistical methods such as outlier detection, regression analysis, and statistical process control help identify data points that deviate significantly from expected patterns, indicating potential sensor faults or environmental influences affecting sensor readings. For example, Rigatos et al. [24] proposed using Kalman filters to emulate the functioning of fault-free sensors, comparing them against the output of actual sensors and using the resulting differences for statistical decision-making. Machine learning models can learn from historical data to predict sensor values and identify anomalies by applying supervised learning techniques for known error patterns [25] and unsupervised learning techniques for detecting new anomalies [26]. And while certain techniques dealing with images have been applied in different domains [27], machine learning approaches in WDN are still costly due to the limited amount of available images of underground legacy systems. Data fusion techniques integrate data from multiple sensors measuring the same parameter to improve accuracy and reliability [28]. Redundancy, where multiple sensors perform the same measurement, allows for cross-validation of data, helping to identify and isolate faulty sensors.

Our contributions share the goal of ensuring data validity. Yet, different from other approaches, the application of formal models brings specific advantages, including rigorous guarantees (i.e., formal verification ensures that the system adheres to its specifications under all conditions, something that statistical methods and ML models, which rely on probabilistic outcomes, cannot guarantee), predictability (i.e., formal methods provide a clear and predictable framework for analyzing system behaviors, making it easier to predict and understand outcomes), exhaustiveness (i.e., formal methods can systematically explore all possible states of a system, including edge cases that might not be covered by historical data, thus helping in identifying potential issues that could be missed by other approaches, which are inherently limited by the scope and quality of the data they are based on), and early error detection. Indeed, formal methods can be applied early in the design phase, enabling the detection and correction of errors before a system is built or deployed, which is the case for legacy analog critical infrastructures undergoing a digital transformation process. This proactive approach can save considerable resources and time compared to other methods, which typically require implementation or data collection before validation can occur.

 doi:10.20944/preprints202403.1611.v1

In this work, we are focusing on providing a design model that would ensure the expected results if the implementation follows the design [29]. This contrasts with the most recent approaches to formal verification that rely on model checking, but that approach is usually costly and cannot be extended easily. Thus they usually focus on analysing the faulty behaviour of the infrastructure (ex-post) instead of its design (ex-ante), since the infrastructure is not prone to change [30,31]. Similarly, formal models have been used to detect attacks in SCADA systems [32], or to model the attacks themselves [33]. Nonetheless, formal methods have been also used to model the infrastructure as in case of Smart Grids. Still, those efforts have focused on understanding the system behavior at runtime [34]. To the best of our knowledge, the only real example of enabling the deployment of a SCADA system while making use of formal models to ensure resiliency can be found in [35]. Still, the extension of the work done is limited to Smart Grids and does not consider the flexibility required in deploying new nodes.

Our contributions build on a previous work [36], where we described an early formal model for the water domain and carried out a theoretical validation. This paper advances this related work by describing a detailed implementation and further demonstrating its maturity through its application in a real-world scenario from C-Town [37], one of the industry standards often used as a benchmark in the domain, thus reaching a technology readiness level of 5[1]: technology validated in a industrially relevant environment.

4. Formal Model

Whenever a SCADA system comes into play, we have to specify if we are considering it from a topological point of view or if, otherwise, we are approaching it from the point of view of its processing of data. In this case, we are tackling the latter, so we will begin by understanding what type of data it produces and processes. At first, it is easy to see that the data primarily generated by the SCADA will come from sensors, and because of that, the data will have the shape of simple statements. For example, it could be the case that a pressure sensor would be reporting "pressure: 2.8", and an adjacent flowrate sensor would be reporting "flow: 4". The work of a PLC could later combine these two simple statements and use syntactic conjunction to give back the complex statement "pressure: 2.8 $\wedge$ flow: 4". With this first notion, we can define simple, complex, and well-formed data statements (wfds). Also, we will define the connectives that can be used for this.

Definition 1 (Simple and Complex Data Statements)**.** *For any simple data statements p, q, ..., any complex statements A, B, ..., the unary connective $\neg$ (Negation), and the binary connectives $\wedge$ (Conjunction), $\vee$ (Disjunction), $\rightarrow$ (Entailment), the following recursive forming rules apply:*

- *(a) For any simple statement p, p is a wfds. Furthermore, if $A = p$, then A is a wfds.*
- *(b) If A is a wfds and $*$ is a unary connective, then $*A$ is a wfds.*
- *(c) If A and B are wfds and $*$ a binary connective, then $A * B$ is a wfds.*
- *(d) There are no more wfds than those defined by the clauses (a), (b) and (c).*

All the different wfds, be they simple or complex, are part of the different data statements generated by the SCADA system, either at the sensor or the PLC level. In this sense, these data statements are definined only w.r.t. a number of propositional connectives, while excluding quantifiers (e. g., $\forall x$, for all x). This is due to multiple reasons. The first one is the ability to create a much more streamlined and simple-to-understand model, making its implementation much easier and excluding complex computational operations. It has also been intended to ensure the results of the formal model rather than going into a much more complex discussion of the different proofs required to ensure the data's validity. Finally, it will make the model lighter and its deployment in resource-constrained devices, such as old PLCs, much easier.

[1] Technology Readiness Levels

 doi:10.20944/preprints202403.1611.v1

Now that the wfds have been defined, we can define and introduce the model. This formal model processes and ensures the logical validity of the different wfds generated by the SCADA. In particular, the validation will be represented by using the $\models$ symbol, which in turn will be used to represent the validity of a wfds A, $\models A$, at the system level. Additionally, this same symbol will be used to represent the validity of the aforementioned wfds at a node a, $a \models A$. The model is also based on the idea of a set of designated nodes. Given how the SCADA systems are always dependent on the topology, these nodes will bear with them some topological requirements. Designated nodes represent those nodes where the data is being held to a much higher level of verification, usually those of the water intake or treatment. In particular, we will refer to this set of nodes by T. In the same sense, all the nodes, not just the designated ones, will be captured in one big set designated K. The final element of the model is the one of the accessibility relation that will be represented by R. This accessibility relation is one of the most important elements of the model and will show how the different nodes connect and interact with each other. Not only that, but it will ensure that the logical validation of the data happens at the right nodes.

