*water*

MDPI

*Article*

# WDSchain: A Toolbox for Enhancing the Security Using Blockchain Technology in Water Distribution System

**Haitham H. Mahmoud** [1], **Wenyan Wu** [1,*] **and Yonghao Wang** [2]

[1] School of Engineering and Built Environment, Birmingham City University, Birmingham B4 7XG, UK; haitham.mahmoud@bcu.ac.uk

[2] School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK; yonghao.wang@bcu.ac.uk

* Correspondence: wenyan.wu@bcu.ac.uk

**Abstract:** This work develops a toolbox called WDSchain on MATLAB that can simulate blockchain on water distribution systems (WDS). WDSchain can import data from Excel and EPANET water modelling software. It extends the EPANET to enable simulation blockchain of the hydraulic data at any intended nodes. Using WDSchain will strengthen network automation and the security in WDS. WDSchain can process time-series data with two simulation modes: (1) static blockchain, which takes a snapshot of one-time interval data of all nodes in WDS as input and output into chained blocks at a time, and (2) dynamic blockchain, which takes all simulated time-series data of all the nodes as input and establishes chained blocks at the simulated time. Five consensus mechanisms are developed in WDSchain to provide data at different security levels using PoW, PoT, PoV, PoA, and PoAuth. Five different sizes of WDS are simulated in WDSchain for performance evaluation. The results show that a trade-off is needed between the system complexity and security level for data validation. The WDSchain provides a methodology to further explore the data validation using Blockchain to WDS. The limitations of WDSchain do not consider selection of blockchain nodes and broadcasting delay compared to commercial blockchain platforms.

**Keywords:** water distribution system; blockchain technology; security; IoT; EPANET; consensus mechanisms

check for updates

**Citation:** Mahmoud, H.H.; Wu, W.; Wang, Y. WDSchain: A Toolbox for Enhancing the Security Using Blockchain Technology in Water Distribution System. *Water* **2021**, *13*, 1944. https://doi.org/10.3390/w13141944

Academic Editor: Francesco De Paola

Received: 27 May 2021
Accepted: 8 July 2021
Published: 14 July 2021

## 1. Introduction

Integrated open-source tools for water distribution systems (WDS) have been widely investigated in the literature for automating and securing the WDS from cyber-physical attacks. These integrated tools can provide operators with insights into the well-being of the network, and inform and mitigate any hacking attempt. These simulation tools that can monitor, diagnose, and analyse the network are crucial to assess the performance of the water networks in standing against information technology (IT) hacking attempts. A comprehensive review of all open-source tools for WDS is discussed (see Section 2). The development of Internet-of-Things (IoT) and smart devices in the WDS, along with the trend of connecting the operational technologies (OT) (e.g., Pumps, sensors, PLCs, pipelines) and IT (e.g., cloud computing, edge computing, blockchain) with the Internet will bring potential risks for the security of cyber physical systems.

Therein, it is essential to use state-of-the-art technologies that can provide secure integration of IT and OT and automation of the systems in WDS. The Maroochy Shire attack in Queensland took place 20 years ago in Australia, and since then, WDS have been targeted by more than 15 major attacks according to [1]. This risk is increasing by integrating the physical assets with the cyber systems and connecting the physical assets to the Internet. In addition, water systems are ranked within the top five industries in terms of the impact of the attacks according to [2]. Attacks will not only disrupt the availability of the service, but could also cause a catastrophic disaster and endanger the public health and safety of human beings. For instance, manipulation of the control signal of the PLC actuators which control the water level of tanks could lead to flooding or drought. Therefore, adopting state-of-the-art technologies in cyber security has been receiving lots of attention in the literature of WDS. After the integration of IT and OT, WDS will have a natural operation as cyber-physical systems; thus, the security measures should not only cover the traditional IT security levels, such as a high encryption level, good authentication, authorisation, and implementing whitelists and blacklists, but should also incorporate intelligent systems in the operational level that could verify the genuinity of the transferred data to avoid Man-in-the-Middle (MiTM) attacks, as mentioned in [3].

Recent research and development of blockchain have shown that blockchain could be the best solution to strengthen the next generation of WDS with redundancy and immutability of the stored data according to [4]. Moreover, it verifies the data to assure that the initiated data have not been compromised during transmission. Apart from the security robustness of blockchain, blockchain can also enhance the automation in WDS, facilitate peer-trading systems to minimise water losses, and offers fair water rights among customers, as it connects several subsystems into one automated system without intermediaries.

Blockchain is a digital structure of the transferred data in a decentralised network that aims to share information securely. The data are aggregated and timestamped in the form of linked blocks forming in a chain, called ledgers [5]. These ledgers are digital transactions, data logs, and executions by smart contracts, blockchain algorithms, with consensus mechanisms as discussed in [6]. The blockchain algorithm is a digital self-executing agreement between network peers (e.g., sensors); it is meant to promote the rules of transactions. A smart contract is a virtual machine on top of the blockchain algorithm doing further functions [7]. The ethereum virtual machine (EVM) is considered as the most popular smart contract with the infrastructure and industrial Internet-of-Things (IIoT) systems. The consensus mechanism is a dynamic approach to achieve the necessary agreement through validation using the hash function. This necessary agreement could include one or multiple nodes during the validation, and the selection of these nodes should also be considered in the algorithm. The hash function is a one-way function that maps data of arbitrary size to fixed-size values, and it is also infeasible to invert or reverse the computation.

For simplicity, the blockchain is a data structure that has two key features of interconnection and distributed nodes. Any change in the data are counted as a new block of data structure. Hence, the existing data block is untampered. Moreover, the data are stored distributively among the blockchain nodes. The way to generate new blocks is by using consensus algorithms. Inside the consensus algorithms, it provides mechanisms to add a block to the blockchain which was agreed by all the distributed nodes. For instance, a blockchain node initiates a transaction that is broadcasted to other blockchain nodes for validation based on the consensus mechanism (see Figure 1). It is proposed to replace the trusted third parties' (TTP) roles by enforcing the roles in the blockchain algorithm, which results in removing intermediaries. Therefore, the blockchain system is tamper-resistant and self-enforceable, as it is installed on the selected peers with the required computational capacity. TTP is an entity that facilitates interactions between two parties who both trust the third party.

