Water Supply

IWA PUBLISHING

Water Supply Vol 22 No 3, 2832 doi: 10.2166/ws.2021.433

A database system for querying of river networks: facilitating monitoring and prediction applications

Erik Bollen [a,b,*], Brianna R. Pagán [c,d], Bart Kuijpers [a], Stijn Van Hoey [e], Nele Desmet [b], Rik Hendrix [b], Jef Dams [b] and Piet Seuntjens [b,f,g]

[a] Databases and Theoretical Computer Science Group and Data Science Institute (DSI), Hasselt University and transnational University Limburg, Hasselt, Belgium
[b] VITO – Flemish Institute for Technological Research, Mol, Belgium
[c] Spacesense.ai, Paris, France
[d] Hydro-Climate Extremes Laboratory (H-CEL), Department of Environment, Ghent University, Ghent, Belgium
[e] Fluves, Ghent, Belgium
[f] Biomath, Department of Data Analysis and Mathematical Modeling, Ghent University, Ghent, Belgium
[g] Institute for Environment and Sustainable Development, University of Antwerp, Antwerp, Belgium
*Corresponding author. E-mail: erik.bollen@uhasselt.be

EB, 0000-0002-9287-1094; BRP, 0000-0001-8028-0310; BK, 0000-0001-5774-0948; SVH, 0000-0001-6413-3185; ND, 0000-0003-3025-7512; RH, 0000-0002-1572-1279; JD, 0000-0002-4302-1034; PS, 0000-0003-0554-9898

ABSTRACT

The increasing availability of real-time *in situ* measurements and remote sensing observations have the potential to contribute to the optimisation of water resources management. Global challenges such as climate change, intensive agriculture and urbanisation put a high pressure on our water resources. Due to recent innovations in measuring both water quantity and quality, river systems can now be monitored in real time at an unprecedented spatial and temporal scale. To interpret the sensor measurements and remote sensing observations additional data, for example on the location of the measurement, and upstream and downstream catchment characteristics, are required. In this paper, we present a data management system to support flow-path-related functionality for decision making and prediction modelling. Adding meta-datasets and facilitating (near) real-time processing of sensor data questions are key concepts for the systems. The potential of the database framework for hydrological applications is demonstrated using different applications for the river system of Flanders. In one, the database framework is used to simulate the daily discharge for each segment within a catchment using a simple data-driven approach. The presented system is useful for numerous applications including pollution tracking, alerting and inter-sensor validation in river systems, or related networks.

Key words: data driven modelling, IoT, recursive querying, relational databases, river monitoring, water management

HIGHLIGHTS

- Line geometries
- Queryable topology
- Flow-path detection
- Discharge prediction
- Data driven hydrological applications

INTRODUCTION

The availability of geospatial data has increased exponentially in recent years, at the same time the expansion of both *in situ* sensor networks and Earth observation data (i.e. remote sensing from satellites) has resulted in unprecedented volumes of information with increasing spatial and temporal resolution (McCabe *et al.* 2017; Reichstein *et al.* 2019) for hydrological applications. One rapidly expanding source of *in situ* measurements comes from the deployment of monitoring networks within 'Internet of Things' (IoT) systems (Zhang *et al.* 2018). As these systems are increasingly adopted by governments and environmental managers, a continuing challenge is how to properly process and integrate new volumes of data into models or frameworks that allow informed decision making (Havlik *et al.* 2011).

Physically based modelling has successfully been used within water resources management for many years. However, the suitability of such models when digesting a vast amount of data from various sources for real-time applications becomes questionable due to the trade-off between more realistic physical representativeness and required computational power. Additionally, commonly used spatially distributed hydrological models (i.e. the Soil and Water Assessment Tool (SWAT) (Arnold *et al.* 1998)), rely on spatial and temporal aggregation, empirical parameterisation and extensive calibration. The implementation of complementary data-driven approaches, which focus on the relationships between input and desired outputs, have become increasingly popular and have proven to successfully represent hydrological processes (Solomatine & Ostfeld 2008; Ahani *et al.* 2018), although most emphasis regarding data-driven approaches in hydrology has been focused on water quality rather than quantity predictions (Maier *et al.* 2010). An exception is the work of Li & Willems (2020), who present a hybrid approach to predict flooding in sewage and surrounding areas. The presented approach uses a graph-based model and graph algorithms to determine the flow path in the sewage network. Adequate computational tools and frameworks are necessary for establishing data-driven hydrological models so that they are easy to use for hydrologists and water managers. Hydrological applications depend on geospatial representations and interconnections to trace the flow path of water through the river system, similar to the tracing used by Li & Willems (2020). Databases are well suited to store and represent this information by translating the geospatial representation into a topology. Relational databases use tables to store the information, therefore, in order to store a topology a specific data model has to be used (Urubkin *et al.* 2020). Relational databases have successfully integrated real-time hydrologic data for storage and retrieval purposes. Examples include the Observation Data Model (Horsburgh *et al.* 2008), which has been applied to physical and chemical water systems by the United States Environmental Protection Agency's Water Quality Exchange and Water Quality Portal (US Environmental Protection Agency 2021), and the United States Geological Survey's National Water Information System (US Geological Survey 2021). The Watershed Monitoring and Storage Database (WMSD) takes these integrations a step further by allowing data manipulations and calculations within the database itself, resulting in time-weighted constituent fluxes (Carleton *et al.* 2005a, 2005b). Outputs from such frameworks can also serve as data input for more traditional physically based models. Lastly, the United States Geological Survey developed an online mapping application for exploring downstream and upstream water flows along America's rivers and streams (US Geological Survey 2015). An application that shows the possibilities of using the US river data is River-runner developed by Sam Learner, which is based on the general NLDI API (Water Data Labs 2021). As River-runner and the NLDI API show, similar systems exist but are most of the time single-purpose builds and therefore are not easily extended with new functionality. This in turn leads to a less wide applicability and less possibility for reuse.

As previously mentioned, prior studies and tools have successfully traced water flow paths, including case studies in Belgium such as the Graph Tracing Engine (GTE) (Geosparc 2019). The GTE was developed to facilitate in-browser sewage tracing where a quick response time is crucial. Therefore, the GTE is built with an in-main-memory structure using JGraph (May *et al.* 2019). The service is queryable through an HTTP API and it provides the possibility to add different datasets. The in-memory storage needs to be built at the start of the service and the overall created connection information is lost as soon as the application stops. Persistent storage would mean that corrections or changes made to the data, during the creation of the overall flow-information, are not lost in the long-term. Persistent storage also adds the ability to set up versioning of the data where different editions of the same dataset can be stored and queried based on the user's preference. Moreover, the recent development and release of the new PostGIS version enables faster querying in relational databases and additional tuning of the database can make the speed argument less relevant (Ramsey 2020).

In this paper, we present a data management system and its set-up that is designed to support flow-path-related functionality for decision making and prediction modelling. Functionalities that can be realised with it include distance calculations, upstream drainage area determination, connectivity analysis and more. The presented system is designed to be reusable in different projects and is able to be easily extended with new functionalities if needed. This data management solution differs from project-specific systems in the sense that it is a central server-based approach that does not need to be rebuilt for each project, reducing project start-up times and more. This topology-based relational database solution using spatial river data is realised by using PostgreSQL together with spatial extensions PostGIS and pgRouting.