The whole model is also created based on a number of definitions and semantic postulates that are key to ensure the different formal results of the model that later will translate in the real/world reliability of the data generated. These definitions and semantic postulates will lead to the topological architecture of the SCADA system. In particular, they will force the creation of a certain number of connections and the eliminition of others, to ensure that the data is processed according to the model to preserve its validity. Given all this, we can define the model as follows:

Definition 2 (WDN-model). *A WDN-model $\mathbb{M}$ is a structure $< T, K, R, \models >$ where K is a non-empty set where every element represents a node of the SCADA system, $T \subseteq K$ where T is the set of designated nodes, and R is a ternary relation on K that represents the connection and data transfer of different nodes. R is subject to the following definitions and postulates for all $a, b, c \in K$:*

$d1.\ a \leq b =_{df} \exists x \in T \mid Rxab$
$d2.\ a = b =_{df} a \leq b \ \& \ b \leq a$
$d3.\ R^2abcd =_{df} \exists x \in K \mid Rabx \ \& \ Rxcd$

$p1.\ a \leq a$
$p2.\ (a \leq b \ \& \ Rbcd) \Rightarrow Racd$

Finally, $\models$ is a valuation relation from K to the set of all data statements such that the following conditions (clauses) are satisfied for every data statement p, A, B and nodes a and b such that $a, b \in K$:

(i). $(a \leq b \ \& \ a \models p) \Rightarrow b \models p$
(ii). $a \models A \wedge B$ *iff* $a \models A \ \& \ a \models B$
(iii). $a \models A \vee B$ *iff* $a \models A$ *or* $a \models B$
(iv). $a \models A \rightarrow B$ *iff for all* $b, c \in K$, $(Rabc \ \& \ b \models A) \Rightarrow c \models B$
(v). $a \models \neg A$ *iff* $a \not\models A$

A crucial aspect of the model that requires thorough explanation involves the ternary accessibility relation, the entailment ($\rightarrow$), and their alignment with the aforementioned clause (iv). To delve into this within the context of WDN and its implications for distributed systems, we will designate a, b, and c as distinct nodes within the network. These nodes are interconnected so that they are represented by the relation $Rabc$. Additionally, we will consider the wfds A and B. Considering this, the ternary accessibility relation mirrors the generation and processing of data in a distributed system. Specifically, wfds A and B, generated and validated by nodes b and c ($b \models A$, $c \models B$), are interconnected through the relation $Rabc$. This connection allows them to collaboratively produce a new and complex data statement as a conditional wfds $A \rightarrow B$. Subsequently, this new statement undergoes validation in the third node, where $a \models A \rightarrow B$.

 doi:10.20944/preprints202403.1611.v1

The above interpretation suggests a dual nature to the concept of the ternary accessibility relation. On the one hand, it can be understood as a form of transitivity, signifying that with $Rabc$, a has access to b, and b has access to c. While this interpretation is the usual and, of course, compelling, it is overly restrictive for certain topologies within the WDN domain. On the other hand, we can alternatively interpret $Rabc$ as the idea that a has access not only to b but also to c. This approach leads us away from a strictly transitive relation and towards a distributed one, where a central node, a, directly accesses the other two interconnected elements, b and c, to infer further knowledge.

As a final point, regarding the metaproperties of the model, it is ensured that it is complete, sound, and decidable. These outcomes are derived through the methodology outlined in [38]. It is essential to highlight that the completeness and soundness of results are not in a strong sense but rather weak. However, this distinction has no consequence for our interest in the WDN use case. In this context, there is no practical distinction between validating all data statements at the SCADA system level versus validating them at the node level. Obtaining the results in a weak sense does not interfere with validating specific data at the node level; they merely constrain it from occurring broadly and in general.

4.1. The Relation R and The Topological Requirements

The model is not only able to describe how the SCADA system can process and manage the data to ensure their validity but also can set the topological requirements of how the different nodes of the system should connect. In particular, this topological requirements are set by the definition d1 to d3, and the semantic postulates p1 and p2. Not all of them carry valuable information for the topology. In particular, d1 only establishes how designated nodes connect with non-designated nodes and creates the hierarchy between the different sets of nodes mentioned earlier. This connection happens as shown in Figure 2(a).

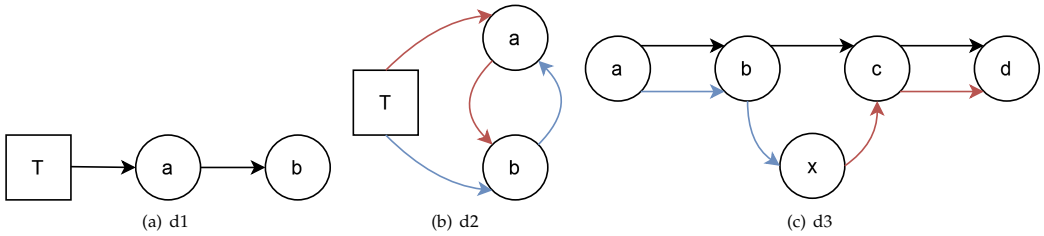


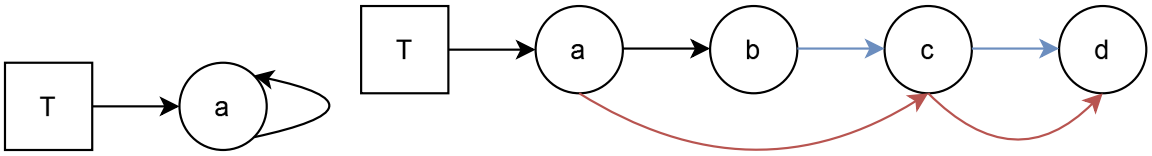
Figure 2. Definitions of the WDN-model as adapted from [36]. (a) Ternary accessibility relation with the designated nodes; (b) Equality as understood in the WDN-model; (c) Extension of the ternary accessibility relation to four elements.

On the other hand, d2 creates a definition of equality for the nodes connected to the set of designated nodes. This definition states that two nodes that connect with the set of designated nodes, and also are connected between them, are just the same node. All this can be seen in Figure 2(b), where a and b are the same nodes. From a formal point of view, this eliminates the surjectiong between T, the set of designated nodes, and K, the set of regular nodes. From a system perspective, it creates a layer of PLC middleware between the water treatment nodes, and the outer nodes of the system's edge. Finally, d3 allows the accessibility relation to be extended from just three (3) elements to four (4) or, virtually infinite, just because these longer relations can be broken down into smaller ones that fit the different clasues of the formal model. In particular, it can be seen in Figure 2(c), where the case for a relation of four (4) elements is represented.

Furthermore, going past the definitions and delving into the semantic postulates, we can get a greater understading of how the model configures the topology of the SCADA system, just as we mentioned earlier. The first semantic postulate, p1, states that for any node a that is directly linked to T, the set of designated nodes, a is going to have access to itself, that is, is self-aware and an accessibility relation starting from T and ending with access to itself. Generally speaking, this means that a will

 doi:10.20944/preprints202403.1611.v1

have some computational capabilites so, for the case of WDN, we will consider a to be a PLC. Let's take a look at the second semantic postulate, p2. It requires creating a new connection: whenever a three-element relation begins with a node that connects to a node that in its turn connects with T, the set of designated nodes, this first node will connect with the last two of the other relation. In this sense, p2 forces the topology like no other definition or postulate, as it requires the creation of newer connections whenever the accessibility relations are developed further away from the designated nodes. Nonetheless, this is not a fatal flaw, but rather an opportunity to enhance the computation capabilities of the whole system, since it will require the addition of new connections with those nodes that have mandatory computational capabilities, thus creating a fog of data processing that otherwise could have been ignored. The extent of both semantic postulates can be seen graphically in Figure 3.