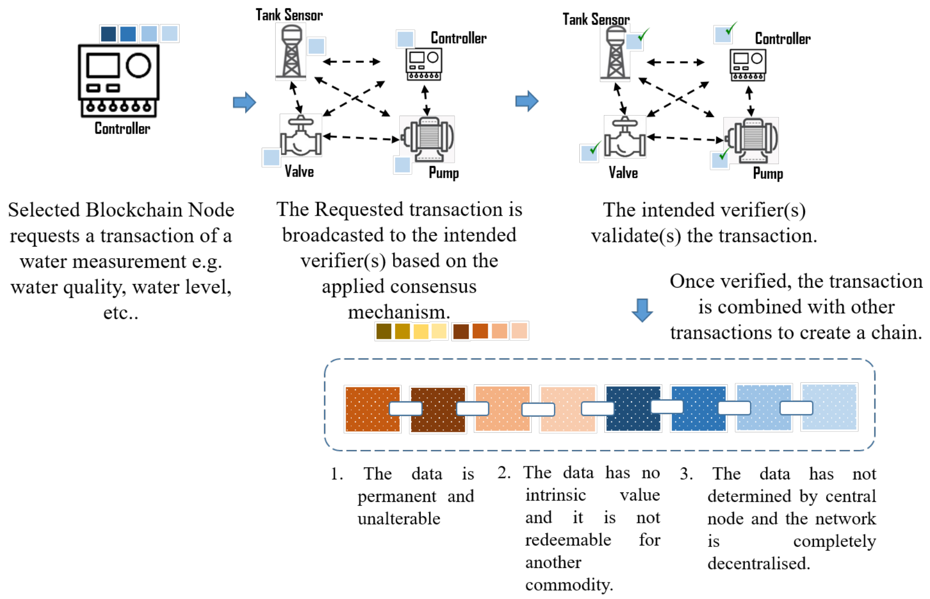
Controller
Tank Sensor
Controller
Valve
Pump
Tank Sensor
Controller
Valve
Pump
Selected Blockchain Node requests a transaction of a water measurement e.g. water quality, water level, etc..
The Requested transaction is broadcasted to the intended verifier(s) based on the applied consensus mechanism.
The intended verifier(s) validate(s) the transaction.
Once verified, the transaction is combined with other transactions to create a chain.
1. The data is permanent and unalterable
2. The data has no intrinsic value and it is not redeemable for another commodity.
3. The data has not determined by central node and the network is completely decentralised.

**Figure 1.** Blockchain diagram.

Applying blockchain to WDS provides secure communication and data transfer within the network. If the meta-data have to be transmitted to the Cloud, then the water systems can be vulnerable to eavesdropping. Hence, another hash function can be implemented on the Cloud to overcome this risk to ensure that the data have not been compromised by comparing the hash and real data according to [8]. Simulating the water system operations with the WDSchain can show the benefits of integration with blockchain with the following features:

- Availability of service: it is crucial to keep the availability of the OT (physical systems) and to avoid any disruption of the service that can endanger the society and health of human beings. The transferred data are verified by other network peers in a decentralised manner to maintain the continuity of the service and mitigate the Denial of Service (DoS) [9].
- Data Immutability: As in the initial step of integrating blockchain, the transferred data have to be stored in the Cloud. Hence, a hash function can be utilised in the verification process in both blockchain and Cloud to compare the chained data with the meta-data. In this approach, the transferred data cannot be manipulated during the transmission as it is validated again at the Cloud. When all the smart networks get matured enough to handle autonomous decision-making, the data may not be required to be transmitted to the Cloud. Then, the WDS will have the full benefit of data integrity, confidentiality, and immutability.
- Data and Process Transparency: The process does not require a centralised network for verifying the data, and the fact that the network nodes participate in the validation process provides transparency of the whole process in general. However, the centralisation level may change based on the consensus mechanism, but the baseline of the decentralisation network provides a sense of transparency.

In contrast, there are a few limitations of blockchain technology that can be summarised as high computations, scalability, and cross-chain compatibility. The most popular blockchain mechanism known as PoW has high computations, yet other mechanisms can have reasonable computations. Moreover, scalability and cross-chain compatibility refer to the limited capability to handle a large amount of transaction data and to expand the system to have multiple integrated blockchains. These issues restrict the development of blockchain and its integration with industries such as water. In favour of this, and as it is still early to rely on blockchain, this paper revealed the issues that could happen upon integration of blockchain with water systems, and it proposes a WDSchain open-source toolbox to be considered by researchers, professionals, and operators to simulate their system for further improvements in the data structure, consensus mechanism, and the conversion of water transactions into the digital structure.

With the current development of the industrial consensus mechanisms, having open-source software that can be the initial milestone for developers for further development is paramount. Therein, this paper introduces WDSchain, an open-source MATLAB toolbox. It extends the features of the EPANET, which allow the water operators or researchers to simulate blockchain on their WDS using two modes: (1) static blockchain, which takes a snapshot of one-time interval data of all nodes in WDS as input and output into chained blocks at a time, and (2) dynamic blockchain, which takes all simulated time-series data of all the nodes as input and establishes chained blocks at the simulated time. Moreover, the developers are not only limited to simulate their network using WDSchain, but can further develop their consensus mechanism in the blockchain and desired data-mapping techniques in the toolbox. Five consensus mechanisms are developed in the WDSchain toolbox and provide data with different security levels for ease of integration with traditional WDS as needed, subject to asset capabilities. In this paper, the evaluation is realised with five performance metrics, including latency, throughput, operations per transaction ($OpT$), the mining metric, and complexity metric.

This paper is organised as follows: Section 2 reviews the related works of blockchain applications in water systems. Section 3 proposes the WDSchain toolbox by introducing the system model and running configuration. Section 4 discusses the case studies of both blockchain modes. Section 5 presents and evaluates the results. Section 6 concludes the work and discusses future work.

## 2. Related Works

Blockchain technology is spreading widely to be engaged in many applications and industries, such as billing, logistics, transportation, healthcare, agriculture, and energy systems, as mentioned in [6,10–13]. The key use case of these applications is to promote the supply chain and automation of the intended industries. In terms of water systems, several articles and initiatives are discussing the potential use-cases of blockchain in water systems, which are water trading, water management, fundraising, water quality, wastewater trading, and wastewater management [14–17]. These works utilise blockchain in promoting automation of water-quality monitoring in rivers, and dam surveillance [14,15]. Predescu et al. [16] utilises the crowd-sensing in validating water in a gaming approach, while Pee et al. [17] proposes a simple water-trading system. These initiatives aim to ease the integration of IT and OT systems occurring in a secure way according to [14–17]. water-trading systems that use blockchain aim to publicise water consumption and create a marketplace for trading water assets between consumers. This approach will encourage the use of alternative water resource channels for collecting water (e.g., rainwater collection systems, improved shallow groundwater resources systems), utilise unused water sources, reduce water leakages, and minimise operating costs, according to [18,19]. Pincheira et al. [18] implemented a simple blockchain that considers water-trading systems on the Ethereum platform and Arduino boards without using a consensus mechanism. Kassou et al. [19] conceptualised a waste-water management system that can trade in the untreated water for irrigation.

Blockchain technology contributes in facilitating the communication of untrusted peers without any centralisation intervention. Similarly to water management and trading systems that can automate the daily routine operations without a centralised system, wastewater systems have the same use-cases, ensuring fair-trading of untreated water and providing a smooth and secure monitoring process of water quality measurements, as in [20]. The development of these applications in the water sector is still in the conceptual phase, and only a few of them managed to make initial prototypes.

Water market and management systems are the two promising blockchain use-cases, and both can have a significant impact on water scarcity, fairness, and system security.

Water Market is a system that deals with the water assets as a tradable commodity to enable the buying and selling of water without centralisation intervention. Fairness of the water distribution based on the demands is another feature discussed in [21]. Water management systems enable the identification of the water bursts and promote the efficiency of the operational and informational technologies. These two use-cases can be implemented to provide secure communication and data transfer using blockchain technologies and other typical security measures (e.g., two-factor authentication, and white and blacklisting). In our previous work, we proposed a data aggregation system with the blockchain and utilised the hash function and bloom filter to preserve the privacy of the transferred data by representing network peers in pseudonyms, as in [22].