The contributions of this paper can be summarised as follows. In general terms, we propose how well-established data management techniques from the fields of computer science can be applied to data management for hydrological modelling. In particular, we show:

1. how to generalise the storage (how river systems and other transportation systems can be modelled and stored in one and the same data store);

2. how to open it up for implementation of algorithmic optimisations (how the proposed model can be used to apply the insights of queries and optimisations that are extensively studied in computer science, for example the shortest-path query or transitive closure queries); and
3. how to enable reusability of functionalities for modelling (how to ensure that functionality that is developed for one application or model can be reused by other, possibly future, applications and models).

A data-driven water-flow heuristic over the Flanders region and a machine-learning-based salinisation prediction case study are used to demonstrate the applicability of the system.

METHODS

When dealing with network systems such as rivers and transportation pathways, it is common to utilise Geographical Information Systems (GIS) to represent specific properties. Here we consider a hydrological dataset which includes rivers represented as line segments, with attributes being properties of each (i.e. drainage area associated with each segment). We demonstrate how this data can be stored and organised to facilitate solving real-world hydrological problems including salinisation modelling and water flow predictions. The steps to set up such a database and process the data are discussed, with special considerations for maintaining data quality. Next, efficient querying is tested and implemented for topology-type problems. Finally, the constructed database and functions are presented for specific hydrologic use-cases including flow-path and drainage retrievals.

Setting Up the database

Different database types exist. Relational databases are chosen here for the river use cases due to their broad functionalities with GIS data. Especially the PostgreSQL relational database seems well suited for this application. The PostGIS and pgRouting extensions are added to apply additional support for geometric data types and more specifically the conversion of single geometries to fully related water flow direction topologies.

Selection of database relation type

The ideal database structure depends on the application. As the envisioned system is meant to represent hydrological flow conditions, segments need to be linked, creating a 'flows-to' relationship for the segments which can be stored. The characteristics of this relation can vary depending on the situation and are more commonly a binary or topology-based relation. In a binary situation, one segment is linked to the segment following directly after it. One segment can have multiple segments to which it flows. For each of these pairings, a tuple must be created. Formally, given segments A and B, let S(A) be the start point of segment A and E(A) be its end point (and similar for B). When the identifiers of the segments A and B are X and Y, respectively, the pair (X, Y) is added to the flows-to relation if $E(A)=S(B)$. For a topology-based relation, according to the standard ISO/IEC 13249-3 (SQL/MM) Topology-Network, each segment is assigned a begin node (source) and an end node (target), corresponding to the start of the segment and the end of the segment. The water in one segment then flows from the source to the target node. Subsequently, the water flows from one segment A to a second segment B if the target of A is the same node as the source of B. When working with a topology, the standard ISO/IEC 13249-3 (SQL/MM) 'Topology-Network' needs to be used and not the 'Topology-Geometry' (also in ISO/IEC 13249-3). Despite its name, the 'Topology-Geometry' framework is not suited for storing the flows-to information as it does not allow two segments to cross each other. This implies that rivers would not be able to intersect without directly flowing into each other. But there are several cases in which two water streams can cross each other. Many basic functionalities are provided for the Topology model by Postgres and its extensions, such as conversion tools and network analysis functions. If the binary flows-to model is chosen, these functionalities must be developed. Subsequently, the Topology-Network-based model is selected for use in this work.

Conversion of river segments to a topology-based database

After setting up PostgreSQL and enabling the PostGIS and pgRouting extension, a collection of river segments can be imported using the available shapefile importing tools. Functionalities from pgRouting and the built-in conversion tool allow for the creation and assigning of nodes to start and end. This includes the creation of geometric points and the tables to store them.

Applying database queries

With the database established, questions regarding the structure of the network or properties of the segments can be solved. In this section a few of these questions are discussed. PostgreSQL is used and the database is queried using SQL (Structured Query Language). Basic queries implemented include ones that retrieve a specific segment, geometry or properties. For example 'what is the length of segment X', with X being the identification of a specific segment. These types of queries are considered to be simple 'SELECT…FROM…WHERE' queries and are not discussed further. Retrieving the flow path, upstream and downstream, as well as finding the distance to a given point (for example the closest outlet to the sea) are queries explained in the rest of this section. These queries are demonstrated using the example of a river network, but can be generalised for any transportation network.

Flow path querying

With the topology or flows-to relation established for the dataset, connectivity queries can be created. The reachability query is a transitive closure-like query that asks where, in this case, the water can flow to or can come from. In order to answer this question all possible upstream or downstream paths need to be found given a specific starting point. This type of query is the basis for answering many other questions related to the river system. Therefore, the topology or flows-to relation together with the reachability query forms the foundation of the system and will be a vital part in the example applications given.

In order to determine the path of water for a particular segment, a recursive query for a given point on the topology needs to be resolved. The Common Table Expressions (CTE) in SQL is used to obtain recursive query behaviour, as follows:

```
WITH RECURSIVE result_relation AS (
    start_query
UNION
    selection_query
    result_query);
```

The *result_relation* defines a temporary relation that is used to store the recursively found data. This relation is filled at the start with the tuples resulting from the *start_query*, effectively placing the start segment in the *result_relation*. With each iteration of the procedure, *result_relation* is filled further with the tuples from the *selection_query* until it is unable to find new tuples that can be added to the *result_relation*. For this use case, this means that at each iteration, the next river segment of each possible path is added until no further river segments can be found. Finally, the *result_query* is executed on the *result_relation* producing the final result of the complete SQL query, effectively returning all segments that are a part of the flow path.

The actual retrieval of the downstream flow path can be calculated using the CTE structure assuming that source and target are the begin- and end-point of a segment, and *seg_table* being the table with all segments of the imported dataset:

```
WITH RECURSIVE outcome(id, source, target) AS (
    (SELECT id, source, target
    FROM seg_table
    WHERE id=id_startsegment)
UNION
    SELECT seg_table.id, seg_table.source, seg_table.target
    FROM outcome, seg_table
    WHERE seg_table.source=outcome.target
SELECT id FROM outcome);
```

The first part of the query will select the identification number, source and target of the first segment. The second half of the query then calculates the paths recursively. The query itself resolves the issue of potential loops within the topology as the union will not add a segment with the same identification, source and target twice. Retrieving the upstream flow-path information is similar to the downstream query with the exception of the WHERE condition in the *selection_query*. Here the condition has to be changed such that the *outcome.source=seg_table.target*.

It may be interesting to limit the query to a certain number of segments that are found. This is achieved by using a numerical limit on the final SELECT FROM WHERE clause. A second, more useful limitation is based on the distance of the path. The length of each segment is provided and where not available, it is calculated with PostGIS. We give the downstream query comparable to the previous one, but this time with a limit on the distance of the path:

```
WITH RECURSIVE outcome(id, source, target, distance) AS (
    (SELECT id, source, target, length as distance
    FROM seg_table
    WHERE id=id_startsegment)
UNION
    SELECT seg_table.id, seg_table.source, seg_table.target,
      (seg_table.length+outcome.distance)
    FROM outcome, seg_table
    WHERE seg_table.source=outcome.target
      AND (seg_table.length+outcome.distance<path_limit)
SELECT DISTINCT id FROM outcome);
```

The maximum length of the path is denoted with *path_limit*. This query takes the distance of each segment, adds the total distance covered since the *start_segment* and stores this information for each segment, until the sum reaches the designated maximum length of the path (*path_limit*). However, in the case of a loop, a segment visited for the second time does not have exactly the same tuple (the distance differs), resulting in the query continuing to iterate until the distance condition is violated. The DISTINCT keyword in the final query can filter out the segments found multiple times but the distance information is lost in the process.