(a) p1 (b) p2

Figure 3. Semantic Postulates of the WDN-model as adapted from [36]. (a) Reflexivity of fog nodes; (b) Topological requirement of long chains of nodes.

5. Implementation

To show the model's validity and application, first we need to transfer it to a real world implementation as a software artifact. We translate the model and its topological requirements into an algorithm and then produce the software artifact that supports the validation.

5.1. Algorithm

Given that all the elements of the model, the set of nodes K, the set of designated nodes T, and the relations between nodes R, are part of the SCADA system that would regulate a smart WDN, we just need to focus on capturing the different topological requirements created by the definitions and semantic postulates. All these can be summed up in the following algorithm. For that matter we will use the the colour coding in which designated nodes will appear in blue, regular nodes will appear in brown, and actions to be taken will appear in red.

Algorithm 1 Topological requirements of the model

if [node_a connects with (designated_node and node_b)] and [node_b connects with (designated_node and node_a] **then**

node_b disconnects from designated_node ▷ d2

end if

if len(nodes_connections)≥ 4 and is even **then**

(new_node connects with node_(len-1) and node_len) and (node_(len-1) disconnects from node_len) ▷ d3

end if

if node_a connects with designated_node **then**

[(node_a connects with node_a) ▷ p1

end if

if [(designated_node connects with node_a) and (node_a connects with node_b)] and [(node_b connects with node_c) and (node_c connects with node_d)] **then**

node_a connects with node_c ▷ p2

end if

 doi:10.20944/preprints202403.1611.v1

Going into details, the first `if` statement captures d2, which leads to not having two different nodes connected and, in turn, both connected to the set of designated nodes. The second `if` statement, representing d3, states that whenever a chain of nodes has a length equal to or superior to 4, a new node will be inserted between the second to last and last nodes, breaking the connection between these two. The third of the `if` statements focus on p1 and shows that whenever a node is connected to the set of designated nodes, it will have access to itself. In particular, this will be represented further by the node having computation capabilities to process its own data. Finally, the fourth and last `if` statement deals with p2. This if statement describes that whenever a connection of four nodes spawns from a node connected directly to the set of designated nodes, a connection will be made from the first non-designated one to the third.

5.2. Architecture

With the algorithm described above, we can proceed to create a software artifact so that the model can be validated. For that matter, the artifact will be codified using the programming language Python 3.12.2, making use of the libraries `wntr`, `chama`, `numpy`, `pandas`, `matplotlib`, and `networkX`. In particular, `wntr` was used as a mean to load the corresponding files for the different scenarios, modifying them (if required), and performing the corresponding simulations; `chama` was used to identify the place in the topology where sensors must be set in order to ensure a certain property of the SCADA system, such as for example given a physical poissoning attack on the waters, the polluted ones will be detected as far as possible; `numpy` provided support to work with multidimensional numeric matrices as well as a set of mathematical functions to operate on these; `pandas` was used to process the data generated by the simulation of the different scenarios; `matplotlib` was utilized to create the different graphics used to understand the results derived from the simulations; and `networkX` supported the manipulation of the different graphs that come to represent the control network deployed on top of the simulation of the physical network providing an implementation of the Prim's Minimal Spanning Tree algorithm [39] that identifies the set of links between elements in a network that scores for the shortest length of links.

In addition to all these libraries, we also used *epanetCPA* [37]. This additional software was developed as "an open-source object-oriented MATLAB toolbox for modelling the hydraulic response of water distribution systems to cyber-physical attacks". With this in mind, *epanetCPA* has the advantage of having several realistic scenarios as well as a number of attacks associated with them. For what matters to us, epanetCPA provided a realistic scenario upon which to build the whole simulation of the WDN on which the model, condensed into the algorithm, would be deployed. It is important to mention that it was possible to use epanetCPA thanks to the interoperability that `wntr` provides. Nonetheless, epanetCPA can provide us with the different variables of the WDN at each moment of the simulation, allowing us to make queries and filters to determine the most valuable variables for the deployment of the model. All this will help to put together an implementation that ensures that the model is actually being used for the deployment of nodes of the SCADA system, and thus, help us in the validation of the formal model itself.

With the above in mind, we decide which variable should be used to monitor the different nodes implemented in the simulation. The variables that were considered were *critical nodes*, *higher demand nodes*, *water quality nodes*, and *vulnerable areas nodes*. Going into detail, critical nodes represented the most important elements of the network, such as intersections, intakes, outtakes, or nodes closer to other critical infrastructures. Higher demand nodes correspond to those within the WDN with the highest registered water usage of all the network. Water quality nodes are those nodes that are key when determining if the quality of water has gone under a certain threshold or if it has been contaminated in one way or another. Finally, vulnerable areas nodes are the ones that correspond to nodes with an importance relative to their geographical location; some example of this could be the nodes in a busy traffic intersection, or those in a difficult to reach area. With all this, it was decided that the simulation would run on the variable of critical nodes, as it is the most comprehensive one and the one that is most used in real-world scenarios.

 doi:10.20944/preprints202403.1611.v1

Afterwards, it is up to us to decide which parameter will be captured by the different network nodes. Given that we are using `wntr` as the library for processing the simulation data, we already have a comprehensive list of the parameters that can be used. These are *Demand*, how much water is used in the different nodes; *Head*, the height of the water column at a given point in the network; *Pressure*, the water pressure at the different nodes; *Quality*, which represents water quality, including the different chemicals that are present in the water; *Flowrate*, the flow of the water on the different parts of the network; *Velocity*, the speed of the water on the different pipes; *Headloss*, the loss of pressure as water flows through the pipes; *Linkquality*, similar to Quality, but focusing on very specifics links of the network; *Status*, the state (operative, open, close...) of the different nodes of the network; *Setting*, shows the different configurations of the nodes; *Rxnrate*, the speed on which different chemicals react within the network; *Frictionfact*, showing the friction factor of the pipes. Out of all of these, both Quality and Flowrate are probably the most interesting, descriptive, and comprehensive. Nonetheless, given `chama`'s optimization for detecting the quality of the water and its algorithm for deploying the optimal sensors for this, the variable chosen has been Quality.

Furthermore, how the impact will be measured in the network needs to be decided. The different options respond to the variety of metrics available in `chama`, leading to the optimal placement of sensors in the network. These are *Minimal detection time*, targeting to detect network events in the shortest time possible. *Minimal undetected impact*, dedicated to minimizing the impact of not detecting an important event in the network. *Maximal coverage*, focusing on ensuring that the network has the maximum coverage possible. *Minimal cost*, supporting a deployment on which the network implementation cost is at a minimum. And *Specific scenarios*, provides support for custom situations. Given that we are targeting a general-purpose network, Minimal detection time has been selected.