To the best of our knowledge, no toolbox in the literature has yet explored the feasibility of blockchain in WDS, and this motivated us to extend our work to implement an open-source MATLAB toolbox of blockchain to provide insights for operators on the benefits of implementing blockchain in WDS. Moreover, this work provides a toolbox for developers to implement further consensus mechanisms, tailored blockchain algorithms and smart contracts, and provide different data-mapping approaches in WDS.

Several EPANET wrappers have been developed in the literature to extend functions in the WDS to ease the interaction with the EPANET environment. MATLAB, R, and Python have successfully been used to import and simulate data from EPANET to their programming environments, as in [23–29].

In the R language, the Refs. [23,24] developed classes called the 'EpanetReader' and the 'epanet2toolkit' that can import, read, simulate, and plot the data. The 'epanet2toolkit' tool supports an extended period of simulations of water quality and hydraulic operations in illustrative applications. Likewise, the 'EPANET-MATLAB toolkit' offers similar functions except for simulating water quality and hydraulic processes in the MATLAB environment, as in [24,30]. The 'EPANET-Matlab toolkit' is the enhanced version of the 'getwdsdata' function in [25] that can import static WDS for all nodes and links. Additionally, the 'EPANET-MATLAB toolkit' was developed in 2016 to add a broader range of simulation functions, such as simulating water quality and other hydraulic processes, as developed by [24]. Finally, the 'EPANETCPA' toolbox was developed for assessing cyber-physical (CP) attacks and their consequences on WDS by [26].

Python has also been embedded in some developments, extending EPANET functions on sensor placements to mitigate contamination events and provide resilience for the water systems as the Threat Ensemble Vulnerability Assessment and Sensor Placement Optimization Tool (TEVA-spot) by [27], Chama by [28], and others based on the Evolutionary algorithm by [29]. These mentioned tools are the improved version of the previous versions of the Water Security Toolkit (WST) and Water Network Tool for Resilience (WNTR) in simulating hydraulic and water-quality models along with analysing the resilience of WDS under natural disasters, as mentioned in [31–33].

However, systems without consensus mechanisms are more computationally friendly, yet the consensus mechanism causes the system to be decentralised and robust against biased verifiers (centralisation systems). Moreover, water systems usually have few network components compared to other fields in which the consensus mechanism will not require a lot of computation capacity. The various consensus mechanisms emphasise supporting real-time data and easing the integration of current physical assets. Proof-of-Trust (PoT) is the most-used consensus mechanism in the literature for IIoT applications [34]. Other references in the literature use the terminology of Proof-of-Authenticity or Proof-of-Authority instead of PoT, and both techniques rely on the trust/reputation of one of the selected blockchain nodes based on prior verification processes. Proof-of-Vote (PoV) is another promising mechanism that relies on the voting of the selected nodes for IIoT applications.

## 3. WDSchain

### *3.1. Proposed Architecture*

The proposed architecture consisted of three key components: importing WDS data into the MATLAB environment, blockchain algorithm and data verification, and the data-mapping approach (see Figure 2). After the data are imported to the MATLAB environment, the consensus mechanism and data-mapping technique are selected by the user with an assumption of the blockchain nodes. The hashed data are transmitted to the database, in which other applications on the server can be used on the data. The key components of the proposed model are described as the following:

- Importing data into the MATLAB environment: The hydraulic data can be imported either from the CSV file (in dynamic blockchain mode) or from the EPANET modeling file (in static and dynamic blockchain modes) into the MATLAB environment using the EPANET-MATLAB toolkit or EPANETCPA for any one-time interval or time-series data, respectively. The data are imported into MATLAB as tables that represent network configurations (e.g., network nodes, type, their coordinates, and the number of edges). Then, the data are saved in the form of blocks and its self-hash value is generated upon data verification. Only the approved blocks are added to the chain along with other blockchain information, such as node ID, self-hash value, and nonce. A nonce is an abbreviation for "Number Only Used Once" that is used during the hashing of the data.
- Blockchain and data-verification algorithms: These two processes are the key parts of any blockchain system. The blockchain algorithm oversees the communication and verification processes based on the selected data-verification algorithm (known as the consensus mechanism). In detail, the blockchain algorithm in this work aims to re-hash the data with the cooperation of the developed consensus mechanisms (see Algorithm 1). In brief, it concatenates the water coefficients to be validated based on the consensus mechanism, as well as the previous hash if previously linked with the block, except for the Genuis block. A consensus mechanism is an algorithm that has a set of rules that all the peers should follow based on the blockchain algorithm and enforce the verification of the block and its transaction. Five consensus mechanisms are developed in this toolbox, which are: Proof-of-Work (PoW), PoT, Proof-of-Assignment (PoA), Proof-of-Vote (PoV), and Proof-of-Authentication (PoAuth). PoA and PoAuth are proposed in the [35] project because the phenomenon of the WDS requires fast-action, voting-based, and real-time verification processes. IoTw is one of the leading blockchain projects that develop blockchain for industries and manufacturers, and it has the highest level of security as well as flexibility of communication with smart devices [35].

A brief comparison of Iotw with other leading industrial blockchain projects (e.g., IoTa and IoTex) can be found in [35]. Thus, PoA, PoV, and PoAuth are developed to attain adequate integration with the water industry. PoW is the first consensus mechanism and has been used in bitcoins. It is a mechanism that asks everyone to verify the transaction, and the data are accepted when one of the validators approves the transaction (see Algorithm 2). The PoT mechanism asks the peers that have the highest reputation to verify the data (see Algorithm 3). It has a reputation table with all blockchain nodes in which every node gets a reputation increment upon data approval. The $Peer_{Choosen}$ is identified based on the node that has the highest reputation, and the reputation table has a maximum threshold of $Reputation_{Threshold}$ to avoid overflow of the reputation that can be changed based on the network size. In general, PoT has successfully improved energy efficiency, but it also conveys the network in a semi-centralised manner. Moreover, PoV is a recently developed algorithm based on the voting system of the hashing function as mentioned in [36]. It is an algorithm that asks all the blockchain peers to verify the data and whether the summation of the feedback has more than 3/4 validity of the data. Then, the block and its transaction will be approved and join the chain (see Algorithm 4).

Alternatively, PoA and PoAuth are two new concepts in the literature to provide a quick and low-processing verification process. PoA is an algorithm where a randomly selected network node is asked to verify the data (see Algorithm 5). It is essential to have a quick-mining mechanism to support real-time data in addition to the simplicity of the verification process, but there is a good chance that the selected random verifier will be malicious. PoAuth of the network nodes is an algorithm that uses self-authentication as proof of the transmitted data. This mechanism is adequate for IoT systems that do not require a high security level. It uses a bloom filter to match the authentication of the nodes, and it can be implemented right away on the traditional assets (see Algorithm 6).

- Data-mapping: There are two modes of data-mapping in joining the chain upon verification of data blocks. In dynamic blockchain, either the data (transaction) are chained per timestamp, or sensors. When all sensor data in the one-time interval are kept in one block, this is referred to as the transaction per timestamp. In contrast, it is the transaction per sensor when all data of one sensor are chained in one block or transaction.