Upstream drainage area

Some properties of the river itself or the state of the water flow are related to the drainage area of the river. In general, there can be polygons that represent an area linked to line segments with a relation between the two. Using the river example, a polygon represents the direct region surrounding the segment that contributes water to the segment. However, it is often more useful to know the entire upstream drainage area for a segment. This means all polygons of all upstream river segments need to be retrieved and combined. Given a segment in the dataset, the upstream flow-path query returns all segments that end up at the given location. Using this result, all relevant upstream drainage areas can be identified and the polygons can be merged into one. The result is one polygon representing the entire upstream area for the given segment, and this can deliver information needed in other processes. The actual SQL query for this looks like the following, assuming that source and target are the begin- and end-point of a segment, *seg_table* is the table with all segments, and *drain_table* contains the drainage area for each segment:

```
SELECT ST_union(geom) FROM drain_table WHERE id IN (
  WITH RECURSIVE outcome(id, source, target) AS (
      (SELECT id, source, target
      FROM seg_table
      WHERE id=id_startsegment)
  UNION
      SELECT seg_table.id, seg_table.source, seg_table.target
      FROM outcome, seg_table
      WHERE seg_table.target=outcome.source)
  SELECT DISTINCT id FROM outcome)
```

Distance to target

In varying applications, it is interesting to know how far a point in a network is from a border, region or other point. For this case study, it is interesting to know how far a location is from the sea, as this is important information in environmental applications to monitor seawater intrusion. It should be noted that distance along the network path is the goal and not bird-flight distance or other straight-line measurement. This can be retrieved using a very similar method to the distance-limited flow-

path query. If the location can be linked to an existing segment in the dataset, the distance-limited flow-path query can be executed until the segment of the intended location is included in the outcome. With the distance incremented during each step, a network distance will be associated with the segment and this distance can then be used in turn. We note that this is not the most efficient approach and in practice the distance to the sea can be calculated iteratively. First, all segments get assigned a null distance value. Segments which are directly connected to an ocean outflow are designated with a zero distance. Next, all segments that have a not-null distance are assigned a distance based on the cumulative lengths of downstream paths in addition to the length of the segment in question. This step of assigning a distance to the segments with no distances is repeated until no more segments can be connected. In the river example, the points can be locations at which the water enters the sea and therefore the distance to the sea can be calculated for each segment. Calculating this property can be done at run-time for the segments or it can be calculated in advance and retrieved from the database if needed.

Applying the database for data-driven predictions

The database presented here is meant to support applications that solve more complex problems. The biggest advantage of using a system as presented in this paper is that recurrent basic tasks can be solved once and reused. Updating the data then also directly renews all applications relying on it. In order to demonstrate which applications are possible, two data-driven examples will be highlighted: *Salinisation Prediction* and *Disaggregation of Discharge*.

Salinisation prediction

Seawater intrusion into freshwater systems is a longstanding environmental issue which can limit water supplies. The salinisation of rivers and groundwater is exacerbated by climate change and itself has a great impact on the surrounding wildlife and agriculture (Gobin 2012). The influence the North Sea has on a river location in Belgium is directly related to the distance of that location to the North Sea. In this example, historical monthly grab samples of salt concentration along the Flanders river network are used along with physical properties of the rivers/neighboring soils to predict salinity concentrations along the entire river segment using a neural network. One physical property deduced from the created system is the distance of each river segment to the ocean.

Disaggregation of discharge

In this use-case we present a proof of concept to illustrate how this database and the connection information can facilitate data-driven predictions. Flow path, drainage area and *in situ* sensor information is used to model discharges over an upstream river network by disaggregating discharge measurements for each individual segment by considering the flow paths. Specifically, existing *in situ* water flow stations are attributed to the nearest river segment in the database. At any selected *in situ* sensor, the database identifies all upstream segments and provides discharge areas for each segment (in addition to the total basin area). This information is then used to derive estimates of distributed upstream flow, calculated as a fraction of outflow proportional to the discharge area of each individual segment to the entire river basin. This proof of concept shows how the demonstrated database system and the topological data of a river system can be used to facilitate data-driven prediction. It also illustrates how the database is used and how different simple applications of the database, for example the Drainage Area Determination, can be combined to enable more advanced use-cases.

The developed disaggregation concept is compared with discharge simulated by a hydrological model at specific locations along the trajectory of the river. Therefore, a series of rainfall–runoff models are set up to simulate the daily discharge of the study area. The Hydrological Model Assessment and Development (Hydromad) (Jakeman *et al*. 2013) and Framework for Understanding Structural Errors (FUSE) (Clark *et al.* 2008) R packages are used in this study to set up hydrological models. The Hydromad and FUSE packages offer a flexible framework to set up a series of soil moisture accounting models (SMA) that allow the simulation of effective rainfall and runoff (Andrews *et al.* 2011). Precipitation, potential evaporation and measured discharge are fed into the packages. The packages offer tools for auto-calibrating and testing the varying soil moisture accounting models. Using the Hydromad and FUSE packages together we use six models to simulate the region in question. Depending on the location for which the discharge is simulated, the auto-calibration assigns the best model performance for the following models: Sacramento Soil Moisture Accounting (Burnash 1995), Génie Rural à 4 paramètres Journalier (Perrin *et al.* 2003), and Single-Bucket model (Bai *et al.* 2009) in Hydromad and TOPMODEL (Beven & Kirkby 1979), Precipitation–Runoff Modeling System (Leavesley *et al.* 1983), and variable infiltration capacity model (VIC) (Zhao 1977) model in FUSE. For each targeted watershed, meteorological data is fed into the

models, together with the upstream drainage area retrieved from the database, and each model self-calibrates with the provided runoff data. The calibrated model is then used to predict discharge at river outlets with provided meteorological data. Only the best performing model of all six calibrated rainfall–runoff models is used for comparison. We note that for each test case and station a different model can be chosen. The model outputs and the results of the disaggregation are then compared with the actual measurements of stations in the upstream river basin in order to evaluate the performance of both prediction methods.

Case study

The established framework and queries are applied to the region of Flanders. The two applications are implemented for the same region and then executed with real data provided by regional environmental organisations.

Region of Flanders

The region of Flanders is located in the northern part of Belgium (Figure 1). In April 2019, the Flemish government initiated the project 'Internet of Water' (IoW) to support more accurate monitoring of water systems. A dense network of water quality sensors is being deployed in rivers, canals and lakes throughout Flanders that can provide data at a sub-hourly resolution. The implementation of such a dense sensor network over a hydrologically complicated and environmentally vulnerable region creates an ideal test environment for the integrated geospatial data-driven database framework.

River datasets for flanders

The official inventory of all rivers in Flanders is referred to as the 'Vlaamse Hydrografische Atlas' (VHA) created by the environmental agency, 'Vlaamse Milieumaatschappij' (VMM) (Geopunt 2020). The dataset is provided as a shapefile with rivers represented as geometric line segments and is updated every few months to reflect any changes along the network. Each segment within the dataset is assigned an identification number (VHAS). The line geometry of a segment represents a river stretch with the same characteristics, and is often a few tens of metres long. All segments joined together form the complete river network.

A second related dataset is the collection of drainage areas where for each segment, identified by the VHAS, a local drainage area is provided as a polygon geometry. This polygon around the segment represents the area that drains to that segment. The calculation and aggregation of drainage areas along the river network are important inputs for later modelling steps.