Another key element in this architecture is the strategy used to connect the different nodes. It is well known that for connecting the nodes of a network, there are two different options: it can be done by using a wired connection or by using a wireless one. The wired connection provides greater stability and increased security, is less prone to interference, and has less latency. Nonetheless, it is also possible to list some of its disadvantages, like an increased cost on the deployment of the nodes, adding to the complexity of expanding the network or limiting the mobility. On the other hand, wireless connections have greater flexibility and mobility and lower installation costs. It also adds to the ease of expansion. Yet it has downsides, like being more prone to interference and having higher latency. Given that choosing one or the other is a matter of preference linked to a specific situation, the implementation has considered both, so it can be decided if the deployment will be wired, wireless, or on a link-by-link basis.

With the above, and as we have discussed, it is clear that the implementation considers the connection of the nodes of the WDN, but this connection happens differently in each case. In the case of the wireless connection, it is established as the shortest Euclidian distance. On the other hand, when we look at the case of the wired connection, it works upon the topology of the simulation using Prim's algorithm. In particular, Prim's algorithm is greedy, and its purpose is to find the minimum spanning tree for a graph in a weighted and undirected manner. To implement Prim's algorithm, and the corresponding method for establishing the initial network of the nodes we used `networkX` as it facilitates the work done on graphs.

All the previous work is needed to support the algorithm in which we have condensed the topological requirements of the model. With that in mind, we can describe the process followed whenever the implementation is started. First, the artifact will load the topology file of the different elements of the network, the number of initial nodes, and their deployment. Then it will perform a check on the initial network using the following steps:

1 If a node is connected to the set of desginated nodes T and it is not a PLC, i. e., it has computation capabilities, the node will disconnect from the original node and will connect to a PLC.
2 If two PLC are connected to T and to each other, the connection of one of them with T will be elimintated.

 doi:10.20944/preprints202403.1611.v1

3 If there is a chain of nodes with a length equal to to or over 4, and the length is an even number, a new node will be inserted in the second to last position, breaking the previous connections.
4 If a chain of over 3 regular nodes connects to a PLC, the second regular node will also be connected to the PLC.

Afterward, it will register the number of nodes aggregated to the preexisting network. Then it will check node by node all the new ones following the steps:

1 It will check if the new node is connected to a node of the set T, the set of designated nodes, and a previously existing one. It will also check if the previous one is connected to T. If that is the case, then the new node will be eliminated.
2 It will ensure the computation capabilities of the new node in case it is connected to T. If it does not have those, it will be eliminated.
3 It will evaluate if the connections described in p2, the fourth `if` statement of the algorithm, exist. Otherwise, it will create them.
4 It will go over the length of the chains of nodes; if those are over 4 elements, the number of elements is even. If those conditions are satisfied, a new node will be added in the second to last position of the chain, breaking the previous connections.

All the previous notions are summed up in the diagram of Figure 4. Finally, let us state that this implementation's overall critical requirements are the SCADA connections, the node list both acquired by using `chama`, and the different types of connections that allow for the calculation of the distances with the use of Prim's algorithm or Euclidean distances depending on the type of connection.

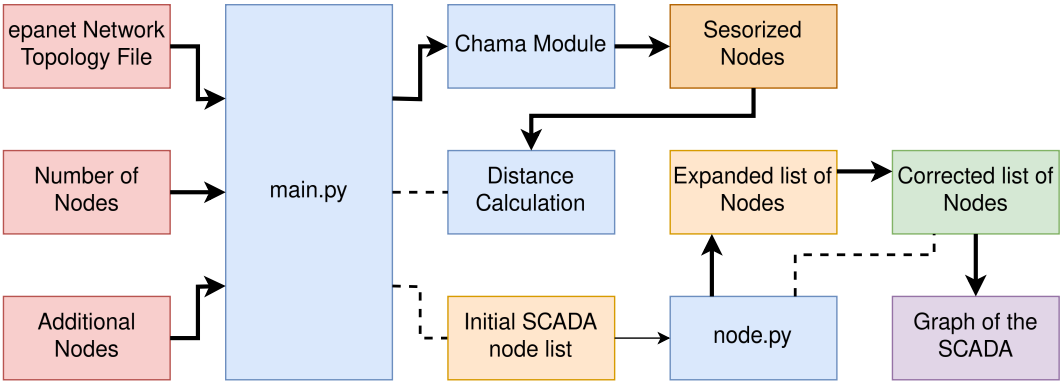


Figure 4. Architecture of the implementation. Red nodes represent external information that is provided to the main components of the implementation (blue nodes). Yellow nodes represent intermediate results that culminate in the final data compiled in the green node. The purple node constitutes a representation of the results.

6. Validation

We use the C-Town network scenario [40] (Figure 5) for our model and architecture validation. C-Town is a real-world, medium-sized network used extensively in WDN simulations for benchmarking different models and analysis [41]. It comprises 429 pipes and 388 demand nodes, and 7 tanks guarantee distribution. It also contains a key element in the shape of a pumping station that acts as the scenario's water intake. In particular, this pumping station will act as T, the set of designated nodes and the central node of the SCADA system whose deployment we are simulating. Let us further state that C-Town is a scenario from which we only take the physical elements, not any of the SCADA elements added later.

 doi:10.20944/preprints202403.1611.v1

Figure 5. C-Town as adapted from [40]. This scenario is a real-world, medium-sized network used extensively in WDN simulations for benchmarking different models and analysis, constituting an industry standard.

Given the C-Town scenario, our first intention is to determine which and how many nodes we will be using for the validation. As stated earlier, we will focus on the critical nodes and how long they take to detect network events. For that matter, the `chama` module of the implementation can be used to assess the relation between detection time and the number of sensors (Figure 6): The performance gains are not relevant once we have introduced seven (7) sensors.