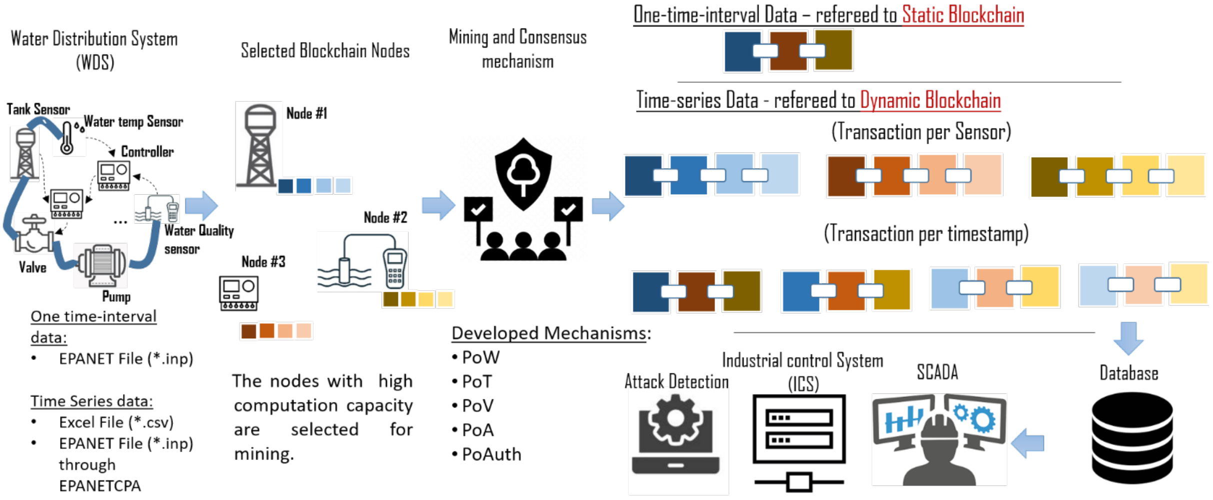
Water Distribution System (WDS)
Selected Blockchain Nodes
Mining and Consensus mechanism
One-time-interval Data – refereed to Static Blockchain
Time-series Data - refereed to Dynamic Blockchain
(Transaction per Sensor)
(Transaction per timestamp)
Tank Sensor
Water temp Sensor
Controller
...
Water Quality sensor
Valve
Pump
Node #1
Node #2
Node #3
One time-interval data:
• EPANET File (*.inp)
Time Series data:
• Excel File (*.csv)
• EPANET File (*.inp) through EPANETCPA
The nodes with high computation capacity are selected for mining.
Developed Mechanisms:
• PoW
• PoT
• PoV
• PoA
• PoAuth
Attack Detection
Industrial control System (ICS)
SCADA
Database

**Figure 2.** WDSchain System architecture.

**Algorithm 1** Blockchain algorithm

```
while blockIndex = (1 to NoOfBlocks) do
    initialblockData = blockArray(blockIndex);
    str = concatenate(num2str(blockIndex), initialblockData);
    selfHash = DataHash(str);
    ConsensusMech = Input('Enter Consensus Mechanism');
    Genuine = Mine(initialblockData, ConsensusMech);
    if Genuine == True then
        if blockIndex == 1 then
            'Generate Genius block' ;
            PreviousHash = null ;
        else
            PreviousHash = selfHash(blockIndex-1);
        end
    else
        Disp('The block has not been approved');
    end
end
```

**Algorithm 2** VerifyNewblock using PoW

```
Open multi-core session;
while (i = 1 to No.of peers in different system core) do
    $newHash$ = DataHash($newblock$, $nonce$);
    $newBlock.selfHash = newHash$ ;
    if newBlock.selfHash == block.selfHash then
        Genuine = True;
    else
        Genuine = False;
    end
    Calculate the mining time;
end
```

**Algorithm 3** VerifyNewblock using PoT

```
$Peer_{choosen}$ = Find($Peer$, $'MaxReputation'$) ;
$newHash$ = DataHash($newblock$);
$newBlock.selfHash = newHash$ ;
if newBlock.selfHash == block.selfHash then
    Genuine = True;
else
    Genuine = False;
end
while ($Peer_{choosen} \leq Reputation_{Threshold}$) do
    if Genuine == True then
        $Peer_{choosen}$=$Peer_{choosen}$ + 1;
    else
    end
end
```

**Algorithm 4** VerifyNewblock using PoV

```
while (count 1 to Networksize) do
    $newHash$ = DataHash($newblock$) ;
    $newBlock.selfHash = newHash$ ;
    if newBlock.selfHash == block.selfHash then
        $voting = voting + 1$;
    else
    end
end

if (res/Networksize) >= 0.6 then
    Genuine = True;
else
    Genuine = False;
end
```

**Algorithm 5** VerifyNewblock using PoA

```
$Peer_{choosen}$ = Find($Peer$, $'Random'$) ;
$newHash$ = DataHash($newblock$);
$newBlock.selfHash = newHash$ ;
if newBlock.selfHash == block.selfHash then
    Genuine = True;
else
    Genuine = False;
end
while ($Peer_{choosen} \leq Reputation_{Threshold}$) do
    if Genuine == True then
        $Peer_{choosen}$ = $Peer_{choosen}$ + 1;
    else

    end
end
```

**Algorithm 6** VerifyNewblock using PoAuth

```
Authent = input('Enter Authentication ID = ');
while (Counter = 1→ Networksize) do
    if Authent == Node.ID(Counter) then
        Genuine = True;
        break;
    else

    end
end
while ($Peer_{choosen} \leq Reputation_{Threshold}$) do
    if Genuine == True then
        $Peer_{choosen}$ = $Peer_{choosen}$ + 1;
    else

    end
end
```

## *3.2. Configuring and Running a Simulation*

### 3.2.1. The Input File

The simulation can be started from the 'WDSchain.m' file. The users shall be asked whether they would like to use static or dynamic blockchain. The static blockchain aims to import a one-time interval of a WDS modelling file into a blockchain where one-sensor measurements (all-time interval readings) are in one block, while the dynamic blockchain offers two modes of importing time-series data either from CSV files or from EPANET through EPANETCPA. To import time-series data from EPANET in the dynamic blockchain mode, a data-flow file has to be provided to the EPANETCPA for simulation.

Several water parameters (e.g., the initial level of water at a tank, or water quality of a certain node) can be selected to be validated and chained upon verification (see Table 1). Moreover, the users are asked to select one of the consensus mechanisms to be used. If the PoAuth method is chosen, then each block has to be authenticated by matching the entered ID with the list of nodes in the software. Furthermore, the data-mapping approach can be selected of either transaction per timestamp or sensor.