During the processing of the vector data, the dataset is screened for issues. One of the first issues noticed is a small drawing error, i. e. where the start- and end-points of two geometries do not align. These misalignments can vary in severity ranging from a few centimetres to a few metres. To resolve this issue, the geometry is corrected or a buffer is applied during the

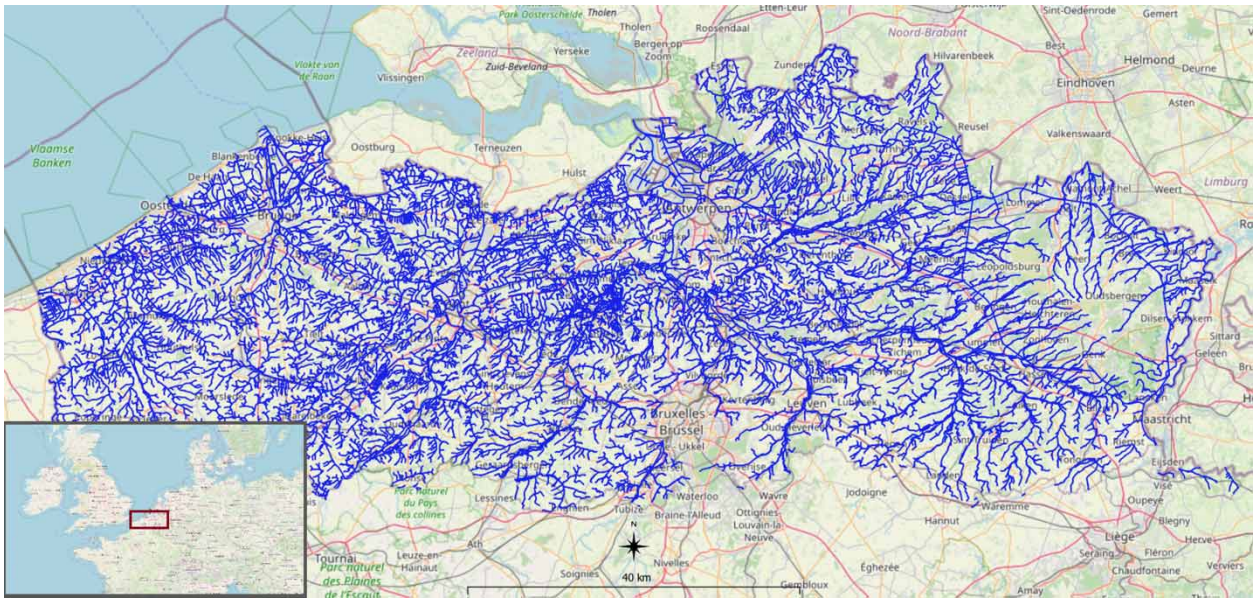

Figure 1 | The Flanders region of Belgium with rivers (blue). Visualised in QGIS using OpenStreetMap as the basemap.

matching step. The first solution, albeit time-consuming, is preferred as it is a permanent solution and no false-positive matches are introduced. A second issue that can be encountered is the wrong direction of a geometry. In this case study, the direction of the segment should correspond to the flow direction of the water. However, in some situations, a direction is assigned incorrectly, causing a loop where more downstream geometries cannot be matched correctly. For these cases, the flow direction must be corrected once manually, because field knowledge is required. Finally, the last issue recognised involves the determination of borders for the dataset. As rivers flow between regional and country boundaries, the same quantity and quality of data is not always available for the same river. For this case study, data ends at the border of a nearby region. In some instances, the river can flow out of the region, and re-enter elsewhere. However, due to missing data in the neighboring area, the two segments are not linked. The only solution to this problem is to add the missing data, although combining these datasets is not trivial as the literature of data conflation shows (Seth & Samal 2016). Finally, we want to note that applying these corrections is only a one-time issue during preprocessing. The space and time requirements of this step do not influence the functioning of the system and its applications later on.

Test cases

The first test case is located on the upper Herk river basin. The two main rivers within the Herk catchment are the Mombeek and the Herk (Figure 2). The station L09_163 is used as input for the disaggregation, and upstream stations LS09_165 and L09_167 are used for validation.

In the second test case the measurement stations are placed on the river Molenbeek (Figure 2). There is one downstream station, LS06_347, and three validation stations, LS06_348, LS06_34D and LS06_34E. This test case is a rather small case as many parts further downstream are influenced by sluices, barriers and pumps.

The third and last test case is the Gete (Figure 2). The downstream station functioning as the input for the disaggregation is L09_153. The two stations upstream, L09_155 and L09_157, are used to validate the results.

RESULTS

In this section, first the querying of the database is shown applied to the Flanders region. The results of the flow-path querying are depicted and followed by the process of determining the distance to the sea and upstream drainage area for the same region. With the actual details established for the region, the two applications are presented, being *Salinisation Prediction* and *Discharge Disaggregation*.

Querying the database

In this subsection the three queries, flow-path query, distance to sea, and upstream drainage-area query, are applied to the case study region.

Flow path querying

In this example the downstream flow-path of a specific segment in the river Gete near Tienen is retrieved. The id, *vhas*, of that segment is 2004971 and the query shown earlier filled in with the name of the segments table, wlas, yields:

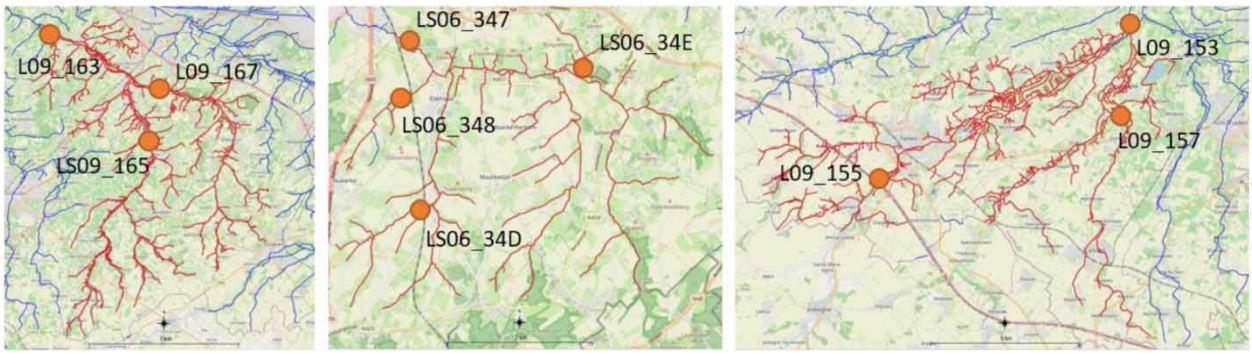

Figure 2 | Left, overview of the Herk test case; middle, overview of the Molenbeek test case; right, overview of the Gete test case. All segments involved are marked red and the measuring stations are indicated with their ID. Visualisation is realised using QGIS and OpenStreetMap.

```
WITH RECURSIVE outcome(vhas, source, target) AS (
    (SELECT vhas, source, target
    FROM wlas
    WHERE vhas=2004971)
UNION
    SELECT wlas.vhas, wlas.source, wlas.target
    FROM outcome, wlas
    WHERE wlas.source=wlas.target
SELECT vhas FROM outcome);
```

The result of this query is visualised in Figure 3 where all found segments are coloured red with, in the background, all segments of VHA in blue and OpenStreetMap.