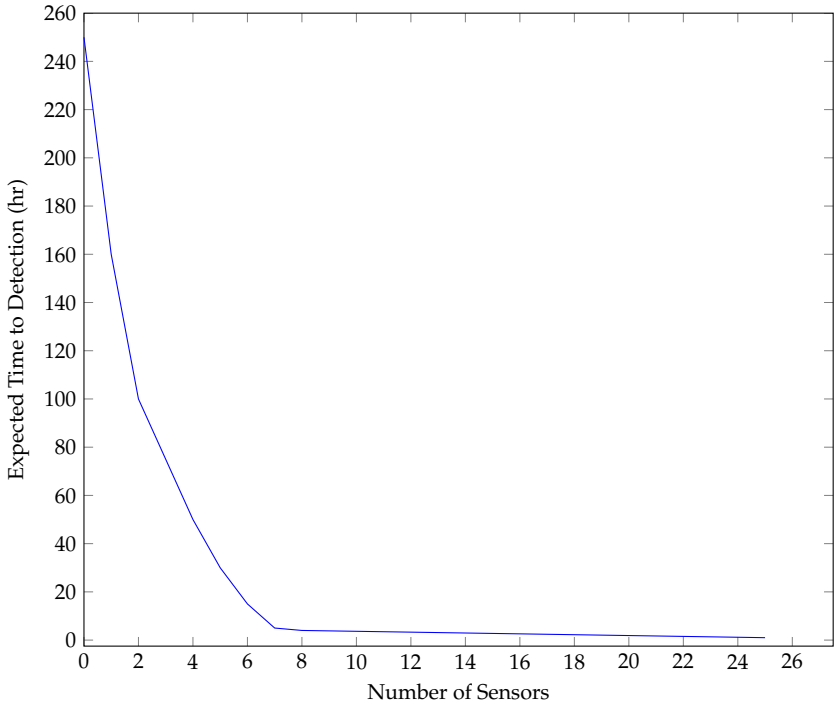


Figure 6. Number of critical nodes against detection time. It can be observed that after the placement of 7 sensors the detection time reaches a point of diminishing returns.

 doi:10.20944/preprints202403.1611.v1

Since the intention is not just to use a real-world scenario, but also create a validation that is based on what could be real-world decisions, we set the number of sensors that will be deployed as part of the SCADA system to seven (7). When deciding where the different sensors will be placed, we use `chama` again. This allows us to determine not only the number of sensors that we will be using but also which of the nodes of the topological scenario of C-Town are the ones that we will be using. Figure 7 shows the nodes that have been selected. It is crucial to mention that, despite being also highlighted, the pumping station (S1), that acts as the SCADA central node and as the set of designated nodes (T), is not included among the number of nodes outputted by `chama`. This is because the decision is made by keeping in mind that there is always a minimum of one node in the SCADA system. Additionally, it should be noted that the module computes the number of *sensors*. In this scenario, the pumping station is not recognized as a sensor but as the central processing unit without any sensing capabilities.

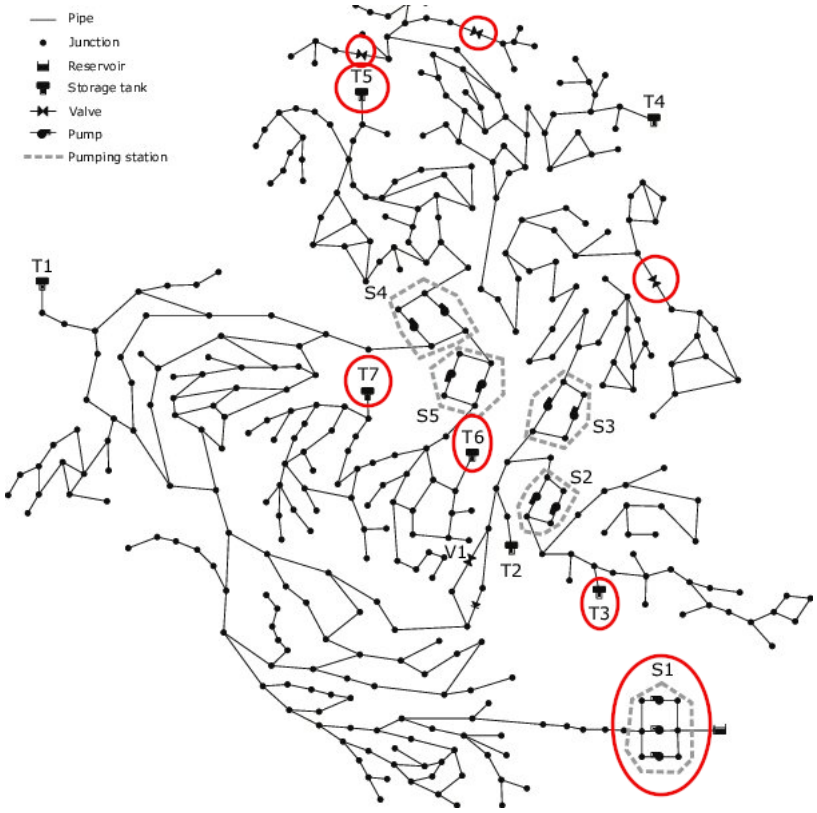


Figure 7. Critical nodes of C-Town as selected for the implementation. The critical nodes have been selected using chama and their number answers to the limit before the point of diminishing returns as it appears in Figure 6.

With all of the above, we can proceed to describe the deployment of the different nodes that we have selected. For that matter, and by using the support provided by `networkX`, we get the result shown in Figure 8(a) when deploying all the nodes with a wired connection to enable communication. In this, and all of the following diagrams, the node in red corresponds to the central node of the SCADA; the node in green is the one that is a PLC, that is, a sensor with computing capabilities as well as sensing ones, establishing them as self-aware. Finally, the nodes in blue represent regular sensors that are not required to have computing capabilities. As seen in Figure 8(a), the PLC connects directly to the SCADA central node, and another acts as the bridge between two different nodes and the PLC. It is easy to understand that there is no rationale behind deploying the different nodes besides being the ones selected by Prim's algorithm.

 doi:10.20944/preprints202403.1611.v1

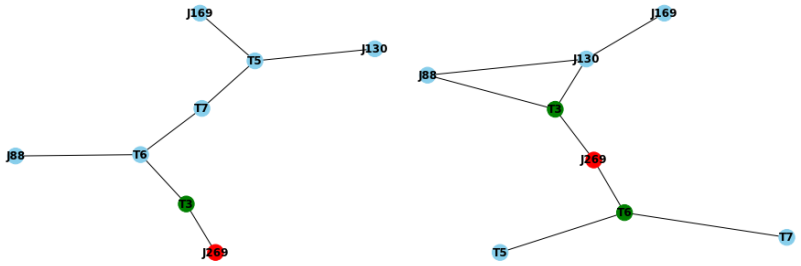


Figure 8. (a). Original node deployment scenario (left), and (b). Verified node deployment scenario - wired connection (right). This figure shows the changes that a SCADA deployment should undergo to ensure the implementation of the model and the validity of the data.

Nonetheless, if we look at Figure 8(b), we see that the outcome of making the deployment compliant with the model is quite different. In this case, T6, a regular node in the original scenario, has become a PLC and now has computing capabilities. Additionally, the nodes T5 and T7 now reach out to J269, the SCADA central node, just through it. On the other side of the diagram, we see that node J88 now acts as an accessory connection between J130 and the PLC T3. Finally, the scenario provides that J169 sits at the end of a chain. This series of changes, which can look drastic, ensures the validity of all the data produced by the system just as a byproduct of compliance with the topology provided by the formal model. Further on, this not only provides the validity of the data statements generated by the different sensors, but it also ensures the validity of the inferences that could be made out of those data. Making the system much more resilient, intelligent, and skilled whenever managing data.