**Table 1.** Available data that can be used in the blockchain.

| Data | Variable Name |
|---|---|
| **Nodes Information** | |
| The value of all node emitter coefficients | d.NodeEmitterCoeff |
| The indices of all nodes | d.NodeIndex |
| The value of all node initial quality | d.NodeInitialQuality |
| The ID label of all nodes | d.NodeNameID |
| The demand categories | d.NodeDemandCategoriesNumber |
| The value of all node pattern indices | d.NodePatternIndex |
| The Computed values of all node pressures | getNodePressure |
| The value of the nodes source quality | d.NodeSourceQuality |
| The tank bulk rate coefficient | d.NodeTankBulkReactionCoeff |
| The node code-index for all nodes | d.NodeTypeIndex |
| **Pipelines Information** | |
| Retrieves the indices of the pipelines | d.getLinkPipeIndex |
| Retrieves all the value of the pipeline lengths | d.LinkLength |
| Retrieves all the value of the pipeline diameter | d. LinkDiameter |
| Bulk chemical reaction coefficient | d. LinkBulkReactionCoeff |

### 3.2.2. The Output File

The complete process of the blockchain is saved in a log file named 'Blockchain_log.txt', and the data are stored in a decentralised network depending on the data-mapping technique. The stored data can be viewed later in the 'Storage' folder. Throughout the processing of the system, the generation of the chain can also be observed in the command window. The block index, data, previous hash, and self-hash can be observed in the command window and storage files. The TankID, TankInitial level, and TankMaximum water level are the considered coefficients to be chained, as we only considered the water tanks as the selected blockchain nodes in our case studies. On the other hand, other coefficients can be considered further depending on the phenomenon of the water system (see Table 1). Moreover, the WDS are plotted and the mining time is calculated automatically upon starting the simulation of the WDSchain.

## 4. Case Studies and Methodologies for Evaluation

As mentioned previously in the related works of blockchain in the literature, there are two discussed crucial use-cases in water systems, which are water-trading and water management. The water management systems can be divided into three sub-systems: water treatment, and water distribution at the supply and demand sides. This work considered the water distributed at the supply side. The water-trading system and the other subsystems of water management use-cases are not considered in this study. The water distribution at the supply side involves the water flow from the treated water and reservoirs, through water tanks (with level sensors), valves, pumps, and pipelines. The hydraulic data in the water distribution at the supply side can be extracted from EPANET software modeling. Four case studies are considered in evaluating the WDSchain, and two modes of static and dynamic blockchains. The case studies can be bundled in the toolbox folder under the 'networks' folder, and they are briefly described (see Table 2). All case studies are simulated on the following specifications i5-6200U, CPU 2.4 GHz, and 8192 MB RAM. The WDSchain toolbox realises a top-view simulation and the network peers participate in the verification process. Moreover, the toolbox does not simulate the communication between the network peers. Hence, the data-broadcasting time is neglected.

**Table 2.** Case-study specifications.

| # | WDS | Description | Specifications |
|---|---|---|---|
| 1 | D-Town | A residential district in the Eastern part of Exeter city. | 407 nodes, 443 pipelines, 11 pumps, 7 tanks, and 1 reservoir. |
| 2 | C-Town | A residential district in the Eastern part of Exeter city. | 396 nodes, 429 pipelines, 11 pumps, 7 tanks, and 1 reservoir. |
| 3 | Net3 | EPANET Example. | 97 nodes, 119 pipelines, 2 pumps, 3 tanks, and 2 reservoirs. |
| 4 | Richmond | A residential district town in the UK. | 872 nodes, 957 pipelines, 7 pumps, 6 tanks, 1 valve, and 1 reservoir. |
| 5 | BWSN | A real WDS are "twisted" to preserve their anonymity. | 129 nodes, 169 pipelines, 2 pumps, 2 tanks, 46 valves, and 1 reservoir. |

Three performance metrics were studied to evaluate system complexity, namely: latency, number-of-operations-per-transaction ($OpT$), and throughput. Latency is the time required for a transaction to be confirmed and become irreversible. It relies on two coefficients: the time taken for generating a data block $t^G$, and time taken for verification ($t^{(v,\delta)}$) in which $\delta$ defines the used consensus mechanism ($\delta \ \epsilon \ [0,1,2,3,4]$ for the four consensus mechanisms). It is denoted by:

$$t^L = t^G + t^{v,\delta}. \tag{1}$$

$OpT$ is the required number of operations in the algorithm for data verification. This metric contributes to identifying the complexity of the consensus mechanism. Throughput ($s^T$) is the amount of transactions in a second. It relies on two parameters: the number of transactions ($N^T$), and the latency ($t^L$). It is denoted by:

$$S^T = \frac{N^T}{t^L}. \tag{2}$$

In terms of security, most consensus mechanisms (e.g., PoW, and PoT) offer a high probability of security, as the network could be vulnerable if (>51%) of the mining power is biased. Moreover, for the PoV mechanism, unambiguous biasing could happen if (>33.3%) of the mining power is biased. Therefore, to make sure that the network is safe with any consensus mechanism ($\delta$) against the number of malicious verifiers ($v^M$), this can be achieved as follows:

$$v^M \leq v^{T,\delta} \tag{3}$$

$$v^{T,\delta} = \left[\frac{N-1}{3}\right], \tag{4}$$

where $v^{T,\delta}$ is the number of true verifiers for certain consensus mechanisms ($\delta$) and $N$ is the total number of peers or verifiers. For the PoA mechanism, there is a probability of ($1/N$) to be a malicious verifier, yet it is reasonable to develop this mechanism as it has the lowest mining needs.

## 5. Results and Discussion

Due to the high computational capacity needed, the toolbox relies on the tanks as selected blockchain nodes for transmitting water measurement data and verifying the data (see Figure 3). Some of the components in the WDS may fail to provide sufficient computational capability to do the processing (e.g., junctions, pumps, and valves). The simulation shows that with an increasing number of blocks/transactions, $t^{(v,\delta)}$ ranges from 0.4272 to 0.8707 s for chaining seven blocks/transactions as in D-Town and C-Town (see Table 3). PoA has the lowest mining time, as only one random node of the selected blockchain nodes is assigned for the verification with 0.46 and 0.45 s for D-town and C-Town, respectively. Moreover, PoW has the highest mining time, as all the blockchain nodes are assigned for data verification in parallel using parallel computation which takes the longest time among the others using normal processing (and not a GPU server) with 0.8707 and 0.6919 s. However, PoV assigns all blockchain nodes to verify the data as PoW, but it takes a shorter mining time since the current processor can support only two parallel users at a time.