Distance to sea

The incremental method discussed in the Methods section is applied in practice to the segments in the Flanders region. First the end segments need to be determined. The segments that end in the sea were in this case manually selected, primarily on the west coast of Flanders, which is directly exposed to the North Sea. Additional segments were chosen more inland and in the north of the country. We note that this is a one-time step that does not influence the querying later on. However this step could be automated if needed by deducing the location of the segments using related GIS data maps. The first are around the Scheldt River, which continues to flow into the Netherlands, but soon after drains to the North Sea. The other segments are

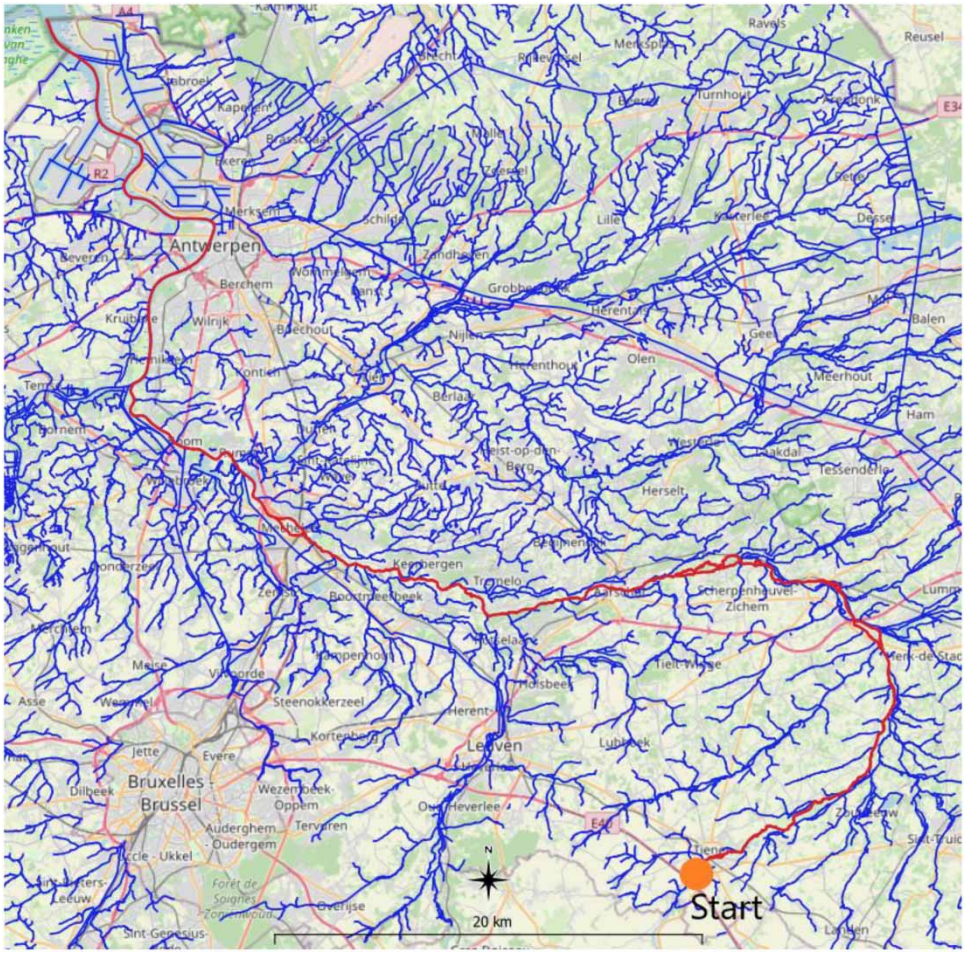

Figure 3 | Example of downstream flow-path result for the river Grote Gete in Tienen. The trajectory is found as far as the border of Belgium, where the river Scheldt flows into the Netherlands.

along the Gent–Terneuzen Channel, which connects the inland port of Ghent also to the North Sea. These segments get a distance of zero assigned, after which the distance for each next segment can be calculated by adding the distance of the last traversed segment. A majority of the segments in the province of Limburg belong to the Maas River basin, which extends beyond the eastern Belgian border. In order to assign those segments a valid value, the last segments of the river Maas in Belgium are assigned an approximate distance to their endings in the Netherlands. In practice, calculating the distance was realised by using PSQL but the same is possible with any scripting language such as Python. For each segment, the pre-calculated distance can be retrieved with a basic SELECT…FROM…WHERE query that takes the VHAS of the segment.

Upstream drainage area

For this application the queries that were established in the previous section are used. With them, all upstream segments can be determined for each segment. In each case the list of IDs, which is the result of the upstream flow-path query, is used in a simple selection process where all corresponding local drainage areas are retrieved from the drainage area dataset. By taking the union of these local drainage areas, one total-upstream drainage area is built. This building process is realised using the ST_Union function that is provided in PostGIS. Notice that this step can be done for each segment separately, resulting in a total-upstream drainage for each segment. If the calculation is done for the segment near Tienen, with *avs* being the table with the drainage area geometries stored in the *geom* column, the query is:

```
SELECT ST_union(avs.geom) FROM avs WHERE vhas IN (
  WITH RECURSIVE outcome(vhas, source, target) AS (
    (SELECT vhas, source, target
    FROM wlas
    WHERE vhas=2004971)
  UNION
    SELECT wlas.vhas, wlas.source, wlas.target
    FROM outcome, wlas
    WHERE wlas.target=outcome.source)
  SELECT DISTINCT vhas FROM outcome)
```

Although it is possible in theory to do this calculation in the moment at which it is needed, it is interesting to pre-calculate this for the entire dataset. The VHA contains roughly 50,000 segments and for some segments around 20,000 segments turn out to be upstream of them. This is for example the case for the segment of the river Scheldt in Antwerp. Taking the union of those 20,000 local drainage area polygons imposes some processing time. Therefore, for the practical realisation of the database systems, it was decided to calculate the total upstream drainage area for each segment in advance and store it in the database. However, recent updates and changes in the Postgres database and PostGIS extensions promise faster processing, because of which this preprocessing may not be needed any more (Ramsey 2020).

Applications

Now the applications are presented for the Flanders region. The queries demonstrated before form an essential part in the realisation of these applications.

Salinisation prediction

The distance-to-sea measure has proven to be a valuable feature for data-driven seawater intrusion modelling in the same region (Pagán *et al.* 2020). In this use-case, 200,000 grab sample measurements of electrical conductivity (EC), a proxy for salinity, are classified into five distinct ranges (very low to very high), and extrapolated to each segment using a neural network (Figure 4). The objective of this use-case is to provide expected salinity for any given month or season, using the given characteristics of the river segment and historical salinity measurements over those segments. For input features, several land-use characteristics including pH, organic matter content, and fractions of sand/silt/clay are considered. In addition, the month of measurement and distance-to-sea measurement explained above are also included as features. A simple feedforward neural network was trained and validated over the 200,000 grab samples, using all input features to correctly classify the salinity ranges. The data was split into 75% training and 25% for validation. The best-performing neural network configuration was rather shallow, with only three layers (two with 24 neurons and a single dense layer), using 100 epochs and a batch size of 50. The training accuracy yielded a coefficient of determination (R^2) of 0.42 and validation R^2 of 0.41. The distance-to-sea