In a similar fashion to the previous scenario, if the connection is decided to be implemented in a wireless manner rather than through the usual method of using a wired connection, the implementation can obtain a solution that is compliant with the formal model and its topology, just as described above. The result, which can be seen in Figure 9, shows some crucial differences with the wired scenario as, for example, there is no extra PLC. The wireless verified scenario does not require any extra PLC but, on the other hand, requires extra nodes to ensure the right length of the chains of nodes. In this case, extra nodes are situated between T6 and J88, T5 and J130, and T5 and J169. Again, all these simple changes, adding extra regular nodes almost at the end of certain chains, are more than enough to ensure that the properties of the model defined above are used in the best manner.

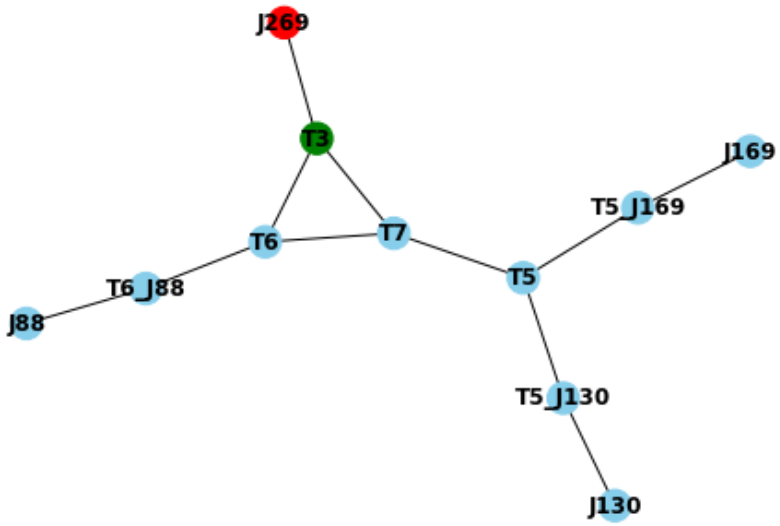


Figure 9. Verified node deployment scenario - wireless connection. In addition to Figure 8(b), here we show the changes that the SCADA system of Figure 8(a) should undergo to ensure the implementation of the model in a wireless fashion.

 doi:10.20944/preprints202403.1611.v1

The final scenario for us to tackle is when a node has been added to the preexisting deployment. Something that could easily happen whenever we face the usual case of the digital transformation of a WDN. In this case, a PLC will be added right after node J88 in the solution corresponding to the wired scenario, as seen in Figure 10(a). It is easy for us to imagine a situation in which the new PLC has been deployed in an area with low reception and that geographically is more difficult to connect, thus needing extra computational capabilities to understand what is occurring in that local area of the WDN. Of course, adding a new node does not imply that automatically it will not follow with the directives propugnated by the model, but in this case, it does.

The solution is quite simple and could be arranged before deploying the new PLC to ensure the SCADA system's validity to the model. Thus, the only real requirement would be to add a node between the new PLC, added_1, and the node J88. Nonetheless, given that the implementation considers the whole network rather than just the new additional node, it also detects that it is possible to strengthen the network adding nodes between the nodes T5 and T6, and T7 and T6. These changes can be seen in Figure 10(b). This would allow us to ensure a higher response on the event detection while keeping all the properties of the model online.

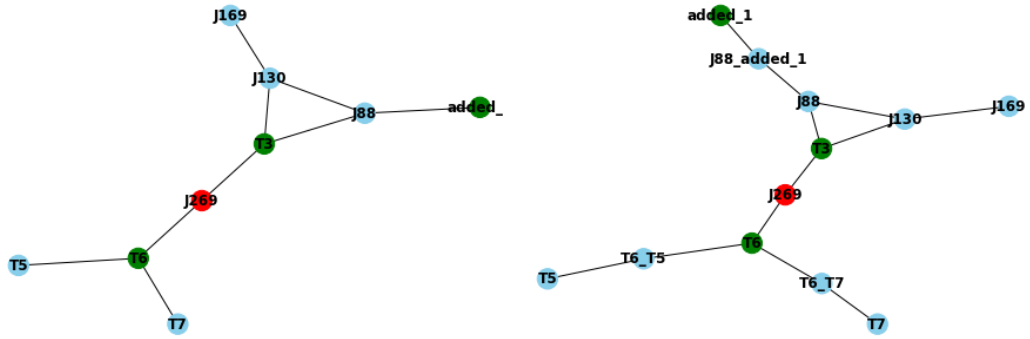


Figure 10. (a). Added node scenario (left), and (b). Verified added node scenario (right).

With all of the above, we have been able to demonstrate how the formal model can be codified with a simple algorithm ensuring that the data collected and processed, as well as the new inferred one, are valid. It also supports adding new nodes to the network and allows the possibility to expand the SCADA system on the fly while keeping all the aforementioned properties.

As a final point to be discussed, it is also possible to show that the model implementation is not only satisfactory, as has been mentioned, but it also provides a further advantage against traditional formal methods such as model checking [30]. Whenever performing model checking, the outcome ensures that a system's current implementation is formally solid and meets the model properties. On the other hand, using a formal model to guide a system's deployment ensures that any system's implementation meets the model properties. This, in particular, allows for a flexible deployment approach, which can be carried out incrementally without needing to model and check each incremental implementation. This helps to reduce the costly effort of creating a formal representation of the system at different times while enabling a solid foundation that provides the same results.

Going into further detail, showcasing some of the cases the model helps validate is possible. In particular, we have stated that one of the main positive characteristics of the model is that it ensures the validity of the different inferences that can be made within the model-generated data. In this sense, we can assume a simple inference of the type $A \rightarrow B$ for any data statements A and B. In the case of Figure 8(a), if A was produced in T7, and B in T5, given that we have the relation RT6T7T5, that would lead, theoretically, to the conclusion that $A \rightarrow B$ in T6. Nonetheless, this is impossible, as T6 has no computing capabilities as of the case displayed in Figure 8(a). On the other hand, if we look at Figure 8(b), where the deployment has changed according to the model, we can easily see that this issue has been solved thanks to the implementation of the model, as now T6 has computational capabilities.

 doi:10.20944/preprints202403.1611.v1

This shows that the implementation not only coincides with what is being described in the model but can also provide obvious and strong support for the validity of inferences.

6.1. Limitations

Despite the validation we carried out, there are still some limitations to our method and its implementation to be mentioned. In particular, using a formal model to drive an implementation might pose some difficulties, as most of the formal results require an extensive knowledge of different techniques required to obtain these. On the bright side, it is always possible to refer to previous specialized literature on which the results have already been published. However, this might not work straight ahead for *ad hoc* formal models, and some tailoring might be required. Another related limitation comes from the applicability of the methodology at scale. An expert on formal methods is required to extrapolate the knowledge contained in the model into an algorithm. This leads to a possible bottleneck in the case that multiple models are used, requiring extensive effort from a limited number of highly skilled experts. When looking at the validation, it is possible to raise some concerns about the model's impact on the network itself, as the model has not been fully tested in a real-world environment. It is also possible that some of the topological requirements would lead to deployment in an area that is difficult to access. Nonetheless, this can be mitigated by the wireless connection that is already considered in the implementation done for the validation of the model. Overall, these are simple drawbacks that are to be accepted in order to attain the different extensive benefits of the model.