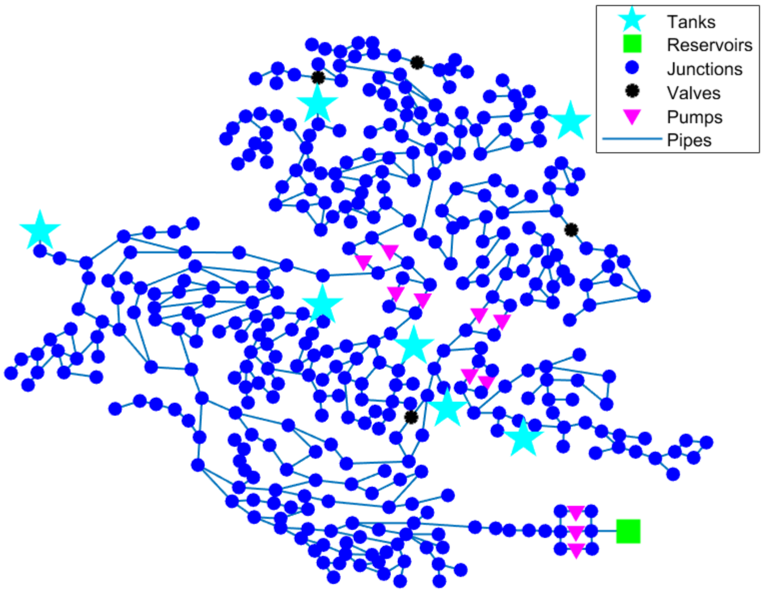

**Figure 3.** C-Town Distribution Model as an example of one of the case studies.

**Table 3.** The calculated performance for static blockchain.

| WDS | Consensus Mechanism | $N^T$ (s) | $t^G$ (s) | $t^{(v,\delta)}$ (s) | $t^L$ (TPS) | $S^T$ | $OpT$ |
|---|---|---|---|---|---|---|---|
| D-Town | PoW | 7 | 0.07 | 0.8707 | 0.9407 | 7.44 | 24 |
| | PoT | | | 0.4634 | 0.5334 | 13.1 | 5 |
| | PoA | | | 0.4372 | 0.5072 | 13.8 | 5 |
| | PoV | | | 0.4447 | 0.5147 | 13.6 | 23 |
| | PoAuth | | | 3.35 + Auth | 3.42 + Auth | - | 9 |
| C-Town | PoW | 7 | 0.07 | 0.6919 | 0.76 | 9.18 | 24 |
| | PoT | | | 0.4500 | 0.52 | 13.4 | 5 |
| | PoA | | | 0.4380 | 0.508 | 13.7 | 5 |
| | PoV | | | 0.4272 | 0.49 | 14.2 | 23 |
| | PoAuth | | | 3.35 + Auth | 3.42 + Auth | - | 9 |
| Net3 | PoW | 3 | 0.03 | 0.2061 | 0.2361 | 12.7 | 12 |
| | PoT | | | 0.1176 | 0.1476 | 20.3 | 5 |
| | PoA | | | 0.1321 | 0.1621 | 18.5 | 5 |
| | PoV | | | 0.0988 | 0.1288 | 23.2 | 11 |
| | PoAuth | | | 0.15 + Auth | 0.18 + Auth | - | 5 |
| Richmond | PoW | 6 | 0.06 | 1.3622 | 1.4222 | 4.21 | 21 |
| | PoT | | | 0.6168 | 0.6768 | 8.86 | 5 |
| | PoA | | | 0.6606 | 0.7206 | 8.32 | 5 |
| | PoV | | | 0.5611 | 0.6211 | 9.66 | 20 |
| | PoAuth | | | 0.3 + Auth | 0.36 + Auth | - | 8 |
| BWSN | PoW | 2 | 0.02 | 0.5741 | 0.5941 | 3.36 | 9 |
| | PoT | | | 0.1774 | 0.1974 | 10.1 | 5 |
| | PoA | | | 0.1810 | 0.201 | 9.95 | 5 |
| | PoV | | | 0.1688 | 0.1888 | 10.5 | 8 |
| | PoAuth | | | 0.1 + Auth | 0.12 + Auth | - | 4 |

Other WDS case studies show monotonic behavior. It is also observed that the mining of one transaction varies between 0.1–0.3 s depending on the consensus mechanism. For PoAuth mechanism, the time taken by the devices to self-authenticate is named ‘Auth’. This factor changes with the speed of the network, network size, type of authentication, communication protocol, Cloud-processing speed, and other factors that affect the processing time. PoT and PoA do not rely on the number of the transaction/blocks, and they have the minimum $OpT$. In contrast, PoW and PoV are heavily dependent on the number of blocks, and they have a significantly higher $OpT$. Moreover, the PoAuth mechanism is slightly affected as the self-authentication process tends to match the sensor name with the others on the network.

In the dynamic blockchain, two options of data-mapping are provided as discussed: (a) transaction per timestamp where all measurements (of all sensors) at one time are chained in one block, and (b) transaction per sensor, where one-time data measurements of one sensor are chained in one-block. Thus, the number of blocks in the first type is similar to the number of timestamps, while the second one is the number of selected blockchain nodes multiplied by timestamp interval. Therefore, the number of blocks significantly increases in the transaction per sensor which affects the mining time of the consensus mechanisms. The transaction per timestamp is a very common type in cryptocurrency platforms. Furthermore, it is adequate for real-time data water systems to perform daily operations and actions. On the other hand, the transaction per sensor can be implemented for other blockchain applications, such as sharing the log files, policy confirmation, assets maintenance history, and so forth. This is because a decision at that moment may require the history of all the previous measurements of that device. The simulation shows that with the increasing number of blocks/transactions, $t^{(v,\delta)}$ ranges from 116.4 to 872.6 s for chaining 953 blocks/transactions as in the C-Town distribution of 953 time-interval data (see Table 4). When the data-mapping mode changes to transaction per sensor in which each sensor data at one time is chained in a single block, the number of transactions is significantly increased by the factor of the number of selected blockchain nodes. A total of 6672 transactions were chained in 12,355, 877.6, 861, 338, and 400+ Auth seconds using PoW, PoT, PoA, PoV, and PoAuth, respectively. The throughput ($S^T$) significantly decreased for the PoW and PoV due to the massive computation.

Applying blockchain systems at the water-tank level mitigates manipulation of the sensing and control signal that can affect the decision of opening valves or pumps that could finally lead to flooding or drought. Some works in the literature did not realise consensus mechanisms in their network in which an authorisation point is in charge of approving the transferred data.

**Table 4.** The calculated performance for dynamic blockchain.

| WDS | Data | Consensus | $N^T$ | $t^G$ | $t^{(v,\delta)}$ | $t^L$ | $S^T$ | $OpT$ |
|---|---|---|---|---|---|---|---|---|
| | Mapping | Mechanism | (s) | (s) | (s) | (TPS) | | |
| D-Town | Per timestamp | PoW | 381 | 3.81 | 277.87 | 281.68 | 1.352 | 1143 |
| | | PoT | | | 54.6 | 58.41 | 6.522 | 5 |
| | | PoA | | | 59.7 | 63.51 | 6.00 | 5 |
| | | PoV | | | 111.8 | 115.61 | 3.29 | 1146 |
| | | PoAuth | | | 19.2 + Auth | 23.01 + Auth | - | 383 |
| | Per sensor | PoW | 2668 | 26.6 | 4079.9 | 4106.5 | 0.649 | 8004 |
| | | PoT | | | 260.71 | 287.31 | 9.286 | 5 |
| | | PoA | | | 265.09 | 291.69 | 9.146 | 5 |
| | | PoV | | | 640.14 | 666.74 | 4.001 | 8007 |
| | | PoAuth | | | 133.4 + Auth | 160 + Auth | - | 2670 |
| C-Town | Per timestamp | PoW | 953 | 9.53 | 872.6 | 882.13 | 1.080 | 2859 |
| | | PoT | | | 109.2 | 118.73 | 8.02 | 5 |
| | | PoA | | | 116.4 | 125.93 | 7.567 | 5 |
| | | PoV | | | 491.3 | 500.83 | 1.902 | 2861 |
| | | PoAuth | | | 48.2 + Auth | 489.53 + Auth | - | 955 |
| | Per sensor | PoW | 6672 | 66.7 | 12,288.8 | 12,355.5 | 0.54 | 20,016 |
| | | PoT | | | 810.9 | 877.6 | 7.602 | 5 |
| | | PoA | | | 794.4 | 861.1 | 7.748 | 5 |
| | | PoV | | | 3314.5 | 3381.2 | 1.97 | 20,019 |
| | | PoAuth | | | 333.6 + Auth | 400.3 + Auth | - | 6674 |