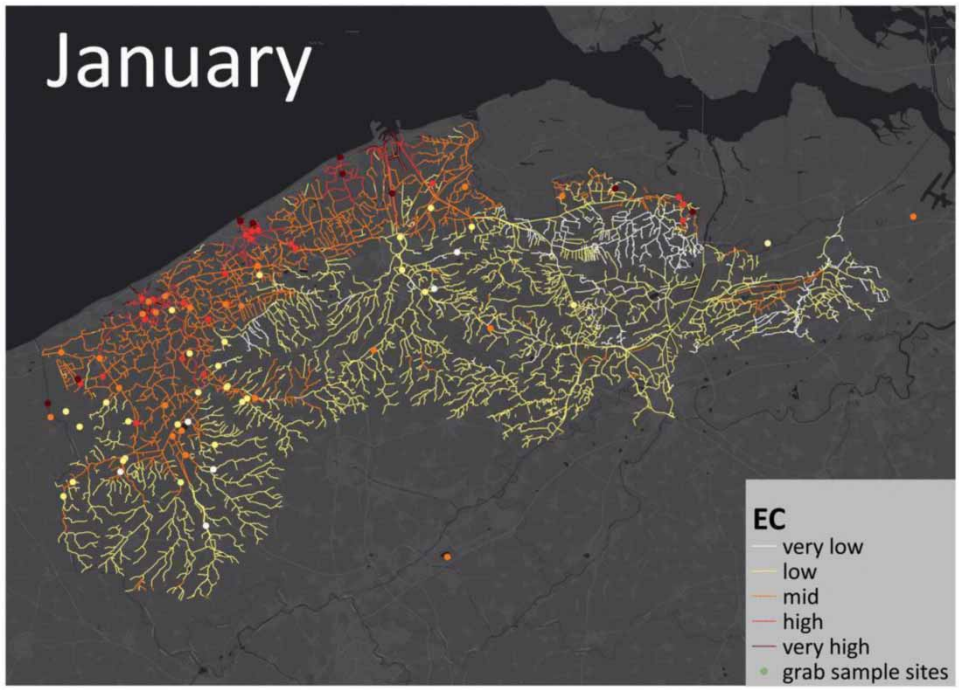

Figure 4 | The classifications of electrical conductivity (EC), a proxy of salinity concentration, and grab samples (points) extrapolated to each segment (line).

measurement had the highest feature importance compared with all other features. Due to the created database, those values can be easily retrieved and recalculated swiftly if other starting values or changes in the data are applied.

Disaggregation of discharge

The disaggregation and rainfall–runoff model outcomes are compared with the actual measurements from the rivers. This comparison is quantified by using the Nash–Sutcliffe efficiency coefficient (NSE) and the Moriasi *et al.* (2007) guideline is applied. This means NSE<=0.50 is considered unsatisfactory, 0.50<NSE<=0.65 satisfactory, 0.65<NSE<=0.75 good, and 0.75<NSE<=1.00 very good. Aggregated daily runoff data provided by *waterinfo.be* are used during the timeframe 1 January 2010 until 31 December 2019. From the available model structures only the best-performing model, based on the obtained NSE coefficient, is selected. For the Herk case, the Sacramento Soil Moisture Accounting (SAC) performed best for station L09_165 and the Génie Rural à 4 paramètres Journalier (GR4 J) model had the best NSE for station L09_167. For the Gete case SAC is used for both stations, and for the Molenbeek case the SAC model was the best-performing one for all three stations.

Table 1 shows the results of the disaggregation and rainfall–runoff models for each case. For the Herk case the results from the disaggregation are a *satisfactory* fit for the first and a *very good* fit for the second location. In comparison, the rainfall–runoff model produces a borderline *good fit* for both locations with an NSE for the two locations of 0.64 and 0.64. More noticeable is the Gete case where the disaggregation results in an NSE of –0.44 at station L09_155. For the second station, L09_157, the NSE coefficient becomes 0.16. In contrast, the rainfall–runoff models can produce an accuracy of 0.54 for the L09_155 station and 0.64 for the L09_157 station. It follows that the results from the disaggregation are *unsatisfactory*, even

Table 1 | NSE coefficients of the two predictions for the three cases

	Herk		Gete		Molenbeek		
	L09_165	L09_167	L09_155	L09_157	LS06_348	LS06_34D	LS06_34E
NSE Hydromad–FUSE	0.64	0.64	0.54	0.64	0.66	0.58	0.15
NSE Disaggregation	0.59	0.83	0.65[a] (−0.44)	0.45[a] (0.16)	0.71	0.82	−0.74

[a]After applying the correction for the drainage area.

though the rainfall–runoff model shows that the region is predictable by returning a *satisfactory fit*. When looking closely at the returned results from the disaggregation, one can determine that results are finding the right signal but they are off by a constant factor. This is likely due to the fact that both basins have a portion extending beyond the Flemish border and consequently there is no drainage area information available. As one station is a constant factor above the real values and the second station a constant factor under the real values, we suspect the proportion, p, of the two drainage areas can account for this discrepancy. By multiplying the results by $1/p$ and p respectively the accuracy increases to 0.65 and 0.45. Therefore, it is assumed that if the correct drainage area information were available, improved results could be expected. For the third case, Molenbeek, the disaggregation results for the first two stations are a *(very) good* fit. The rainfall–runoff model has a *good* and satisfactory fit. Both models have a *satisfactory* fit for the third station. In Figure 5 the disaggregation result is shown for the Herk case on one day and in Figure 6 the disaggregation is plotted against the real measurements.

In summary, the disaggregation can quite accurately calculate discharge of upstream river segments using the flow-path calculations and drainage area information. The database and algorithms can therefore facilitate more advanced real-time applications related to the propagation of water and substances through the river network. It is, however, clear that the model cannot guarantee usable results for all possible locations but neither is this true for the more classical rainfall–runoff model used here. Secondly, the disaggregation requires complete drainage area data to perform at its best. We note that the disaggregation, as presented here, is only applicable for natural flowing basins. This means that it does not take into account human influences on the river system, such as barriers, sluices or pumps. Expanding the system in order to