7. Conclusion

This paper has defined a formal model that ensures that the digital transformation of WDN would be easy, reliable, and flexible, further providing extra assurance regarding the data generated, processed, and managed within the SCADA system of the WDN. This model supports the deployment, not only of the first iteration of the WDN but of its possible expansions as time passes. This is crucial as the digital transformation of legacy critical infrastructures is not expected to happen at once. Furthermore, since the model transfers its properties to the SCADA system, it significantly increases the reliability of the generated data statements. We have demonstrated the model applicability in a real-world scenario, showing how it will adapt a rationally sound deployment into one that keeps the same decisions but ensures certain properties that otherwise would be unreachable.

Future research lines include developing a version of the model that allows node validation of the data statements generated. It is also interesting to further iterate on the implementation and apply it to an industry setting. Another future line of research could be adapting the model so it takes into account data tampering attacks leading to invalid inferences; thanks to that the model could also be used for the detection of attacks on the SCADA system. Finally, given the extension of WDN and how prone they are to failure, it would be interesting to modify the model to deal with inconsistent data as proposed in [42].

Author Contributions: Conceptualization, J.M.B., J.M.A., J.C.D. and F.C.; validation, J.M.B.; formal analysis, J.M.B.; investigation, J.M.B.; writing—original draft preparation, J.M.B. and J.M.A.; writing—review and editing, J.M.B., J.M.A., J.C.D. and F.C.; visualization, J.M.B. and J.M.A.; project administration, J.M.A. and J.C.D; funding acquisition, J.M.A., F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the SUNRISE project and the CEDAR project, funded from the Horizon Europe research programme (2021-2027) under grant agreements no. 101073821. and no. 101135577

Acknowledgments: The authors would like to thank Alana Fernández Basquero for her contributions and efforts on the technical aspects of the implementation.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Chren, S.; Rossi, B.; Pitner, T. Smart grids deployments within EU projects: The role of smart meters. 2016 Smart Cities Symposium Prague (SCSP), 2016, pp. 1–5. doi:10.1109/SCSP.2016.7501033.
2. Zhu, Z.Y.; Xie, H.M.; Chen, L. ICT industry innovation: Knowledge structure and research agenda. *Technological Forecasting and Social Change* **2023**, *189*. doi:10.1016/j.techfore.2023.122361.
3. Carriço, N.; Ferreira, B.; Antunes, A.; Caetano, J.; Covas, D. Computational Tools for Supporting the Operation and Management of Water Distribution Systems towards Digital Transformation. *Water (Switzerland)* **2023**, *15*. doi:10.3390/w15030553.
4. Yao, C.; Fan, B. Spatiotemporal Vulnerability Analysis of Large-Scale Infrastructure Systems under Cascading Failures: Case of Water Distribution Networks. *Journal of Infrastructure Systems* **2023**, *29*. doi:10.1061/JITSE4.ISENG-1677.
5. Pereira, L.; Morais, D.; Figueira, J. Using criticality categories to evaluate water distribution networks and improve maintenance management. *Sustainable Cities and Society* **2020**, *61*. doi:10.1016/j.scs.2020.102308.
6. Boyer, S.A. *SCADA: supervisory control and data acquisition*; International Society of Automation, 2009.
7. Routley, R.; Meyer, R.; Brady, R.; Plumwood, V. *Relevant Logics and Their Rivals 1*; Ridgeview: Atascadero, CA, 1983.
8. Blanco, J.M.; Rossi, B.; Pitner, T. A Time-Sensitive Model for Data Tampering Detection for the Advanced Metering Infrastructure. 2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS), 2021, pp. 511–519. doi:10.15439/2021F106.
9. Mays, L.W., Ed. *Water Distribution System Handbook*, first edition. ed.; McGraw-Hill Education: New York, 2000.
10. Alves, D.; Blesa, J.; Duviella, E.; Rajaoarisoa, L. Topological analysis of water distribution networks for optimal leak localization. 2023, Vol. 1136. Cited by: 0; All Open Access, Bronze Open Access, doi:10.1088/1755-1315/1136/1/012043.
11. Carriço, N.; Ferreira, B.; Antunes, A.; Caetano, J.; Covas, D. Computational Tools for Supporting the Operation and Management of Water Distribution Systems towards Digital Transformation. *Water* **2023**, *15*. doi:10.3390/w15030553.
12. Ciliberti, F.G.; Berardi, L.; Laucelli, D.B.; Giustolisi, O. Digital Transformation Paradigm for Asset Management in Water Distribution Networks. 2021 10th International Conference on ENERGY and ENVIRONMENT (CIEM), 2021, pp. 1–5. doi:10.1109/CIEM52821.2021.9614864.
13. Wu, Z.Y.; Chew, A.; Meng, X.; Cai, J.; Pok, J.; Kalfarisi, R.; Lai, K.C.; Hew, S.F.; Wong, J.J. High Fidelity Digital Twin-Based Anomaly Detection and Localization for Smart Water Grid Operation Management. *Sustainable Cities and Society* **2023**, *91*, 104446. doi:https://doi.org/10.1016/j.scs.2023.104446.
14. Zekri, S.; Jabeur, N.; Gharrad, H. Smart Water Management Using Intelligent Digital Twins. *Comput. Informatics* **2022**, *41*, 135–153. doi:10.31577/cai_2022_1_135.
15. Korotkova, N.; Benders, J.; Mikalef, P.; Cameron, D. Maneuvering between skepticism and optimism about hyped technologies: Building trust in digital twins. *Information & Management* **2023**, *60*, 103787. doi:https://doi.org/10.1016/j.im.2023.103787.
16. Adedeji, K.B.; Ponnle, A.A.; Abu-Mahfouz, A.M.; Kurien, A.M. Towards Digitalization of Water Supply Systems for Sustainable Smart City Development - Water 4.0. *Applied Sciences* **2022**, *12*. doi:10.3390/app12189174.
17. Mohapatra, H.; Mohanta, B.K.; Nikoo, M.R.; Daneshmand, M.; Gandomi, A.H. MCDM-Based Routing for IoT-Enabled Smart Water Distribution Network. *IEEE Internet Things J.* **2023**, *10*, 4271–4280. doi:10.1109/JIOT.2022.3216402.
18. Guo, Y.; Wang, S.; Taha, A.F.; Summers, T.H. Optimal Pump Control for Water Distribution Networks via Data-Based Distributional Robustness. *IEEE Trans. Control. Syst. Technol.* **2023**, *31*, 114–129. doi:10.1109/TCST.2022.3167844.
19. van Lagen, G.; Abraham, E.; Esfahani, P.M. A Bayesian Approach for Active Fault Isolation With an Application to Leakage Localization in Water Distribution Networks. *IEEE Trans. Control. Syst. Technol.* **2023**, *31*, 761–771. doi:10.1109/TCST.2022.3201334.
20. Lasi, H.; Fettke, P.; Kemper, H.G.; Feld, T.; Hoffmann, M. Industry 4.0. *Business and Information Systems Engineering* **2014**, *6*, 239 – 242. Cited by: 2895, doi:10.1007/s12599-014-0334-4.
21. Alcaraz, C.; Zeadally, S. Critical infrastructure protection: Requirements and challenges for the 21st century. *International Journal of Critical Infrastructure Protection* **2015**, *8*, 53–66. doi:https://doi.org/10.1016/j.ijcip.2014.12.002.