Water systems require two key aspects in terms of data validation, namely, high security and quick verification. Consequently, PoT and PoV are the two suggested consensus mechanisms to maintain the requirements. PoV has better security measures than PoT, but it is very energy-consuming and has higher latency, since all the selected blockchain nodes have to process the data. Since the selected blockchain nodes in the water systems are limited (in tens), the processing can still support real-time processing (see Table 3). PoT can be deployed as well to relax the consumption if the network does not require extreme security measures.

A comparison of the developed consensus mechanism is applied from mining, complexity, and throughput metrics (see Table 5). A mining metric is an average number of mining times concerning the chained transactions. It is denoted by $Mine^{Ave}$ (see Equation (5)). A complexity metric is an average number of operations conducted concerning the chained transactions, and it is denoted by $Compl^{Ave}$ (see Equation (6)). PoAuth has the highest throughput, but also the Internet speed, and self-authentication timing affects its performance. In general, PoW and PoV have the highest security, latency, energy consumption, and the least throughput. On the other hand, PoAuth, PoA, and PoT have the least security, latency, energy consumption, and highest throughput, respectively.

$$Mine^{Ave} = \frac{t^{v,\delta}}{N^T} \tag{5}$$

$$Compl^{Ave} = \frac{OpT}{N^T} \tag{6}$$

The current version of this toolbox has not considered these following points: First, the communications protocols (e.g., the remote procedure, called 'RPC') of the transmitted data are not considered in this toolbox. Second, the EPANET does not simulate the controller (i.e., PLCs) data, and consequently, our toolbox cannot chain the control data. Third, the broadcasting delay and processing of the data transmission are not counted. Fourth, the selected blockchain nodes are assumed to have direct communication between them. In the PoW mechanism, it relies on parallel computing in MATLAB where the license and computer specifications affect the mining time. If the WDSchain system operated on a GPU, the performance including the mining time of the PoW will be significantly enhanced. Moreover, the used core processors are not mapped with the network peers. In the PoT mechanism, the reputation matrix is randomly initiated, and then the verifier peer gets an increment of their reputation stake.

**Table 5.** Comparisons of the developed consensus mechanisms.

| | Mining Metric (s) | Complexity Metric (TPS) | Throughput Metric | Cons | No. of Verifiers | Verification Based on |
|---|---|---|---|---|---|---|
| PoW | 0.1–0.3 | 3–20 | 0.73–6.84 | Extremely High Energy consumption | All peers | The Computational capacity |
| PoT | 0.09–0.19 | 0.005–0.5 | 0.9–9.7 | Initially random generation of reputation matrix | One peer | Peers Reputation |
| PoA | 0.09–0.17 | 0.09–0.68 | 5.48–10 | A random selection of a verifier | One peer | Random-selection |
| PoV | 0.09–0.68 | 0.034–0.065 | 1.4–18.3 | High Energy consuming | All nodes | Vote of peers |
| PoAuth | 0.03–0.06 + Auth | - | - | Assumes all peers are genuine | One peer | Authentication |

## 6. Conclusions and Future Work

This work introduced an open-source MATLAB toolbox called WDSchain. It is a blockchain simulation toolbox that facilitates the simulation of blockchain on WDS for water operators, professionals, and researchers, which can bring security advantages of blockchain into the water systems. The toolbox offers two simulation modes: (1) static blockchain, which takes a snapshot of one-time interval data of all nodes in WDS as input and output into chained blocks at a time, and (2) dynamic blockchain, which takes all simulated time-series data of all the nodes as input and establishes chained blocks at the simulated time. The toolbox helps us to evaluate the most suitable consensus mechanism in the WDS. Five consensus mechanisms were developed to attain different security levels and ease of integration with current water assets according to the computational capacity and the required security. If the water operators can tolerate the security of a water system (the shared data cannot endanger the utility's premises or disrupt the service), PoA and PoAuth are more likely to be used. Moreover, they can work on the current devices' 'water assets' as they do not require a lot of computations. In contrast, PoW, PoV, and PoT can be applied for powerful devices and in environments where a high level of security is needed. Five performance metrics were designed to evaluate the consensus mechanisms using this toolbox.

WDSchain provides insights into the best-suited consensus mechanism based on the five performance evaluation metrics and the required security and physical level of the system. In addition, the water operators, professionals, and researchers can further develop their own consensus mechanism, data-mapping, or consider further functions in the blockchain algorithm on this toolbox as it is an open-source toolbox. It is advised that water researchers, professionals, and operators implement their system using blockchain platforms (e.g., Ethereum, Hyperledger Fabric) after simulating it using WDSchain. Future work can be extended to propose new consensus mechanisms that can involve data intelligence technologies and develop extended functions of simulating the communication between the selected blockchain nodes with the comprehensive review of the blockchain use-cases in the water systems.

**Author Contributions:** Conceptualization, H.H.M., W.W. and Y.W.; methodology, H.H.M. and W.W. and Y.W.; software, H.H.M.; validation, H.H.M., W.W. and Y.W.; formal analysis, H.H.M., W.W., Y.W.; investigation, H.H.M., W.W. and Y.W. ; data curation, H.H.M.; writing—original draft preparation, H.H.M.; writing—review and editing, H.H.M., W.W. and Y.W.; supervision W.W.; funding acquisition W.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Training Networks (ITN)-IoT4Win grant agreement No [765921].

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Software Availability. Name of software: Water Distribution System in Blockchain toolbox (WDSchain). License: Birmingham City University (BCU) and IoT4Win H2020 project. Software required: MATLAB, EPANET2 programmers Toolkit. Available on GitHub: https://github.com/HaithamHmahmoud/WDSchain, accessed on 12 July 2021.

**Acknowledgments:** The authors like to acknowledge the funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Training Networks (ITN) IoT4Win grant agreement No [765921].

**Conflicts of Interest:** All Authors report no conflict of interest relevant to this article.

## References

1. Hassanzadeh, A.; Rasekh, A.; Galelli, S.; Aghashahi, M.; Taormina, R.; Ostfeld, A.; Banks, M.K. A review of cybersecurity incidents in the water sector. *J. Environ. Eng.* **2020**, *146*, 03120003. [CrossRef]
2. Fekete, B.M.; Revenga, C.; Todd, M. *The Global Risks Report 2018*, 13th ed.; World Economic Forum: Geneva, Switzerland, 2018.