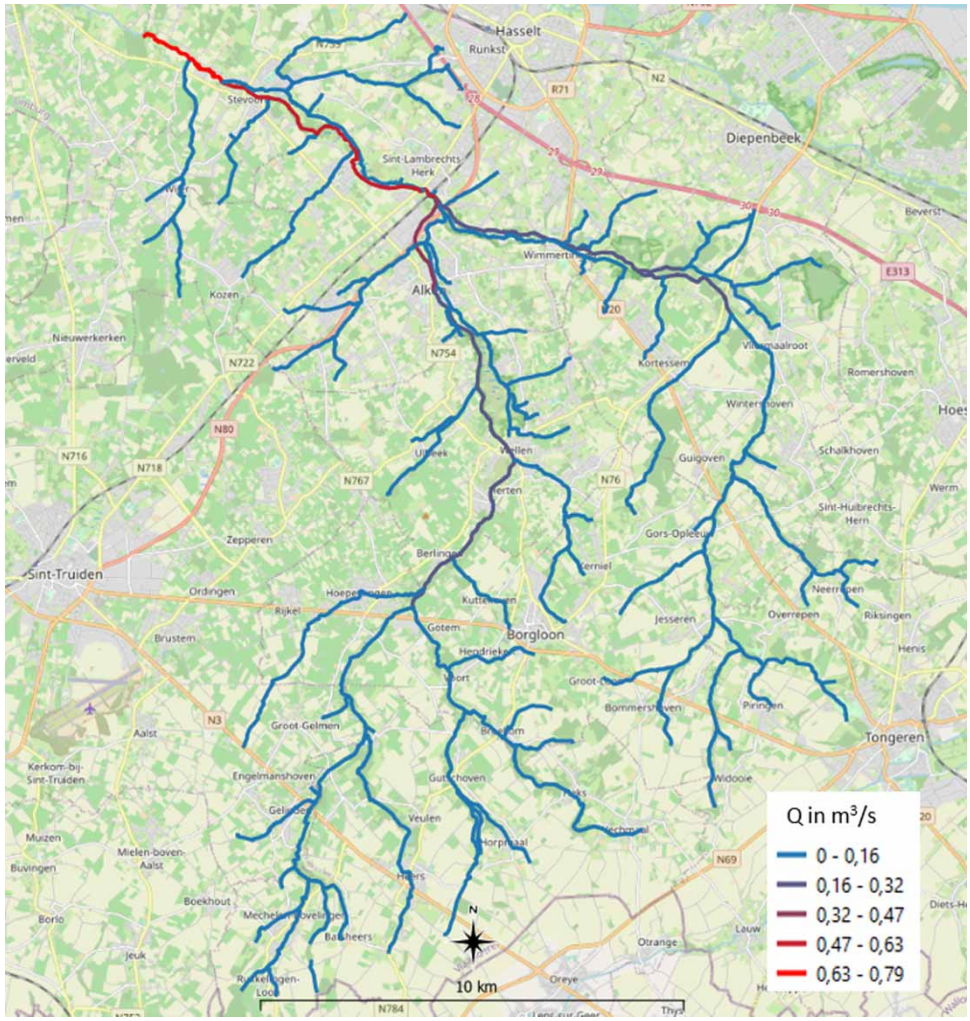

Figure 5 | The result of the disaggregation shown for the Herk case. The discharge (Q) is shown for 31 December 2019 for each segment separately.

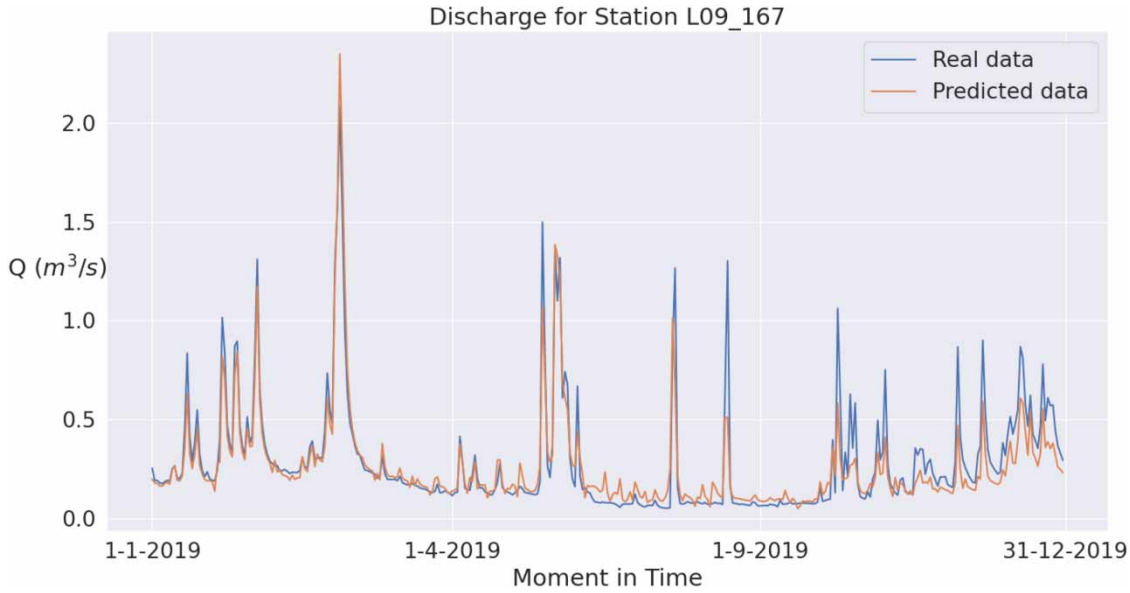

Figure 6 | Plot of the disaggregated data prediction against the measured discharge (*Q*) at station L09_167 for the year 2019.

cope with such situations is a challenge for further improvements in future work. However, the use-cases demonstrate that this database system, together with its flow-path information and drainage-area selection, can facilitate more complicated predictions. The biggest advantage of this system is that there is less set-up time needed, once such a system is in place, compared with the hydrological model, because the different parts of the database can be reused. Therefore, we would like to stress that this use-case does not aim to replace hydrological models or render them useless. This should show that for (near-real-time) applications, a data-driven approach based on this system can be a viable solution next to the existing models

Limitations

The framework and system presented have various limitations that can be improved upon. First, there are many rivers that are partially disconnected because they enter a neighbouring region or country and then re-enter the region under study. Extending the database with datasets from those regions would therefore yield even more useful information. This in turn would enhance the results of the application and use-cases. The model in itself can easily be extended by adding other datasets, however, aligning and linking the relevant parts of the different datasets is a more complicated problem, generally known as conflation, that could be the subject of other work. A similar problem arises when the data itself is updated and the newer version needs to be consolidated in the database. This is a topic also discussed in conflation papers and applying the results found in those studies and how they can be applied to allow updates of the rivers to be added to the database is to be further investigated.

Attention is placed on the fact that the topology in the database, or the flows-to relation in general, can be considered a graph. Recent advances in the domain of graph databases provide the opportunity to investigate whether the traditional relational databases can be replaced by graph databases. Whether such a replacement yields performance improvements or more solutions can be created for existing problems is an open question.

Additionally, there is also a wide range of database optimisation techniques that can be utilised to speed up processing. This also has a profound effect on applications and the probability to be used in real-time context. However, optimising the systems itself is a topic worth its own research and discussion.

Lastly it should be noted that the applications shown here are only a few examples of what could be possible with such a database. More applications can be created, also in other fields, that make use of a topology-based information system. Intersensor validation is another feasible example. Finding additional applications and integrating them into the existing system is ongoing work.

CONCLUSION

Here, we demonstrate a data-driven flow-path database system and how it can be built. Together with a set of example applications, it is shown how this system can enable river properties and discharge predictions that can support more advanced

real-time data-processing and analyses in river networks. Specifically, river segment data is used in combination with PostgreSQL, PostGIS and pgRouting to create a database containing the overall flow information. This database allows for easy extraction of downstream or upstream river flow paths and, additionally, related data can be extracted at the same time. The examples include data-inconsistency detection, drainage-area selection, and distance-to-sea measurements using flow paths. With a drainage-area-based disaggregation algorithm, as proof of concept, we show that quick, data-driven water-flow predictions are feasible. We show, using test cases, that the results produced by this disaggregation can be as good as a classical rainfall–runoff model when validating with historical observational river-flow data. Summarised, the paper provides a data management plan and implementation for dealing with river networks that aids in data storage, querying and modelling. Applying well-established data-management techniques from the fields of computer science promotes data management for hydrological models by generalising storage, opening it up for implementation of algorithmic optimisations, and enabling reusability of functionalities. This all results in reducing set-up times and data redundancy as well as facilitating the reuse of functionality for prediction modelling. By generalisation, the techniques can be extended to other fields such as road networks, electricity grids, heat networks and more. However, the existing framework has several limitations and further development is needed to capture the full complexity of hydrological conditions. The data used, for example, is limited to a small region and incorporating more data needs to be considered. We do not cover database optimisation, but this is an important step to consider in order to keep response time low for real-time applications. As the deployment of the dense IoW sensor system begins, additional near-real-time data, and the sensor data itself, will be added to the database, facilitating even more advanced applications like inter-sensor validation or more complicated water quality tracings and predictions.