22. Yadav, G.; Paul, K. Architecture and security of SCADA systems: A review. *International Journal of Critical Infrastructure Protection* **2021**, *34*, 100433. doi:https://doi.org/10.1016/j.ijcip.2021.100433.
23. Dolatshahi-Zand, A.; Damghani, K.K.; Raissi, S. An evolutionary approach with reliability priority to design Scada systems for water reservoirs. *Evol. Syst.* **2022**, *13*, 499–517. doi:10.1007/s12530-022-09438-0.
24. Rigatos, G.; Serpanos, D.; Zervos, N. Detection of attacks against power grid sensors using Kalman filter and statistical decision making. *IEEE Sensors Journal* **2017**, *17*, 7641–7648.
25. Suaboot, J.; Fahad, A.; Tari, Z.; Grundy, J.; Mahmood, A.N.; Almalawi, A.; Zomaya, A.Y.; Drira, K. A Taxonomy of Supervised Learning for IDSs in SCADA Environments. *ACM Computing Surveys* **2020**, *53*. Cited by: 37; All Open Access, Green Open Access, doi:10.1145/3379499.
26. Ducharlet, K.; Travé-Massuyès, L.; Le Lann, M.V.; Miloudi, Y. A multi-phase iterative approach for anomaly detection and its agnostic evaluation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2020**, *12144 LNAI*, 505 – 517. Cited by: 3; All Open Access, Green Open Access, doi:10.1007/978-3-030-55789-8_44.
27. Ren, Z.; Wang, S.; Zhang, Y. Weakly supervised machine learning. *CAAI Transactions on Intelligence Technology* **2023**, *8*, 549–580, [https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/cit2.12216]. doi:https://doi.org/10.1049/cit2.12216.
28. Wang, P.; Xiang, M.; Lei, R. Research on Multi-Sensor Data Fusion Algorithm for Monitoring of Power Distribution Station. *Smart Innovation, Systems and Technologies* **2022**, *257*, 207 – 213. Cited by: 1, doi:10.1007/978-981-16-5164-9_25.
29. Bjesse, P. What is formal verification? *SIGDA Newsl.* **2005**, *35*, 1–es. doi:10.1145/1113792.1113794.
30. Mercaldo, F.; Martinelli, F.; Santone, A. Model Checking for Real-Time Attack Detection in Water Distribution Systems. *Informatics and Automation* **2022**, *21*, 219–242. doi:10.15622/ia.21.2.1.
31. Vistbakka, I.; Troubitsyna, E. Modelling and Verification of Safety of Access Control in SCADA Systems. Risks and Security of Internet and Systems; Garcia-Alfaro, J.; Leneutre, J.; Cuppens, N.; Yaich, R., Eds.; Springer International Publishing: Cham, 2021; pp. 354–364.
32. Mercaldo, F.; Martinelli, F.; Santone, A. Real-Time SCADA Attack Detection by Means of Formal Methods. 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2019, pp. 231–236. doi:10.1109/WETICE.2019.00057.
33. AL-Dahasi, A.E.M.; Saqib, B.N.A. Attack tree Model for Potential Attacks Against the SCADA System. 2019 27th Telecommunications Forum (TELFOR), 2019, pp. 1–4. doi:10.1109/TELFOR48224.2019.8971181.
34. Kousar, S.; Nazir, A.Z.; Ali, T.; Alkhammash, E.H.; Hadjouni, M. Formal Modeling of IoT-Based Distribution Management System for Smart Grids. *Sustainability* **2022**, *14*, 4499.
35. Jakaria, A.H.M.; Rahman, M.A.; Gokhale, A. A Formal Model for Resiliency-Aware Deployment of SDN: A SCADA-Based Case Study. 2019 15th International Conference on Network and Service Management (CNSM), 2019, pp. 1–5. doi:10.23919/CNSM46954.2019.9012715.
36. Blanco, J.M.; Ge, M.; del Alamo, J.M.; Dueñas, J.C.; Cuadrado, F. A formal model for reliable digital transformation of water distribution networks. *Procedia Computer Science* **2023**, *225*, 2076–2085. 27th International Conference on Knowledge Based and Intelligent Information and Engineering Sytems (KES 2023), doi:https://doi.org/10.1016/j.procs.2023.10.198.
37. Taormina, R.; Galelli, S.; Douglas, H.; Tippenhauer, N.; Salomons, E.; Ostfeld, A. A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems. *Environmental Modelling & Software* **2019**, *112*, 46–51. doi:https://doi.org/10.1016/j.envsoft.2018.11.008.
38. Robles, G.; Blanco, J.M.; López, S.M.; Paradela, J.R.; Recio, M.M. Relational semantics for the 4-valued relevant logics BN4 and E4. *Logic and Logical Philosophy* **2016**, *25*, 173–201. Number: 2, doi:10.12775/LLP.2016.006.
39. Prim, R.C. Shortest connection networks and some generalizations. *The Bell System Technical Journal* **1957**, *36*, 1389–1401. doi:10.1002/j.1538-7305.1957.tb01515.x.
40. Pournaras, E.; Taormina, R.; Thapa, M.; Galelli, S.; Palleti, V.; Kooij, R. Cascading Failures in Interconnected Power-to-Water Networks. *SIGMETRICS Perform. Eval. Rev.* **2020**, *47*, 16–20. doi:10.1145/3397776.3397781.

 doi:10.20944/preprints202403.1611.v1

41. Taormina, R.; Galelli, S.; Tippenhauer, N.O.; Salomons, E.; Ostfeld, A. Characterizing Cyber-Physical Attacks on Water Distribution Systems. *Journal of Water Resources Planning and Management* **2017**, *143*, 04017009. doi:10.1061/(ASCE)WR.1943-5452.0000749.
42. Blanco, J.M.; Ge, M.; Pitner, T. Modeling Inconsistent Data for Reasoners in Web of Things. *Procedia Computer Science* **2021**, *192*, 1265–1273. doi:10.1016/j.procs.2021.08.130.