3. Mahmoud, H.; Wu, W. Cyber-Physical System Security Open Challenges in Smart Water Networks. *Zenodo* **2020**. [CrossRef]
4. Dogo, E.M.; Salami, A.F.; Nwulu, N.I.; Aigbavboa, C.O. Blockchain and Internet of things-based technologies for intelligent water management system. In *Artificial Intelligence in IoT*; Springer: Antalya, Turkey, 2019; pp. 129–150.
5. Suciu, G.; Nădrag, C.; Istrate, C.; Vulpe, A.; Ditu, M.C.; Subea, O. Comparative analysis of distributed ledger technologies. In Proceedings of the 2018 Global Wireless Summit (GWS), Chiang Rai, Thailand, 25–28 November 2018; pp. 370–373.
6. Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renew. Sustain. Energy Rev.* **2019**, *100*, 143–174. [CrossRef]
7. Fontein, R. Comparison of static analysis tooling for smart contracts on the evm. In Proceedings of the 28th Twente Student Conference on IT, Twente, The Netherlands, 2 February 2019.
8. Hill, E. Who's Snooping on Your Blockchain Transactions? Medium. 2019. Available online: https://medium.com/hackernoon/whos-snooping-on-your-blockchain-transactions-7c4ae1c556d3 (accessed on 12 July 2021).
9. Singh, R.; Tanwar, S.; Sharma, T.P. Utilization of blockchain for mitigating the distributed denial of service attacks. *Secur. Priv.* **2020**, *3*, e96. [CrossRef]
10. Astarita, V.; Giofrè, V.P.; Mirabelli, G.; Solina, V. A review of blockchain-based systems in transportation. *Information* **2020**, *11*, 21. [CrossRef]
11. Dwivedi, A.D.; Srivastava, G.; Dhar, S.; Singh, R. A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors* **2019**, *19*, 326. [CrossRef] [PubMed]
12. Pincheira, M.; Vecchio, M.; Giaffreda, R.; Kanhere, S.S. Cost-effective IoT devices as trustworthy data sources for a blockchain-based water management system in precision agriculture. *Comput. Electron. Agric.* **2021**, *180*, 105889. [CrossRef]
13. Perboli, G.; Musso, S.; Rosano, M. Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. *IEEE Access* **2018**, *6*, 62018–62028. [CrossRef]
14. Pérez Ortiz, Y. How Blockchain Technology Could Improve the Quality of Drinking Water in Puerto Rico. 2018. Available online: https://ssrn.com/abstract=3266166 (accessed on 12 July 2021).
15. Youssef, S.B.H.; Rekhis, S.; Boudriga, N. A blockchain based secure IoT solution for the dam surveillance. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference, Marrakesh, Morocco, 15–18 April 2019; pp. 1–6.
16. Predescu, A.; Arsene, D.; Pahonțu, B.; Mocanu, M.; Chiru, C. A Serious Gaming Approach for Crowdsensing in Urban Water Infrastructure with Blockchain Support. *Appl. Sci.* **2021**, *11*, 1449. [CrossRef]
17. Pee, S.J.; Nans, J.H.; Jans, J.W. A simple blockchain-based peer-to-peer water-trading system leveraging smart contracts. In Proceedings of the International Conference on Internet Computing, Philadelphia, PA, USA, 18–20 October 2018; pp. 63–68.
18. Pincheira, M.; Vecchio, M.; Giaffreda, R.; Kanhere, S.S. Exploiting constrained IoT devices in a trustless blockchain-based water management system. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency, Toronto, ON, Canada, 2–6 May 2020; pp. 1–7.
19. Kassou, M.; Bourekkadi, S.; Khoulji, S.; Slimani, K.; Chikri, H.; Kerkeb, M. Blockchain-based medical and water waste management conception. In *E3S Web of Conferences*; EDP Sciences: Paris, France, 2021; Volume 234, p. 00070.
20. Campbell, R. *The Genesis System Wants to Record Cleaned Fracking Water on the Blockchain*; Bitcoins Magazine: Nashville, TN, USA, 2017.
21. Sriyono, E. Digitizing water management: Toward the innovative use of blockchain technologies to address sustainability. *Cogent Eng.* **2020**, *7*, 1769366. [CrossRef]
22. Mahmoud, H.H.M.; Wu, W.; Wang, Y. Secure Data Aggregation Mechanism for Water Distribution System using Blockchain. In Proceedings of the 2019 25th International Conference on Automation and Computing, Lancaster, UK, 5–7 September 2019; pp. 1–6.
23. Arandia, E.; Eck, B.J. An R package for EPANET simulations. *Environ. Model. Softw.* **2018**, *107*, 59–63. [CrossRef]
24. Demetrios, G.; Eliades, M.; Stelios, V.; Polycarpou, M.M. EPANET-MATLAB Toolkit: An Open-Source Software for Interfacing EPANET with MATLAB. In Proceedings of the IWC—14th International CCWI Conference Computing and Control for the Water Industry Conference, Amsterdam, The Netherlands, 7–9 November 2016; pp. 7–9.
25. Jonkergouw, P.M. Simulating Chlorine Decay in Water Distribution Systems. Ph.D. Thesis, University of Exeter, Exeter, UK, 2007.
26. Taormina, R.; Galelli, S.; Douglas, H.; Tippenhauer, N.O.; Salomons, E.; Ostfeld, A. A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems. *Environ. Model. Softw.* **2019**, *112*, 46–51. [CrossRef]
27. Berry, J.; Riesen, L.A.; Hart, W. *TEVA-SPOT Toolkit 1.2*; Technical Report; Sandia National Laboratories: Albuquerque, NM, USA, 2007.
28. Klise, K.A.; Nicholson, B.; Laird, C.D. *Sensor Placement Optimization Using Chama*; Sandia National Laboratories: Albuquerque, NM, USA, 2017.
29. Shahra, E.Q.; Wu, W. Water contaminants detection using sensor placement approach in smart water networks. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–16. [CrossRef]
30. Eck, B.J. An R package for reading EPANET files. *Environ. Model. Softw.* **2016**, *84*, 149–154. [CrossRef]
31. Hart, D.; Klise, K.A.; Bynum, M.L.; Laird, C.D.; Seth, A. *Water Network Tool for Resilience (WNTR) v. 2.0*; Technical Report; Sandia National Lab.: Albuquerque, NM, USA, 2019.
32. Klise, K.A.; Bynum, M.; Moriarty, D.; Murray, R. A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study. *Environ. Model. Softw.* **2017**, *95*, 420–431. [CrossRef]

33. Klise, K.; Siirola, J.; Hart, D.; Hart, W.; Phillips, C.; Haxton, T.; Murray, R.; Janke, R.; Taxon, T.; Laird, C.; et al. *Water Security Toolkit User Manual Version 1.2*; Sandia National Lab.: Albuquerque, NM, USA, 2014.
34. Zou, J.; Ye, B.; Qu, L.; Wang, Y.; Orgun, M.A.; Li, L. A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services. *IEEE Trans. Serv. Comput.* **2018**, *12*, 429–445. [CrossRef]
35. Iotw. IOTW Project. Available online: https://iotw.io/ (accessed on 12 July 2021).
36. Nguyen, G.T.; Kim, K. A survey about consensus algorithms used in blockchain. *J. Inf. Process. Syst.* **2018**, *14*, 101–128.