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

REFERENCES

Ahani, A., Shourian, M. & Rahimi Rad, P. 2018 Performance assessment of the linear, nonlinear and nonparametric data driven models in river flow forecasting. *Water Resources Management* **32** (2), 383–399.

Andrews, F. T., Croke, B. F. W. & Jakeman, A. J. 2011 An open software environment for hydrological model assessment and development. *Environmental Modelling & Software* **26** (10), 1171–1185. doi:10.1016/j.envsoft.2011.04.006.

Arnold, J. G., Srinivasan, R., Muttiah, R. S. & Williams, J. R. 1998 Large area hydrologic modeling and assessment part I: model development. *Journal of the American Water Resources Association* **34** (1), 73–89.

Bai, Y., Wagener, T. & Reed, P. 2009 A top-down framework for watershed model evaluation and selection under uncertainty. *Environmental Modelling & Software* **24** (8), 901–916. http://dx.doi.org/10.1016/j.envsoft.2008.12.012.

Beven, K. J. & Kirkby, M. J. 1979 A physically based variable contributing area model of basin hydrology. *Hydrological Sciences Bulletin* **24**, 43–69. https://dx.doi.org/10.1080/02626667909491834.

Burnash, R. J. C. 1995 The NWS River Forecast System – catchment modeling. In: *Computer Models of Watershed Hydrology* (Singh , V. P. ed.), Water Resources Publications, Highlands Ranch, CO, USA, pp. 311–366.

Carleton, C. J., Dahlgren, R. A. & Tate, K. W. 2005a A relational database for the monitoring and analysis of watershed hydrologic functions: I. Database design and pertinent queries. *Computers & Geosciences* **31** (4), 393–402.

Carleton, C. J., Dahlgren, R. A. & Tate, K. W. 2005b A relational database for the monitoring and analysis of watershed hydrologic functions: II. Data manipulation and retrieval programs. *Computers & Geosciences* **31** (4), 403–413.

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., Wagener, T. & Hay, L. E. 2008 Framework for Understanding Structural Errors (FUSE): a modular framework to diagnose differences between hydrological models. *Water Resources Research* **44**, W00B02. https://doi.org/10.1029/2007WR006735.

Geopunt 2020 Vlaamse Hydrografische Atlas – Zones, 12 maart 2020. Available from: https://www.geopunt.be/catalogus/datasetfolder/3c22f409-ed0e-4867-a310-50cf4de853b1 (accessed 29 June 2021).

Geosparc 2019 Graph Tracing Engine Ontwikkeld in Kader van Baanbrekend BEGOOD-Project. Available from: https://www.geosparc.com/nl/news/2019/7/23/geosparc-helpt-u-samen-met-vmm-uw-afvalwater-te-volgen-mz3ps (accessed 25 June 2021).

Gobin, A. 2012 Impact of heat and drought stress on arable crop production in Belgium. *Natural Hazards and Earth System Sciences* **12** (6), 1911–1922.

Havlik, D., Schade, S., Sabeur, Z. A., Mazzetti, P., Watson, K., Berre, A. J. & Mon, J. L. 2011 From sensor to observation web with environmental enablers in the Future Internet. *Sensors* **11** (4), 3874–3907.

Horsburgh, J. S., Tarboton, D. G., Maidment, D. R. & Zaslavsky, I. 2008 A relational model for environmental and water resources data. *Water Resources Research* **44** (5), W05406.

Jakeman, T., Croke, B. & Guillaume, J. 2013 Hydrological Model Assessment and Development. Available from: http://hydromad.catchment.org (accessed 25 June 2021).

Leavesley, G. H., Lichty, R. W., Troutman, B. M. & Saindon, L. G. 1983 *Precipitation-Runoff Modeling System: User's Manual*, Water-Resources Investigations Report 83–4238. US Geological Survey, Denver, CO, USA.

Li, X. & Willems, P. 2020 A hybrid model for fast and probabilistic urban pluvial flood prediction. *Water Resources Research* **56**, e2019WR025128. https://doi.org/10.1029/2019WR025128.

Maier, H. R., Jain, A., Dandy, G. C. & Sudheer, K. P. 2010 Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. *Environmental Modelling & Software* **25**, 891–909.

May, O., De Roo, B., Miserez, K. & Luts, D. 2019 Follow your wastewater using the open source Graph Tracing Engine. Slidedeck FOSS4G, BE-GOOD, Interreg North-West Europe, Brussels, Belgium, 24 October. Available from: https://drive.google.com/file/d/1TBxqlcLLrSxsIAy3yPSNQYu0YMG1NDqd/view.

McCabe, M. F., Rodell, M., Alsdorf, D. E., Miralles, D. G., Uijlenhoet, R., Wagner, W., Lucieer, A., Houborg, R., Verhoest, N. E. C., Franz, T. E., Shi, J., Gao, H. & Wood, E. F. 2017 The future of Earth observation in hydrology. *Hydrology and Earth System Sciences* **21** (7), 3879–3914.

Moriasi, D. N., Arnold, J. G., Van Liew, M. W., Bingner, R. L., Harmel, R. D. & Veith, T. L. 2007 Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE* **50** (3), 885–900. https://doi.org/10.13031/2013.23153.

Pagán, B., Desmet, N., Seuntjens, P., Bollen, E. & Kuijpers, B. 2020 Data driven methods for real time flood, drought and water quality monitoring: applications for Internet of Water. In: *EGU General Assembly 2020*, 4–8 May, EGU2020-9291. https://doi.org/10.5194/egusphere-egu2020-9291.

Perrin, C., Michel, C. & Andréassian, V. 2003 Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology* **279** (1–4), 275–289. https://doi.org/10.1016/S0022-1694(03)00225-7.

Ramsey, P. 2020 PostGIS 3.1.0. postgis.net (18 December). https://postgis.net/2020/12/18/postgis-3.1.0/.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N. & Prabhat 2019 Deep learning and process understanding for data-driven Earth system science. *Nature* **566** (7743), 195–204.

Seth, S. & Samal, A. 2016 Conflation of features. In: *Encyclopedia of GIS* (Shekhar, S., Xiong, H. & Zhou, X. eds.), Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-319-23519-6_181-2.

Solomatine, D. P. & Ostfeld, A. 2008 Data-driven modelling: some past experiences and new approaches. *Journal of Hydroinformatics* **10** (1), 3–22.

Urubkin, M., Galushka, V., Fathi, V., Fathi, D. & Gerasimenko, A. 2020 Representation of graphs for storing in relational databases. E3S Web of Conferences **164**, 09014.

US Environmental Protection Agency 2021 Water Quality Data. https://www.epa.gov/waterdata/water-quality-data (accessed 25 June 2021).

US Geological Survey 2015 Streamer. https://txpub.usgs.gov/DSS/streamer/web/ (accessed 25 June 2021).

US Geological Survey 2021 USGS Water Data for the Nation. https://waterdata.usgs.gov/nwis/ (accessed 25 June 2021).

Water Data Labs 2021 Hydro-Network Linked Data Index. USGS. Available from: https://labs.waterdata.usgs.gov/about-nldi/index.html (accessed 22 October 2021).

Zhang, D., Lindholm, G. & Ratnaweera, H. 2018 Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring. *Journal of Hydrology* **556**, 409–418.

Zhao, R. J. 1977 *Flood Forecasting Method for Humid Regions of China*. East China College of Hydraulic Engineering, Nanjing, China.

First received 8 September 2021; accepted in revised form 26 November 2021. Available online 13 December 2021